UCLouvain

epl

École polytechnique de Louvain

Real-time intention detection of locomotion modes using Electromyography

Author: **Virginie GILLIS**
Supervisor: **Renaud RONSSE**
Readers: **Frédéric CREVECOEUR, Aleksandar JANKOVSKI, François HEREMANS**
Academic year 2018–2019
Master [120] in Electro-mechanical Engineering

Abstract

The good control of a lower limb powered prosthesis is important to allow the amputee to recover a normal and comfortable gait. The high-level control of such a prosthesis can be implemented based on the user's locomotion modes intention. To this end, this thesis investigates the use of electromyography (EMG) to detect the gait intention of healthy subjects. The muscle activities of eight muscles of the thigh were measured and the EMG signals were sampled at a frequency of 1000 [Hz]. Time-domain features were extracted from the signals over time-windows obtained via a sliding window algorithm. The patterns were classified with a linear discriminant analysis (LDA). The best accuracy, 96,43%, was acquired when four tasks were differentiated at a normal gait cadence: still standing, level-ground walking, stair ascent and descent. Experiments conducted with the same parameters but other subjects scored 92% and 90%. However, when adding the ramp ascent and descent tasks, when using a fast or slow gait cadence or when reducing the number of measured muscles, the classification accuracy was significantly reduced. In order for the results to be clinically accepted, a compensation for the EMG's noise sensitivity and non-stationarity is needed. This can be done by adding mechanical sensors as for instance, IMU's or instrumented insoles. Such neuromuscular-mechanical fusion methods are a step further to the good control of powered lower limb prostheses.

Acknowledgements

First of all, I would like to thank my supervisor for his support and precious advices.
For providing me all the needed materials on short notice, thank you to Paul Sente.
Many thanks also to the people who helped me placing the surface electrodes: Bastien Brack, Jonathan Cachet, Brieuc Dubois, Eva Wolff, Souphie Boulet and Antoine Laloux.
Last but not least, I would like to thank all my experimental subjects who spend couple of hours walking and climbing stairs and ramps: Arnaud Sibille, Anouk Rogiest, Agathe Kelecom and Thomas Devos.

List of Abbreviations

CNS	Central nervous system
PNS	Peripheral nervous system
CPG	Central pattern generator
BCI	Brain-computer interface
EEG	Electroencephalography
FNIRS	Functional near-infrared spectroscopy
IAE	Implanted array of electrodes
EMG	Electromyography
MMG	Mechanomyography
ID	Inverse dynamics
RF	Rectus femoris
VL	Vastus Lateralis
BF	Biceps Femoris
SMT	Semitendinosus
GMax	Gluteus Maximus
GMed	Gluteus Medius
SAR	Sartorius
GR	Gracilis
AM	Adductor magnus
TFL	Tensor fascia latae
ROS	Robot operating system
ADC	Analog-to-digital converter
GPIO	General purpose Input/Output
IMU	Inertial measurement unit
MAV	Mean absloute value
ZC	Zero crossings
SSC	Slope sign changes
WFL	Waveform length
VAR	Variance
RMS	Root mean square
ARC	Autoregressive coefficients
FM	Frequency mean
LDA	Linear discriminant analysis
QDA	Quadratic discriminant analysis
DBN	Dynamic Bayesian network
SVM	Support vector machine

ANN	Artificial neural network
CA	Classification accuracy
ST	still Standing
WA	level-ground Walking
SA	Stair ascent
SD	Stair descent
RA	Ramp ascent
RD	Ramp descent

Contents

Introduction

The number of amputees in the world were estimated at 0.06% by a study performed in 2004 [1]. Among them, 95% are lower-limb amputees with 52% of transtibial amputations (amputation below the knee) and 38% of transfemoral amputations (amputation above the knee). According to a study from UCL conducted in 2013 [2], the number of amputees in Belgium is evaluated to 7700 with 1800 new cases per year.
Passive prostheses have been used for a long time to aid amputees to recover their locomotion. However, they do not completely respond to the needs of the user. Their locomotion is often limited to slow walking because of their impaired muscles that can not provide the necessary energy to perform other gait types. This renders any other locomotion modes as fast walking or stair climbing very effort-demanding and uncomfortable. To enhance the amputees' condition and try to provide a natural and comfortable way of locomotion, powered prostheses have been developed over the last decades. Indeed, they can provide the extra needed energy to perform all the possible locomotion modes.

However, even if a few prosthesis slowly begin to appear on the market as for example, the iWalk BiOM foot [3], the majority of them are still developed in research fields and are subject to engineering challenges. One of these challenges is the control strategy needed in active prosthesis. Indeed, based on the performed movement, an active prosthesis needs a different controller. For instance, the controller used for level-walking will not be the same as the one used for stair climbing. Therefore, to ensure that the prosthesis follows the user's intention in a comfortable way, the prosthesis needs to know the locomotion mode intention of the user. This would then allow to design a high-level control of the prosthesis. In fact, once the locomotion mode is detected, the high-level control can proceed the information to the mid-level controller which transforms the information in a signal that can be used as an input by the lower-level controller which can then directly control the actuators.

Therefore, the goal of this thesis is to detect, in real time, the intention of different locomotion modes of healthy subjects with the further motive to aid the implementation of a high-level control of an active transtibial prosthesis. The gait types that are willing to be differentiated are: still standing, level-ground walking, stair ascent, stair descent, ramp ascent and ramp descent.
Firstly, all the known and tested methods aiming the gait intention detection are presented through a state of the art in chapter 1 and a promising method is then chosen to be investigated in this thesis. Then comes an introduction to the biomechanics of the human gait presented in chapter 2. The *Method* chapter goes through all the implemented steps

to achieve the objective. Afterwards, chapter 4 presents the obtained results and finally, chapter 5 proposes a discussion of these results and of the used methods.

Chapter 1

State of the art

The first part of this chapter presents a review of the methods used in the literature to detect the locomotion modes intention for lower-limb amputees. It then compares these different methods and presents the chosen method that will be investigated during this thesis.

1.1 Presentation of the different methods

Tucker et al.[4] propose a review of the control strategies for active lower-limb extremity prosthetics and orthotics. A prosthesis replaces a removed limb as an orthesis assists an impaired limb. The intention of the user to perform an action can be detected by the sensing of the neural activity in the brain, the neural activity in the muscles, the posture of the user and the environment in which the user moves. A comprehensive review of the possible methods used to sense these is proposed by the article of Tucker et al.[4] and is presented in the next subsections.

1.1.1 Neural activity

The human nervous system can be divided into two main parts: the central nervous system (CNS) including the brain and the spinal cord and the peripheral nervous system (PNS) which relays the neural information of the CNS to the rest of the body. As shown in figure 1.1, the PNS is made of the efferent pathway that conducts the motor information from the brain to the muscles and the afferent pathway that conducts the sensory feedback from the muscles to the brain. The movements achieved by the muscles are controlled vias three different sources : basic motor patterns generated at spinal level by the central pattern generator (CPG), the voluntary control starting at the cortical level (brain) and the reflexive path which bypass the supraspinal level and comes from the sensory feedback through the afferent path.

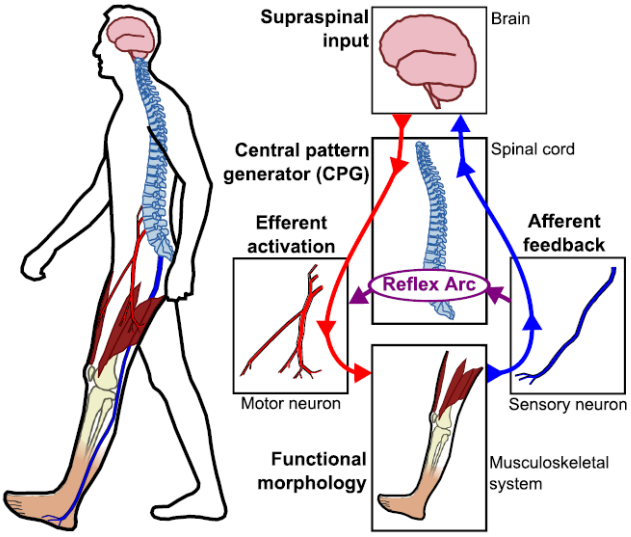


Figure 1.1: Nervous system of the human body [4]

1.1.1.1 Central nervous system signals

In order to sense the neural activity at the cortical level, brain-computer interfaces (BCI's) are developed. The most common are the BCI's using electroencephalography signals but other techniques also exist.

Electroencephalography (EEG) EEG is the measure of the potential difference between electrodes on the cortical surface. Electrodes are positioned on the scalp using an array of surface electrodes according to a standardized system, the international 10-20 system shown in figure 1.2.

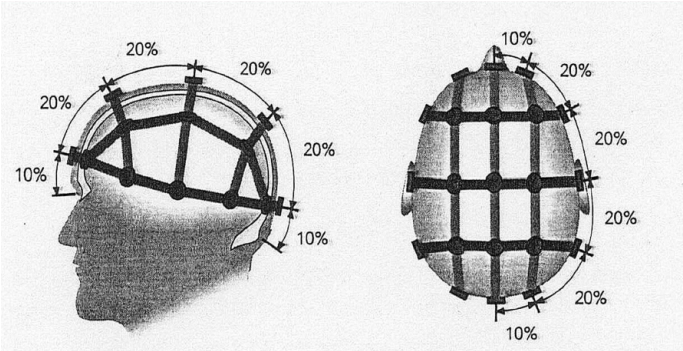


Figure 1.2: International 10-20 array of surface electrodes used for EEG [5]

The main disadvantages of this technique are the extreme difficulty and time to put on the electrodes by one self and the sensitivity to motion artifacts and electrical noise. J. d. R. Millan et al. [6] propose a review and the challenges of BCI's used with assistive technologies.

Functional near-infrared spectroscopy (FNIRS) This technique "uses optical light emitters and receivers placed on the scalp to sense the haemodynamic response of the brain, which correlates with brain activity"[4].
The FNIRS has a reduced sensitivity to head motion artifacts but on the other hand is sensitive to haemodynamic delay and optodes (optical sensor) have to be worn on the head. Rana M. et al.[7] investigate fNIRS-BCI applications in rehabilitation of patients with lower limb motor impairment after stroke.

Implanted array of electrodes (IAE) With this technique electrodes are implanted in the cortex to record a wide variety of movements. In addition, it could allow for a sensory feedback. However, it is a highly invasive technique which requires a surgical procedure. Bensmaia SJ. and Miller LE.[8] investigate the possibility to restore the sensory-motor function with implanted BSI's.

All these methods sense the activity at the cortical level and hence, only the information about the voluntary control is captured which is one of the two main drawbacks of these techniques. Indeed, the automated basic motor patterns responsible for the basic gait and the reflexive motions control are generated at a lower level than the supraspinal level. Hence, the sensing of the neural activity in the brain will not contain this information yet very important. Secondly, the brain is the centre of a lot of neural activity not only related to the locomotion of the lower limbs. Therefore, to capture useful signals, the subject should focus hard on its gait. That renders the use of these methods in an external environment where distractions can occur very difficult.

1.1.1.2 Peripheral nervous system signals

The closer the neural activity is recorded to the actual muscles performing the action, the more information the signals will contain. Indeed, as shown in figure 1.1, if signals are measured on the peripheral muscles, information about all the control sources can be obtained: the voluntary control, the basic motor patterns control and the reflexive control. However, the capture of the reflexive control is only possible for impaired limbs if the sensory feedback is restored. There is one dominant technique to measure the muscle activity which is the electromyography.

Electromyography(EMG) EMG records the neuromuscular activity thanks to electrodes that can be invasive or not (surfaces electrodes). According to the articles [9] and [10], the electromechanical delay between the motor commands and the force generation in the muscle are between 30 and 100 [ms]. That means that a certain advance on the actual movement can be obtained with the EMG. Together with the mechanical methods, the EMG is one of the most used technique in the literature for locomotion patterns recognition. The following articles are all using EMG technique to detect the gait intention for lower-limb amputees: [11], [12], [13], [14], [15], [16] and [17]. The main drawback is that the electrodes are sensitive to motion artifacts, electrical noise, crosstalk between the adjacent muscles and are subjected to muscle fatigue. John A. Spanias et al. [18] propose a compensation for these disturbances. The EMG method also requires a pattern recognition

technique and hence a calibration phase.

1.1.2 Mechanical sensors

Mechanomyography (MMG) This technique estimates force production in the muscle by measuring the sound or vibrations on the surface of the skin using microphones or accelerometers. An advantage over the EMG technique is that it is less sensitive to fatigue but on the other hand, it is highly sensitive to motion artifacts. Ashwin P.H. Needham et al. [19] investigate the use of MMG in classification of gait intent.

Joints position and moments by inverse dynamics (ID) Inverse dynamics is a method to calculate the angles, the forces and the moments at each joint (ankle, knee and hip) based on kinematics measurements and external forces. The limb segment orientations can be measured by goniometers, inclinometers, accelerometers, gyroscopes, magnetometers or inertial measurements unit (IMU). External forces as the ground reactions can be sensed by using instrumented insoles worn under the foot or via foot switches that provide binary ground contact information as force-sensitive resistors, sensed air pressure in sealed tube or physical switches. This method is one of the most used in the literature. Abdul Hadi Abdul Razak et al. [20] propose a review of the foot plantar pressure measurement systems. In order to detect the locomotion modes, Aaron J. Young et al.[21][22] use 13 on-board sensors including potentiometers and encoders at the knee and ankle, an axial load cell and a six axis inertial measurement unit (IMU's) located on the shank. For the same purpose Maja Gorsic et al. [23] use two wireless pressure-sensitive shoe insoles and seven IMU's attached to human body segments and Domen Novak et al. [24] use also sensorized insoles and 9 IMU's, one on each foot, shank, thigh and upper arm and a ninth IMU was placed on the back. As for Dongfang Xu et al. [25], they only use two IMU's placed on the shank anf the foot and a load cell fixed in the mid-shank region to measure the interaction force between the human body and the prosthesis.
The main drawbacks of the mechanical methods is that the intent in often detect after the actual movement and that it can not detect voluntary movements.

1.1.3 Environment

Sensing the environment can be very useful to face complex environments and obstacles. However, the only knowledge of the environment is not sufficient to be able to detect the locomotion modes intention. And as proof, gait intention detection applications using only environment methods are not easily found in the literature. Indeed, these methods have to be coupled with others techniques proposed previously in this section.
The state of the environment can be directly determined by using sensors made for this purpose or indirectly through the state of the user or the prosthesis. Here follows some studies found for the environment detection but not specially used for rehabilitation robotics.
For implicit environment sensing, the properties of the terrain as the slope can be identified through IMU's or EMG as done by [26], [27] and [28].

As for explicit environment sensing, the slope and height of the terrain can be estimated with a body-worn laser and IMU's [29] or else obstacles could also be detected thanks to an array of sonar sensors or digital cameras. This has be done by [30] for a intelligent wheelchair application.

1.2 Methods comparison

To chose the method to be investigated for this thesis, a comparison of the previously presented techniques is made in table 1.1. This comparison is based on the following criteria:

- Timing: evaluates the delay between the actual movement and the prediction
- Voluntary control: evaluates if the technique is able to detect voluntary control occurring at supraspinal level in addition to the basic motor patterns generated at spinal level.
- Implementation: evaluates the ease to install the needed devices.
- Invasiveness: evaluates if the needed device is invasive or not.
- Sensitivity : evaluates the sensitivity to the electrical noise, motion artifacts, crosstalk,...
- Prosthesis integration: evaluates the possibility to embed to needed devices in the prosthesis.

	FNIRS	**EEG**	**IAE**	**EMG**	**MMG**	**ID**
Timing	-	+	+	+	-	-
Voluntary control	+	+	+	+	-	-
Implementation	-	-	-	+	+	++
Invasiveness	+	+	-	+	+	+
Sensitivity	-	-	-	-	-	+
Prosthesis Integration	-	-	-	+	+	+

Table 1.1: Comparison of the methods used in the literature to detect the intention of locomotion modes

The two best methods are the electromyography and the inverse dynamics computation through mechanical sensors. The main drawback of the EMG is its sensitivity to noise but it certainly have two advantages over the inverse dynamics: the advance which with it can provide a prediction and the fact that it includes voluntary control. And because, the sensitivity drawback is something that can be worked on (techniques exist to reduce this sensitivity to noise) the **Electromoyography** is the method chosen to be investigated in this thesis.

Chapter 2

Biomechanics of human gait

This chapter introduces all the biomechanical aspects needed to understand the human gait. It begins by explaining the neuromuscular activity and hence the electromyography technique. Then, the locations and functions of the muscles of the thigh are presented followed by the different gait cycles. Finally, the choice of recorded muscles is justified.

2.1 Electomyography

The main idea behind electromyography is to analyse the muscle activity during one particular task to know when which muscle is activated and when it ceases activity. This way, information about the control of the joints movements is provided. The electrical activity in the muscles can be measured by electrodes placed on the muscle belly. In order to understand what an EMG signal is made of, the notion of motor unit and action potential must first be explained.

Motor unit A motor unit, as shown in figure 2.1 consist of a single motor neuron and all fibers it innervates (10 - 1000 fibers). Activation of a motor neuron will always induce an activation of muscle fibers. Groups of motor units work together to coordinate the contraction of one muscle.

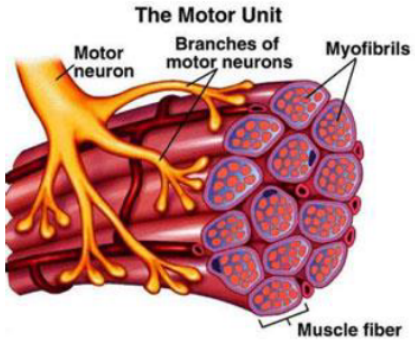


Figure 2.1: Illustration of a motor unit [5]

Action potential An action potential is an alteration of the membrane voltage. This depolarization causes a current through the membrane. The current forms a current loop that will rise the potential at the resting part of the membrane and when it rises above the threshold, the action potential will spread. Figure 2.2 shows a schematic action potential and figure 2.3 shows the propagation of an action potential through an axon (nerve fiber).

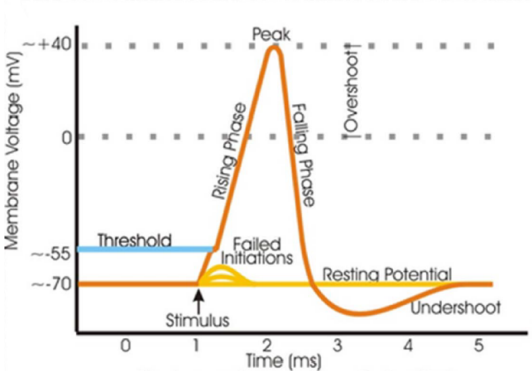


Figure 2.2: Signal of a schematic action potential[5]

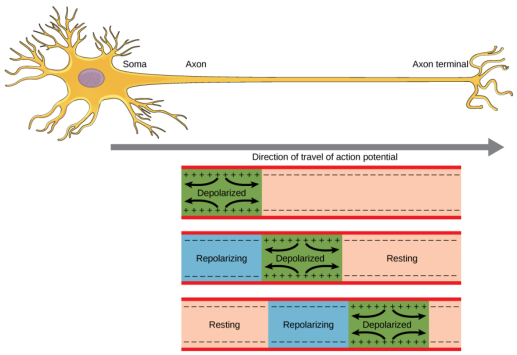


Figure 2.3: Propagation of an action potential[31]

The electromyogram The measure of this action potential is called an electromyogram. It is composed of multiple action potentials of all active motor units superposed. Figure 2.4 shows how an electromyogram is build and figure 2.5 shows a typical raw electromyogram. The raw signal has positive and negative components and has an amplitude smaller than 1 [V]. This amplitude varies with a number a factors, most of them related to the muscle properties and to the electrodes placement. Moreover, the amplitude of the EMG does not always have a linear relationship with the muscle force. The signal is used to be able to determine the order of different muscles activation and not the produced forces by the muscles.

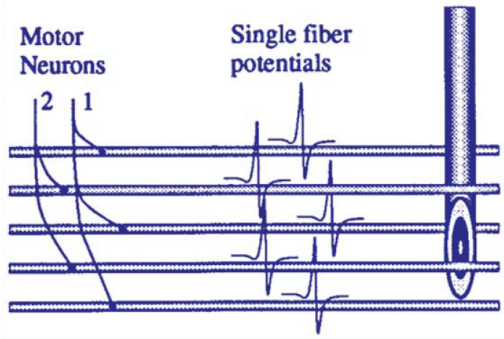


Figure 2.4: Illustration of the components of an electromyogram [5]

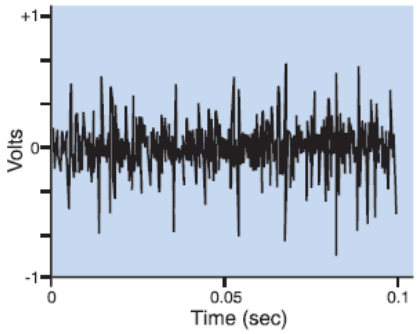


Figure 2.5: A sigle raw EMG record [32]

The EMG envelope In order to get information about the amplitude of a raw electromyogram, an EMG envelope can be determined. The envelope is more easily interpreted visually and represent the volume of activity. The first step to get the envelope is to rectifie the signal, that is, taking the absolute value of the signal. This step is called full wave rectification and is important because an EMG signal is naturally nearly zero mean so smoothing the raw signal directly will just give zero. After the rectification, they are two ways to get the EMG envelope. The first one is to filter out the signal of its high

frequency content. This determines a linear envelope. The second way is to integrate the rectified signal which represent the area under the curve of this signal or in other words the accumulated EMG activity. In order to do that, the signal is divided into fixed-width timeslices: the integral is calculated over each time slice and is reset at the beginning of each slice. The maximum of each timeslice is then taken to give an approximation of the EMG envelope.
The EMG envelope can be obtained thanks to electrical hardware. The transformed signals starting from the raw EMG are shown in figure 2.6.

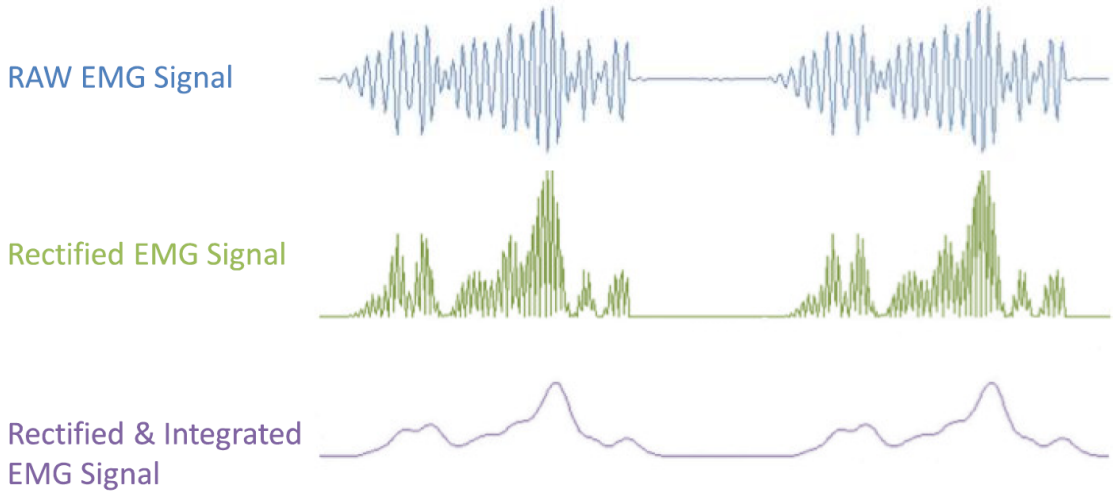
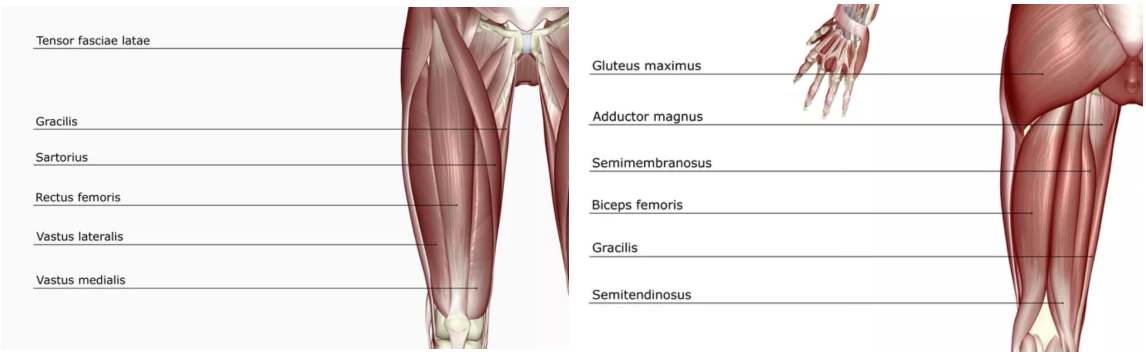


Figure 2.6: Illustration of a raw EMG signal, a rectified EMG signal and an EMG envelope [33]

2.2 Choice of the measured muscles

The further purpose of the thesis is to control a prosthesis for an amputee. For transtibial amputees (under the knee), only the muscles of the leg are impaired and for transfemoral amputees (above the knee) these muscles are absent and the muscles of the thigh are impaired. Hence, it is better to work with the muscles of the thigh to reduce the probability to have absent or impaired muscles. Moreover, techniques as targeted muscle reinnervation [34] exist for transfemoral amputees to retrieve the musclar activity of the impaired muscles. Thus, the chosen recorded muscles will be muscles from the thigh.
The EMG signals are recorded with surfaces electrodes which means that only superficial muscles (muscles that are in surface, in opposition to deep muscles) can be measured. Figure 2.7 shows the location of the superficial muscles of the thigh.



(a) Front view **(b)** Back view

Figure 2.7: Location of the muscles of the thigh[35]

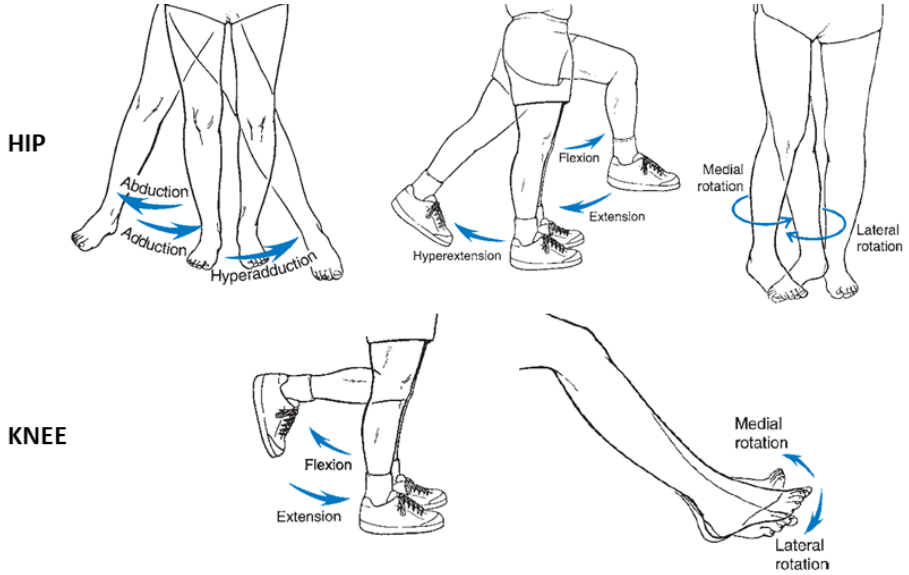


Figure 2.8: Movements of the leg and the thigh at the knee and hip joints [32]

The different actions that the muscles of the thigh can perform are the following and are illustrated in figure 2.8.

- Extension and flexion of the leg at the knee (knee extensors and flexors)
- Extension and flexion of the thigh at the hip (hip extensors and flexors)
- Abduction and addcution of the thigh at the leg(hip adductors and abductors)
- Medial and lateral rotation of the leg at the knee (knee medial and lateral rotators)
- Medial and lateral rotation of the thigh at the hip (hip medial and lateral rotators)

The muscles presented before are each responsible for some of these actions. Figure 2.9 shows which muscle plays which role. Because to perform the desired tasks (still standing, level-ground walking, stair ascent and descent, ramp ascent and descent) the rotation

actions are not the mainly used movements, they are not considered in the following classification.

Knee Extensors • Rectus Femoris • Vastus Lateralis • Vastus Medialis } Quadriceps	**Hip Extensors** • Gluteus Maximus (GMax) • Biceps Femoris (BF) • Semitendinosus (SMT) • Semimembranosus (SMM)	**Hip Abductors** • Gluteus Maximus • Tensor Fascia Latae (TFL)
Knee Flexors • Biceps Femoris • Semitendinosus • Semimembranosus } Hamstrings • Sartorius (aids) • Gracilis (aids)	**Hip Flexors** • Rectus Femoris (RF) • Vastus Lateralis (RF) • Vastus Medialis (RM) • Sartorius (SAR)	**Hip Adductors** • Adductor Magnus (AM) • Gracilis (GRA)

Figure 2.9: Classification of the thigh muscles based on the joints movements they are responsible for

Moreover, it is useful to see which muscles are mainly activated during a gait cycle of the different tasks. A gait cycle can be divided in different phases. The important events are the heel strike and the toe-off which separate the stance phase from the swing phase. One cycle corresponds to two steps. Figure 2.10 shows which muscles are activated during a walking cycle and figure 2.11 during a stair ascent and a stair descent cycle. For the ramp ascent and descent task, this information was not found in the literature.

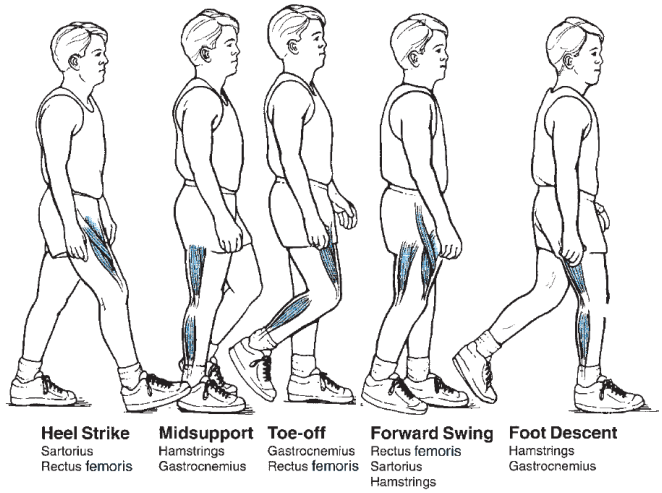


Figure 2.10: Main muscles activity during different phases of a gait cycle during walking [32]

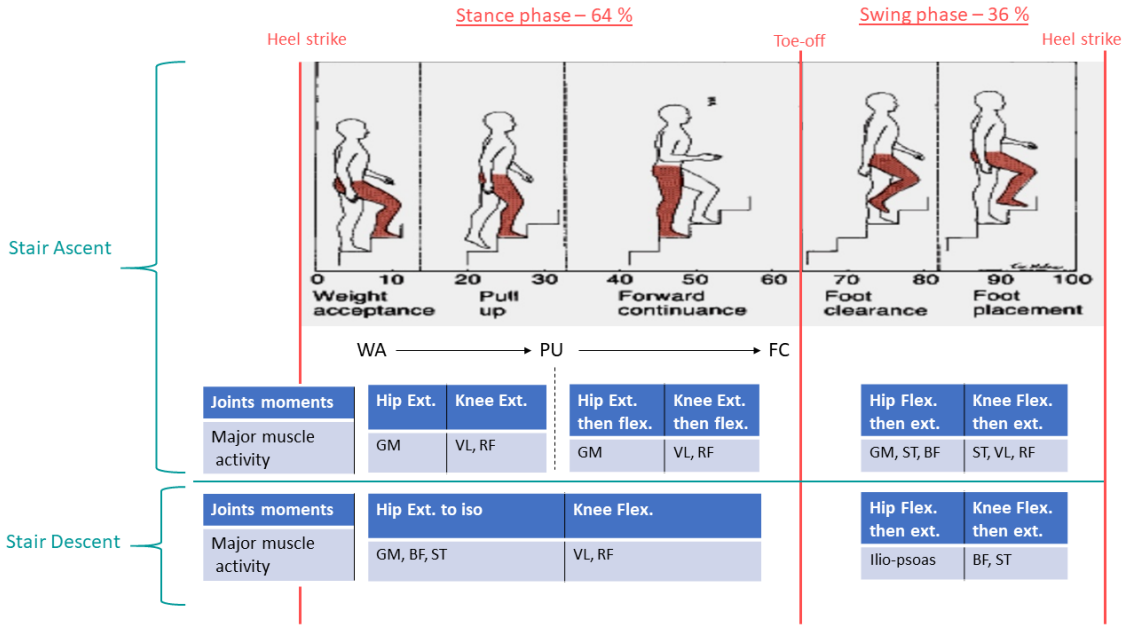


Figure 2.11: Main muscles activity during different phases of a gait cycle during stair ascent[36] and stair descent [37]

A logical choice for the recorded muscles should be a choice where the six different actions described in figure 2.9 could all be performed with the chosen muscles. To aid the selection of the muscles, a review of the literature is made in table 2.1.

Table 2.1 - Review of the literature : Recorded Muscles

Articles		**Muscles**	**Application**
Eva Lendaro et al.[17] 2016	8	SMT, BF, TFL, RF, VL, VM, SA, GRA	Potential Treatment for Phantom Limb Pain
John A. Spanias [18] 2016	9	SMT, BF, TFL, RF, VL, VM, SA, AM, GRA	Detection of and Compensation for EMG Disturbances
Aaron J. Young , et al.[38] 2013	9	SMT, BF, TFL, RF, VL, VM, SA, AM, GR	Intent recognition strategy for transfemoral amputee ambulation
He Huang , et al. [39] 2011	8	ST,BF, RF, VL, VM, SA, AM, GR	Locomotion Mode Identification for prosthetics legs
Ann M. Simon , et al.[40] 2013	8	ST, BF, TFL, RF, VL, SA, AM, GR	Powered prosthesis control during non-weight bearing activities
Aaron J. Young, et al.[41] 2013	7	ST, BF, TFL, VM, SA, AM, GR	Classifying the Intent of Novel Users during human locomotion

E.C. Wentink,et al. [42] 2013	6	GMax, GMed, TFL, RF, VL, BF	Intention detection of gait initiation

Table 2.1: Review of the literature: Comparison of the recorded muscles

Based on this review and the muscles functions, the following eight muscles are chosen: the Rectus femoris, the Vastus lateralis, the Adductor magnus, the Biceps femoris, the Semitendinosus, the sartorius, the Tensor fascia latae and the Gracilis. Figure 2.12 shows that with these eight muscles all the main actions can be performed.

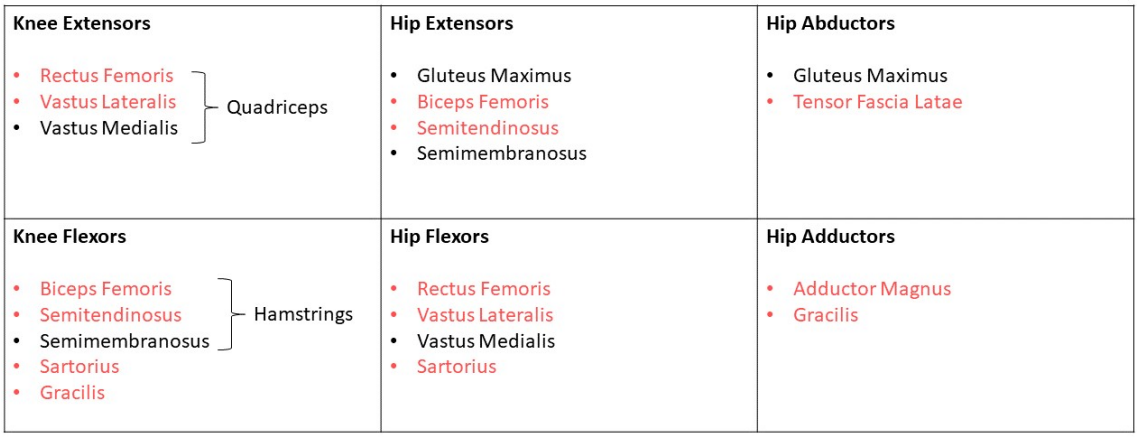


Figure 2.12: Classification of the thigh muscles based on the joints movements they are responsible for, with the chosen measured muscles highlighted

Chapter 3

Method

The goal of this thesis is to be able to detect the intention of six different modes of locomotion : still standing, level-ground walking, stair ascent, stair descent, ramp ascent and ramp descent. This chapter covers all the steps implemented to reach this objective. From the structure of the system and the used devices to the classification, going through the signal treatment. The methods used to evaluate the system are then explained and the chapter ends with the experimental protocol.

3.1 Structure of the system

The block diagram in figure 3.1 shows the global structure of the system. The device used to measure the activity of the muscles is called a MyoWare^{TM} and it filters, amplifies, rectifies and integrates the raw EMG signal measured by the electrodes to output an EMG envelope. In order to process the EMG's in real-time, the signal first needs to be sampled via an analog-to-digital converter. The signal can then be divided into time-dependent analysis windows via a sliding-windows algorithm which will allow to extract features of the signal over these windows. Based on the computed features, the signal can finally be classified.

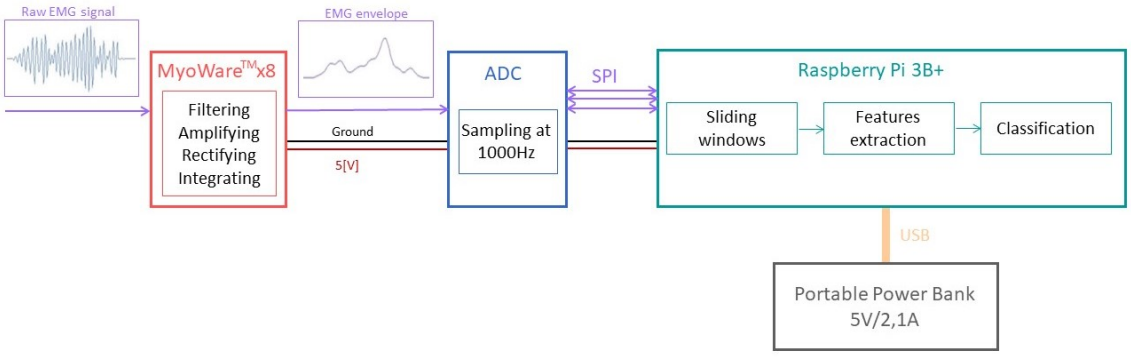


Figure 3.1: Block diagram of the global structure of the system

Regarding the microprocessor of the system, that is the brain of the system, a Raspberry Pi 3B+ is chosen. Indeed, in order to make a future interface between the intention detection system and the prosthesis the easiest, the system is implemented with ROS

(Robot Operating System). To allow this, a Raspberry Pi with Ubuntu Bionic 18.04 and ROS Melodic installed on it is used. The Raspberry Pi 3B+ and the ananolg-to-digital converter, which device choice is explained in sub-section 3.1.2, needs a 5 [V] power supply to work correctly. Therefore, the system is powered by a portable power bank with 5V and 2.1A output. Figure 3.2 shows pictures of the system.

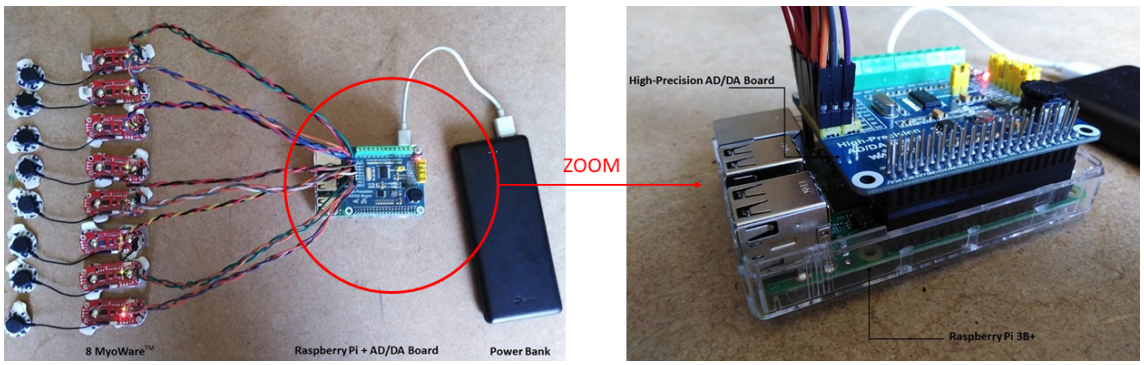


Figure 3.2: Pictures of the system including the 8 MyoWare^{TM}'s, the ADC, the Raspberry Pi 3B+ and the power bank

This section first introduces the device used to record the EMG signals, it then justifies the choice of the ADC device and ends with a presentation of the ROS global structure implemented on the Raspberry Pi.

3.1.1 Myoware^{TM}

Electrodes The electrical activity of the muscles can be measured with electrodes. There are two types of electrodes: invasive (indwelling electrodes) or non-invasive (surface electrodes). These last ones are mainly used for superficial muscles measurement, that is muscles that are located in surface, as the indwelling electrodes are used for measuring deeper and smaller muscles. For practical reasons, this thesis will prefer the use of surface electrodes. There are two different possible configurations for surface electrodes: a **Monopolar** arrangement and a **Bipolar** arrangement. The monopolar configuration consists of one electrode placed on the muscle and a second one placed on an electrical neutral site (the reference electrode). However, this arrangement is not adapted for non-isometric movements. Indeed, the muscle can change in length both ways during a contraction : *Concentric* , that is the muscle shortens as it is contracting; *Excentric*, that is the muscle lengthens as it is contracting and *Isometric*, that is the muscle is in tension but no change in the muscle length occurs. Hence, non-isometric movements are the most common in biomechanics and therefore the bipolar configuration is preferred to the monopolar one. The bipolar arrangement consists of two electrodes placed on the muscle (1.5 to 2 [cm] apart) and a third reference electrode.

Device choice To sum up, the needed device has to have a bipolar configuration of surface electrodes and, as explained in section 2.1, ideally, it should treat the signal to have

an EMG envelope output which is more easy to use with a microcontroller.
The chosen device is therefore, the **MyoWare^{TM} Muscle sensor** [33] shown in figure 3.4. It contains a bipolar configuration of surfaces electrodes and the necessary hardware to amplify, filter, rectify and integrate the signal. It outputs an EMG envelope of 0 to 5 [V] that can easily be used with a microcontroller if it is sampled via an analog-to-digital converter (ADC).
The correct placement of the device is important. The pair of electrodes have to be placed in the middle of the muscle, in the fiber alignment and the reference electrode must not be placed on the same muscle. Figure 3.3 shows the effects of misplaced electrodes on the raw EMG signal.

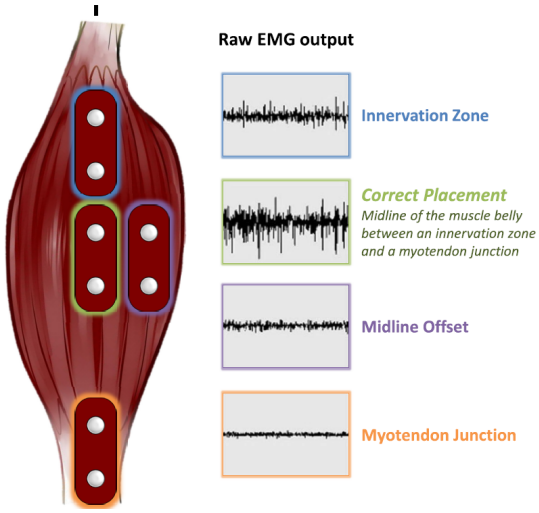


Figure 3.3: Effects of the misplacement of the surface electrodes on the signal[33]

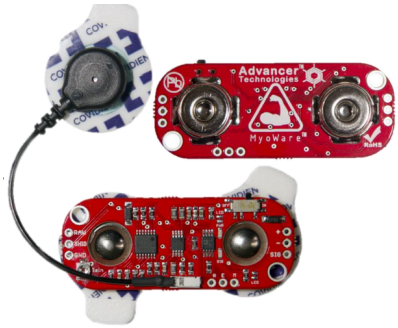


Figure 3.4: Picture of the MyoWare^{TM} Muscle sensor [33]

3.1.2 Analog-to-Digital Converter (ADC)

For choosing which ADC device was going to be used, three major parameters had to be considered :

- The device had to be compatible with the Raspberry Pi 3B+.
- The device had to be able to sample at the desired rate.
- The device had to have minimum 8 analog inputs.

Number of analog inputs Indeed, according to the articles shown in table 2.1, an average of eight muscles are measured to ensure that the classification is accurate enough. It was therefore decided to record eight muscles (section 2.2) and an ADC board with eight analog inputs channels is required.

Sampling rate To determine the desired sampling rate, the frequency content of an EMG is observed. Figure 3.5 shows that the highest frequency content, f_{max}, of an electromyogram is typically of 500 [Hz]. According to the Nyquist theorem, the sampling rate, F_s, has to be greater than $2 * f_{max}$ giving a sampling rate of minimum 1000 [Hz]. Moreover, a sampling rate of 1000 [Hz] is chosen by most of the articles out of the literature presented in table 3.1 and therefore, this value is chosen for this thesis.

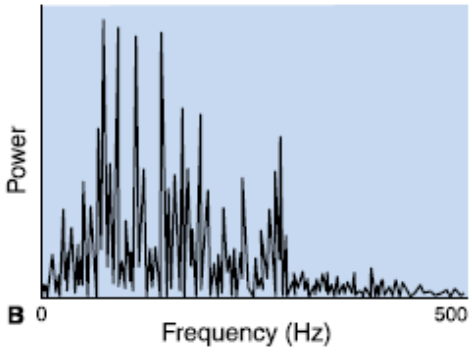


Figure 3.5: Frequency content of an EMG signal[32]

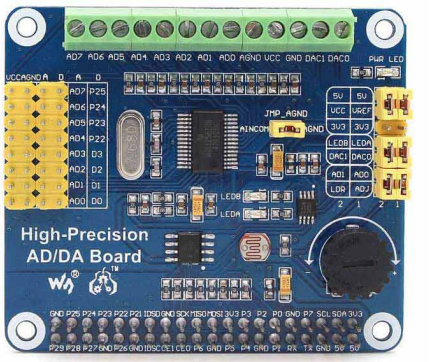


Figure 3.6: Picture of the High-Precision AD/DA Board [43]

Table 3.1 - Review of the literature : Sampling frequency F_s

Articles	F_s **[Hz]**	**Application**
Aaron J. Young , et al.[38] **2013**	1000	Intent recognition strategy for thansfemoral amputee ambulation
He Huang , et al. [39] **2011**	1000	Continuous Locomotion-Mode Identification for prosthetics legs
Ann M. Simon , et al.[40]**2013**	1000	Powered prosthesis control during non-weight bearing activities
Aaron J. Young, et al.[41] **2016**	1000	Classifying the Intent of Novel Users during human locomotion
Eva Lendaro and Max Ortiz-Catalan [17] **2016**	2000	Potential Treatment for Phantom Limb Pain
J A Spanias, et al. [44] **2018**	1000	Online adaptive neural control of a robotic lower limb prosthesis
Aaron J. Young, et al.[45] **2014**	1000	Enhance intent recognition in powered lower limb prostheses

Articles	F_s **[Hz]**	**Application**
Deepak Joshi, et al. [28] **2016**	1500	Terrain and Direction Classification of Locomotion Transitions
D.C. Tkach and L.J. Hargrove [46] **2013**	1000	Predicting Ambulation Mode Transitions for Trans-Tibial Amputees
Angkoon Phinyomark, et al. [47] **2013**	2048	EMG feature evaluation
Jason D. Miller, et al. [13] **2013**	1500	Myoelectric Walking Mode Classification for Transtibial Amputees

Table 3.1: Review of the literature: Sampling frequency value choice

Chosen device The "High-Precision AD/DA Board" [43] shown in figure 3.6 meets all the requirements. It has an on-board ADS1256 integrated circuit [48], eight channels, a good precision of 24 bit, a sampling rate up to 30 [kHz] and a Raspberry Pi GPIO interface. The programmable gain amplifier (PGA) is set to 1 to so that the signal is scaled between ± 5 [V]. Moreover, with a 24 bit resolution and a reference voltage of 5 [V] the last significant bit can be determine :

$$LSB = \frac{2 * V_{ref}}{PGA * (2^{23} - 1)} = 1.1921\,[\mu V]$$

Which is totally acceptable for an amplified EMG signal. Note that the output of the MyoWare™ is only a positive signal because of the rectifying so the ADC will never receive negative values.

Timing According to the datasheet of the ADS1256 chip[48], the multiplexer cycling throughput for a sampling rate of 1000 [Hz] is 837 [Hz], that is, one sample for one channel is obtained in 1.19 [ms]. Moreover, the signals of the eight channels are not sampled all at the same time but are selected by a multiplexer each in turn. Hence, getting one sample for each of the eight channels can not be done faster than 104.625 [Hz]. That is the theory, in real, it has been observed that the system can get one complete sample (1value/channel) at a average rate of 97 [Hz]. This timing is illustrated in figure 3.7. The reason of this delay is not known for sure. It could be that the low-level code to read the channels in not perfectly optimal or that the operating system ROS works not perfectly in real-time.

Communication The Myoware™ send the EMG envelope signal via a wire to the ADC. The output of the ADC, the sampled data, is sent to the Raspberry Pi by SPI (Serial Peripheral Interface) which is a synchronous serial communication interface used for short-distance communication using a master-slave architecture. This protocol only needs 3 GPIO (General Purpose Input/Output) pins, that is, the standard 3 wire serial protocol: SCLK (serial clock), MOSI (Master Out Slave In) and MISO (Master In Slave out).

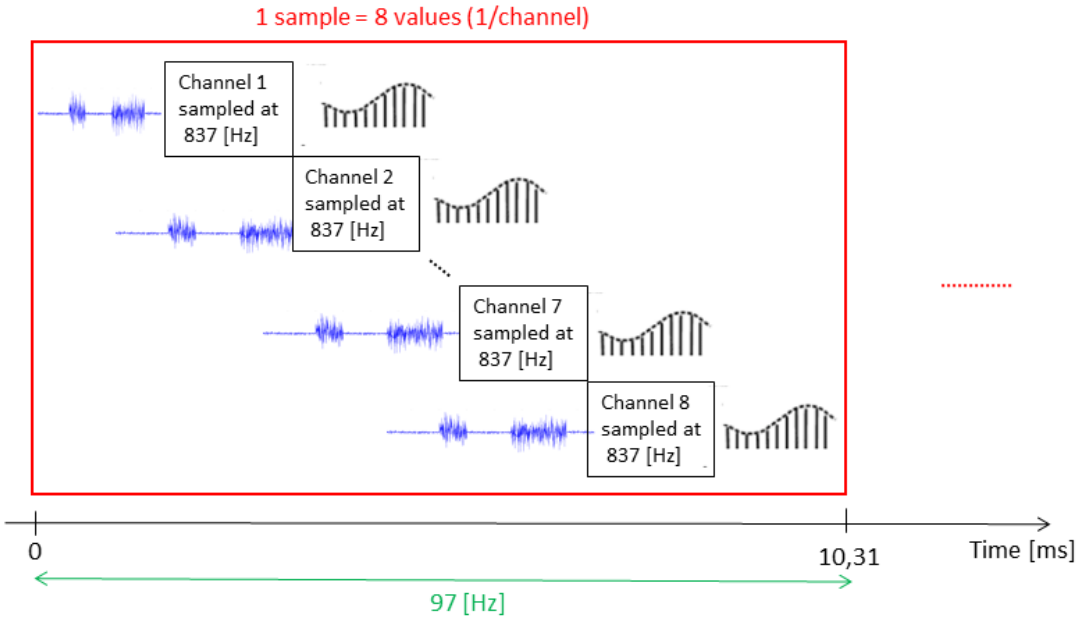


Figure 3.7: ADC Timing One complete sample is composed of one value for each of the eight channels; the ADC is able to get one sample at a rate of 97 [Hz]

3.1.3 ROS nodes structure

Now that the signal is sampled and ready to be used with a microcontroller, it can be processed and then classified. This will be done by the Raspberry Pi 3B+ with the use of the Robot Operating system (ROS) which is a flexible framework for writing robot software. ROS is mainly made of [49] :

- Nodes: an executable that uses ROS to communicate with other nodes. They can subscribe or publish to a topic and also provide or use a service. Nodes can be written in c++ or in python.
- Messages: ROS data type when subscribing or publishing to a topic
- Topics: Nodes can publish messages to a topic or subscribe to a topic to receive a message

Figure 3.8 shows the nodes implemented in ROS in order to perform the signal processing and the classification. A first node, the publisher node, reads the values of the eight channels, stores this values in a message and publishes this message on a topic. A second node, the subscriber node can then subscribe to this topic and get the values of the EMG's stored in the message. Then, through the calling of several services, the subscriber node can perform the signal processing and the classification. In the end, if the user inputs the true tasks that were performed, the system is also able to calculate a score which represent the classification accuracy. The code of all this nodes are accessible in the appendix A The signal processing and the classification are explained in more details in the two next

sections 3.2 and 3.3.

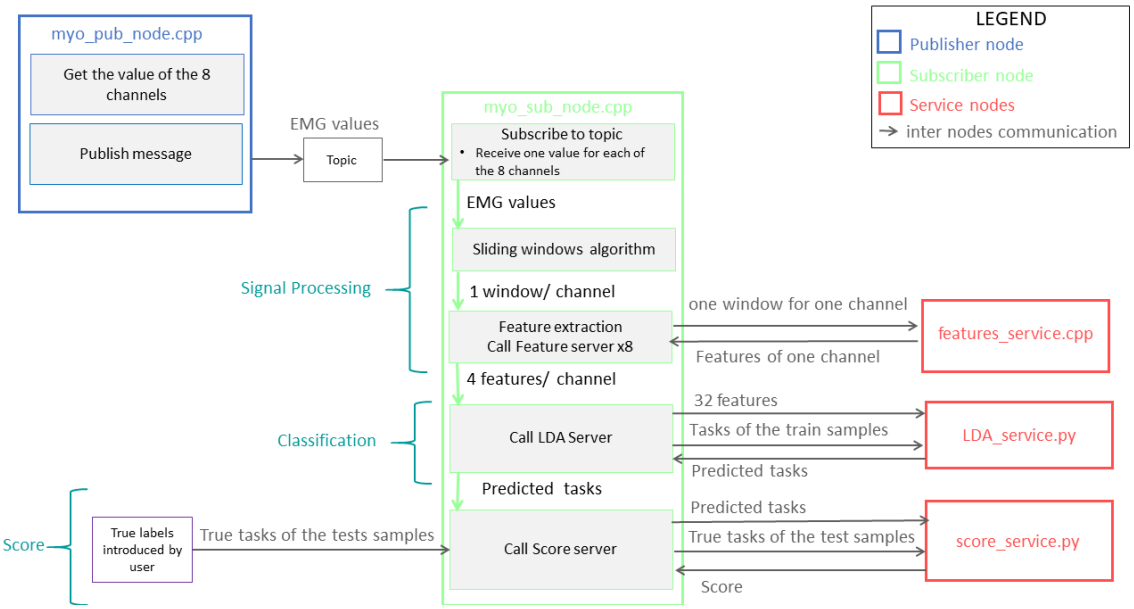


Figure 3.8: Block diagram of the implemented ROS nodes and the data interaction between these nodes

3.2 Signal processing

In order to classify the signals, they have to be compared, a common and easy way to compare different signals is by extracting some features of the signals. To allow this feature extraction in real-time and thereby to provide continuous classification decisions, a sliding window algorithm is needed and is presented in the first part of this section. The extracted features are presented in the second part. Figure 3.9 is zooming into the signal processing part of the implemented nodes block diagram shown in figure 3.8.

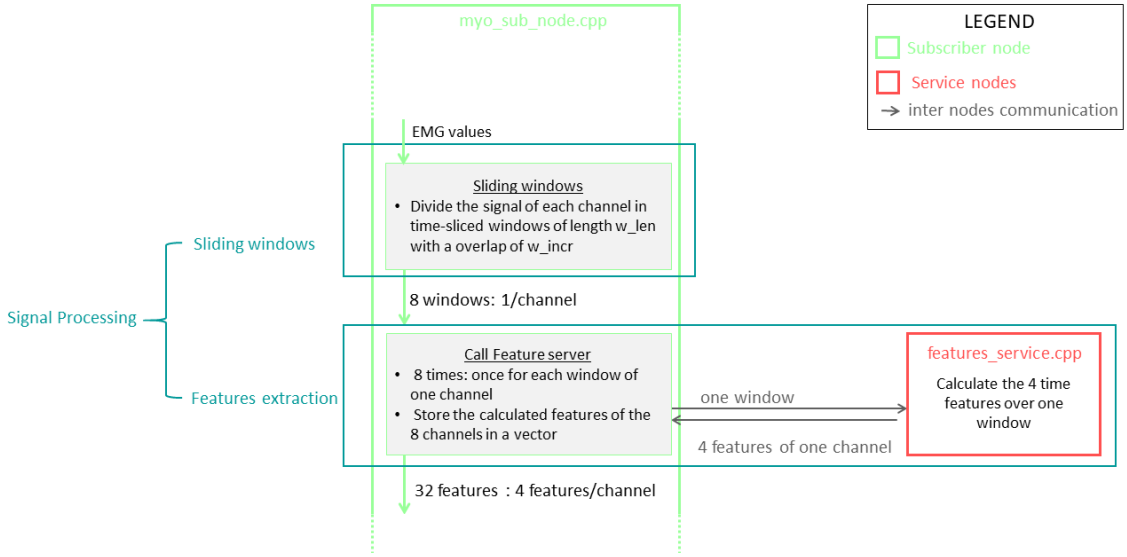


Figure 3.9: Zoom into the signal processing part of the ROS nodes structure shown in figure 3.8

3.2.1 Sliding windows analysis

A sliding window algorithm performs a data segmentation by taking analysis windows of the signal. That is, the signal is sliced into time-dependent windows. One window is characterised by its length expressed in time units or in samples. The windows can also be captured in a way that two consecutive windows overlap each other. This allow more frequent decisions. One decision stands for the action performed at the end of a window, over the data recorded in this window. The extend of the overlapping is determined by the increment value, expressed in time units or in samples. The increment value determines how much data two adjacent windows will share. Hence, one window is characterised by its length but also by its increment as shown on figure 3.10.

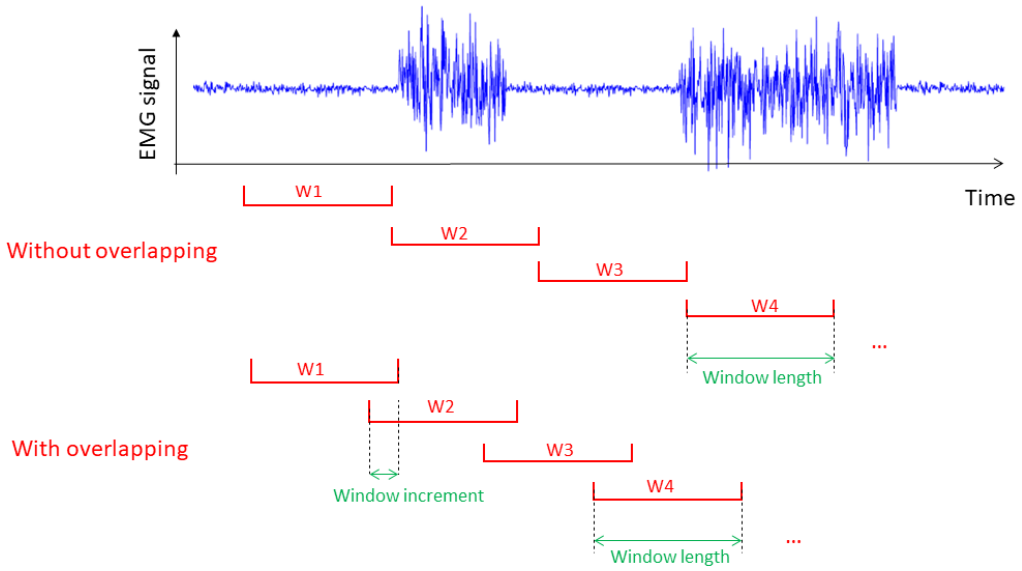


Figure 3.10: Illustration of a sliding windows analysis

Window length and increment value choice The value for the length of these windows and the increment has been decided based on the articles shown in table 3.2.

Table 3.2 - Review of the literature : Window length W_{len} and increment W_{incr} choice

Articles	W_{len} $[ms]$	W_{incr} $[ms]$	**Application**
Aaron J. Young , et al.[38] **2013**	300	-	Intent recognition strategy for thansfemoral amputee ambulation
He Huang , et al. [39] **2011**	150	12	Continuous Locomotion-Mode Identification for prosthetics legs
Ann M. Simon , et al.[40] **2013**	250	15	Powered prosthesis control during non-weight bearing activities
Aaron J. Young, et al.[41] **2016**	300	-	Classifying the Intent of Novel Users during human locomotion
Ou Bai, et al [50] **2017**	100	-	Characterization of lower limb activity during gait
Eva Lendaro et al. [17] **2016**	50	-	Potential Treatment for Phantom Limb Pain
J A Spanias, et al. [44] **2018**	300	-	Online adaptive neural control of a robotic lower limb prosthesis
Deepak Joshi, et al. [28] **2016**	150	50	Terrain and Direction Classification of Locomotion Transitions
D.C. Tkach and L.J. Hargrove [46] **2013**	250	20	Predicting Ambulation Mode Transitions for Trans-Tibial Amputees
Angkoon Phinyomark, et al. [47] **2013**	250	125	EMG feature evaluation

Articles	W_{len} $[ms]$	W_{incr} $[ms]$	**Application**
Jason D. Miller, et al. [13] **2013**	150	-	Myoelectric walking mode classification for transtibial amputees
Deok-Hwan Kim, et al [12] **2014**	100	-	Real-time locomotion mode recognition

Table 3.2: Review of the literature : Window length W_{len} and Window increment W_{incr} choice

These values of the length and increment will have an impact on the decision stream rate. Because the system works in real-time, the faster the decisions are taken the better. Therefore, the chosen window length will be one of the smallest proposed by the literature: the values chosen are 150 [ms] and 12 [ms] for the window length and increment respectively and are based on the article of He Huang, et al. [39] shown in table 3.2.
The number of samples is then determined by $\frac{F_s\,[sps]}{length\,[s]}$. With $F_S = 1000$ [HZ], the following values are set in the code :

- W_{len} = 150 samples
- W_{incr} = 12 samples

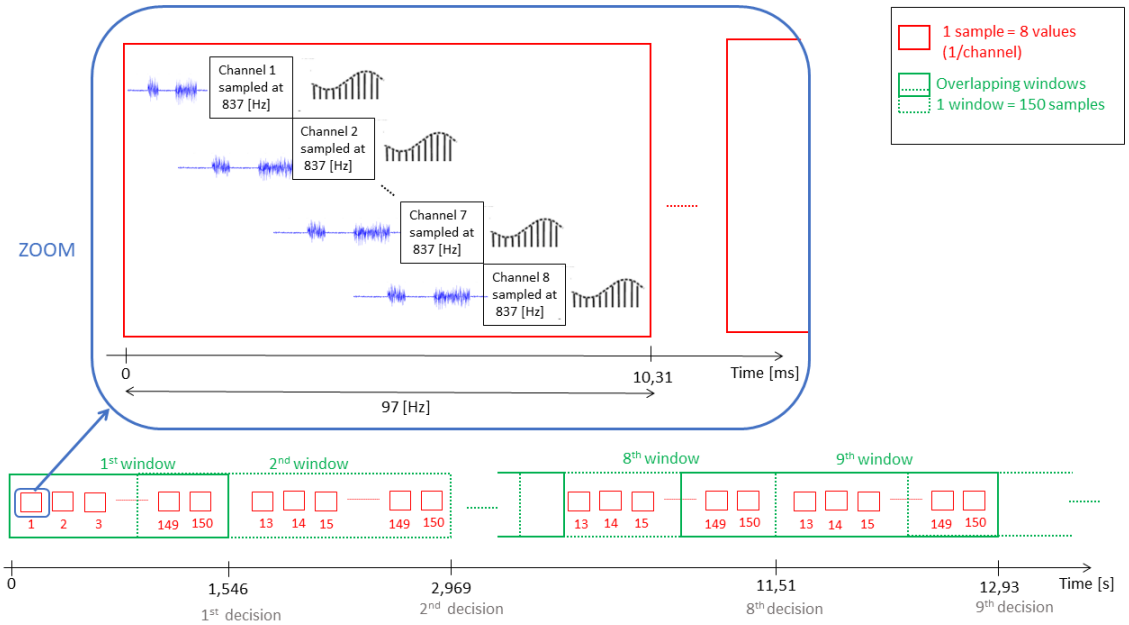


Figure 3.11: Illustration of the decision stream rate

Decision stream rate The time the system takes to make one decision can thus be determined and is illustrated in figure 3.11. As explained in section 3.1.2, one sample is obtained at a rate of 97 [Hz]. Knowing that one window is made of 150 samples and

each window overlaps the next one over 12 samples, the first window will be obtained in about 1.546 [s] and the next windows will be completed in about 1.42 [sec] due to the overlapping. Indeed, it was observed that the system has an average decision stream rate of 0.69 [Hz].

Implemented algorithm The sliding window algorithm is implemented in the myo_sub_node.cpp as shown in figure 3.9 and is build as follows :

Algorithm 1 Sliding window algorithm

while ros::spin **do**
  Subscribe to topic to get data of 8 channels
  Add the 8 EMG values (= 1 sample) to the *window* array
  if length of *window* ==W_{len} **then**
    calculate the features over the window
    classify the window;
    keep the W_{incr} last samples of the current window as the first samples of the next window
  end if
end while

3.2.2 Features Extraction

Features extraction highlight the relevant data of a signal. Thanks to the data segmentation in time-windows made before with the sliding window algorithm, features of the signal can be extracted over each window. Once a set a feature is determined, it can be used to classify the signal. Hence, one decision, that is, the prediction of the task to which the signal belongs, can be made for each window (every W_{len} [ms]). There are three main categories of features for operations on EMG signals: Time-domain, frequency-domain and time-frequency domain features. Here follows some examples of time- and frequency-domain features :

1. Time-domain :
   - Mean absolute value (MAV)
   - Zero crossings (ZC)
   - Slope sign changes (SSC)
   - Waveform length (WFL)
   - Variance (VAR)
   - Root mean square (RMS)
2. Frequency-domain :
   - Autoregressive coefficients (ARC)
   - Frequency mean (FM)

However, frequency related features require transformations involving heavy computations. Hence, time-domain features are more commonly used for EMG signal analysis. Indeed, they are easy and quick to calculate ad they are computed based on the signal input amplitude. Moreover, according to the articles of D. Tkach et al.[51] and A. Phinyomark et al.[47], minimum four features are required to ensure a good classification accuracy. Thus, based on a review of the articles presented in table 3.3, four time-domain features are chosen to be computed.

Table 3.3 - Review of the literature : Features extraction

Articles	**Features**	**Application**
Aaron J. Young , et al.[38]**2013**	MAV, ZC SSC, WFL	Intent recognition strategy for thansfemoral amputee ambulation
He Huang , et al. [39]**2011**	MAV, ZC SSC,WFL	Continuous Locomotion-Mode Identification for prosthetics legs
Ann M. Simon , et al.[40]**2013**	MAV, ZC SSC, WFL	Powered prosthesis control during non-weight bearing activities
Aaron J. Young, et al.[41] **2016**	MAV, ZC SSC, WFL 2 ARC	Classifying the Intent of Novel Users during human locomotion
Eva Lendaro et al. [17] **2016**	MAV, ZC SSC, WFL	Potential Treatment for Phantom Limb Pain
J A Spanias, et al. [44]**2018**	MAV, ZC SSC, WFL 6 ARC	Online adaptive neural control of a robotic lower limb prosthesis
Aaron J. Young, et al.[45]**2014**	MAV, ZC SSC, WFL 2 ARC	Enhance intent recognition in powered lower limb prostheses
Deepak Joshi, et al. [28] **2016**	MAV, ZC SSC, WFL RMS, 3 ARC	Terrain and Direction Classification of Locomotion Transitions
D.C. Tkach and L.J. Hargrove [46] **2013**	MAV, ZC SSC, WFL	Predicting Ambulation Mode Transitions for Trans-Tibial Amputees
Jason D. Miller, et al. [13]**2013**	MAV, ZC SSC, WFL VAR	Myoelectric Walking Mode Classification for Transtibial Amputees
Deok-Hwan Kim, et al [12] **2014**	MAV, ZC SSC, WFL VAR	Real-Time Locomotion Mode Recognition

Table 3.3: Review of the literature: Features choice

The four most used features are: the mean absolute value, the zero crossing, the slope sign

changes and the waveform length. However, the zero crossing is irrelevant for this study because a rectified signal is used. Hence, the variance will be computed instead of the zero crossing.

For each channel, the four chosen features are calculated over one window as follows.

1. **Mean absolute value**: The Mean Absolute Value (MAV) is a method of detecting and gauging muscle contraction levels.

$$MAV = \frac{1}{N}\sum_{k=1}^{N}|x_k|$$

   - x_k = k$^{\text{th}}$ sample of the window
   - N = number of samples in the window

2. **Slope sign changes**: Slope sign change is related to signal frequency and is defined as the number of times that the slope of the EMG waveform changes sign within an analysis window. The slope sign changes counter, SSC, is incremented as follows :
   - if $x_k > x_{k-1}$ and $x_k > x_{k+1}$
   - or if $x_k < x_{k-1}$ and $x_k < x_{k+1}$

3. **Waveform length** : The waveform length is defined as the cumulative length of the waveform over a time segment. This feature provides a measure of the complexity of the signal.

$$WFL = \sum_{k=1}^{N}|\Delta x_k|$$

   - with $\Delta x_k = x_{k+1} - x_k$

4. **Variance**: The variance expresses the power of the EMG signal.

$$VAR = \frac{1}{N-1}\sum_{k=1}^{N}x_k^2$$

   - x_k = k$^{\text{th}}$ sample of the window
   - N = number of samples in the window

The MAV, the VAR and the WFL features are illustrated in figure 3.12 via an EMG signal recorded on the sartorius during walking. The features are calculated over a time window of 150 [ms]. These three features are all expressed in volts. The SSC, on the other hand is a counter and is illustrated in figure 3.13 with a different y axis.

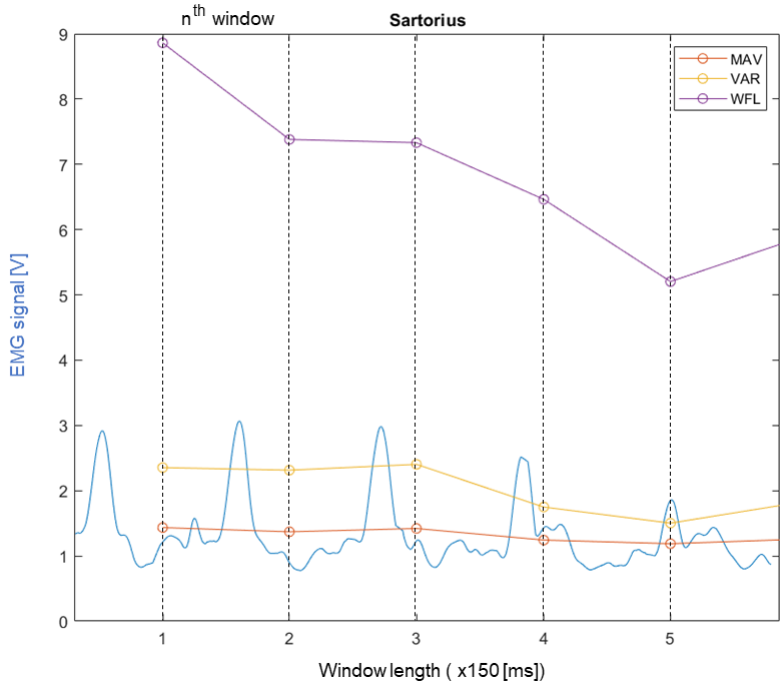


Figure 3.12: Illustration of the MAV, the VAR and the WFL computed over several time-windows for an EMG signal recorded on the sartorius during walking

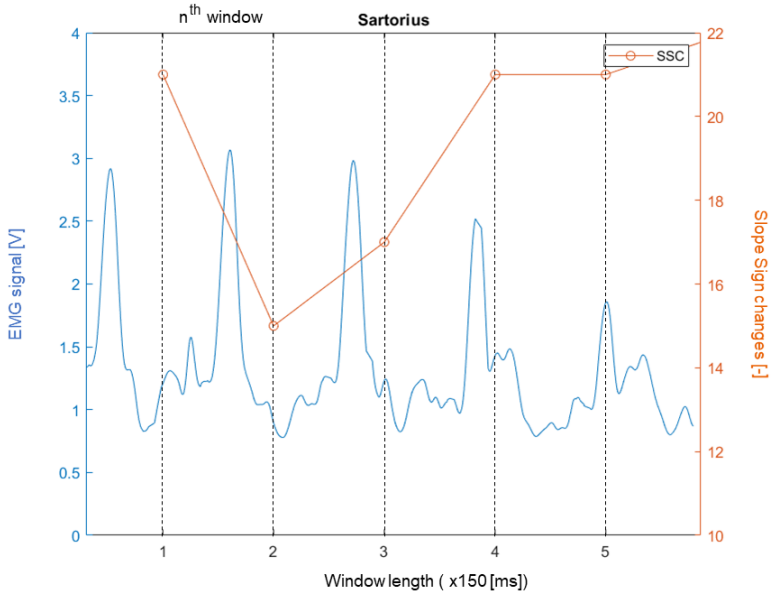


Figure 3.13: Illustration of the SSC computed over several time-windows for an EMG signal recorded on the sartorius during walking

In the ROS nodes structure, these four features determine a feature vector for each

channel i :

$$\underline{f_{ch_i}} = [MAV, SSC, WFL, VAR]$$

This vector is calculated by the service node features_service.cpp node as shown in figure 3.9 (code available in appendix A.1). The service requires a window of the channel number i obtained at the end of the sliding window algorithm, and it returns the vector $\underline{f_{ch_i}}$. The next step is to get one big features vector including the four features of the 8 channels over one window, that is:

$$\underline{F_{win}} = [\underline{f_{ch_1}}, \underline{f_{ch_2}}, \underline{f_{ch_3}}, \underline{f_{ch_4}}, \underline{f_{ch_5}}, \underline{f_{ch_6}}, \underline{f_{ch_7}}, \underline{f_{ch_8}}]$$

$\underline{F_{win}}$ is a vector of size 32 with all the features of the w$^{\text{th}}$ window and is the one used for the classification of the signal.

3.3 Classification algorithm

Once the features are extracted, it is then possible to classify the signal based on these features. Different classification methods exist for pattern recognition of EMG signals. Hence, this section begins by a review of the literature in order to choose which classifier will be used. It then explains the theory behind the chosen classifier and ends by presenting the implementation of the classification in the system.

3.3.1 Choice of the classifier

Here follows some of the most common method used for pattern recognition:

- Linear discriminant analysis (LDA)
- Quadratic discriminant analysis (QDA)
- Dynamic Bayesian network (DBN)
- Support vector machine (SVM)
- Artificial neural network (ANN)

As shown in table 3.4, the LDA is commonly used for pattern recognition and classification. According to Jason D. Miller et al.[13], there is no increase in accuracy using a SVM compared to a LDA. In addition, another research by E. Scheme et al.[52] states that while the LDA is a simple classifier it has similar performance to nonlinear and more sophisticated classifiers. Hence, the LDA is often chosen because it provides good accuracy and is easily implemented. Therefore the chosen classifier for this thesis will be the LDA.

Table 3.4 - Review of the literature : Classifier choice

Articles	**Classifier**	**Application**
Aaron J. Young , et al.[38]**2013**	DBN	Intent recognition strategy for thansfemoral amputee ambulation

Articles	Classifier	Application
He Huang , et al. [39]**2011**	SVM	Continuous Locomotion-Mode Identification for prosthetics legs
Ann M. Simon , et al.[40]**2013**	LDA	Powered prosthesis control during non-weight bearing activities
Aaron J. Young, et al.[41]**2016**	LDA	Classifying the Intent of Novel Users during human locomotion
Eva Lendaro et al. [17]**2016**	LDA	Potential Treatment for Phantom Limb Pain
J A Spanias, et al. [44]**2018**	DBN	Online adaptive neural control of a robotic lower limb prosthesis
Deepak Joshi, et al. [28]**2016**	LDA SVM	Terrain and Direction Classification of Locomotion Transitions
D.C. Tkach and L.J. Hargrove [46] **2013**	LDA	Predicting Ambulation Mode Transitions for Trans-Tibial Amputees
Angkoon Phinyomark, et al. [47]**2013**	LDA	EMG feature evaluation

Table 3.4: Review of the literature : Classifier choice

3.3.2 Linear Discriminant Analysis LDA

Here follows some explanations of the LDA from different sources :

- From Wikipedia[53] : The LDA is a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterises or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.
- From Sebastian Raschka [54]: Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order to avoid overfitting (“curse of dimensionality”) and also reduce computational costs.
- From Jason Brownlee [55] : It consists of statistical properties of your data, calculated for each class. For a single input variable (x) this is the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix. These statistical properties are estimated from your data and plug into the LDA equation to make predictions.

The development to determine the equations of the LDA problem [56] is presented here-after.
Setting the following variables :

- $\underline{C}$ = vector of the different tasks to perform which are referenced as classes = $[C_1, C_2, ..., C_N]$; $n \in [1, N]$
- N = total number of different tasks to classify
- $\underline{F}_w$ = the feature vector obtained at the end of section 3.2.2. To simplify the writing and because the analysis applies to only one window, the w subscript is dropped : $\underline{F}_w \rightarrow \underline{F}$

A discriminant analysis, DA, consists of maximising the posterior probability $P(C_n|\underline{F})$. That is, finding the estimation of the class as follows :

$$\tilde{C}_n = \text{arg (max [ln } P(C_n|\underline{F})]) \tag{3.1}$$

The probabilities are defined as follows :

– $P(C_n|\underline{F})$ = the probability that the signal is of class n given the feature vector $\underline{F}$.
– $P(\underline{F}|C_n)$ = the probability to have the feature vector $\underline{F}$ given that we are in class n.

These two are linked by Bayes' theorem :

$$P(C_n|\underline{F}) = \frac{P(\underline{F}|C_n)P(C_n)}{P(\underline{F})} \tag{3.2}$$

It is supposed that $P(C_n) = \frac{1}{N} = \pi_n$ is known exactly and it is assumed that $\underline{F}$ has a multivariate normal (MVN) distribution:

– $P(\underline{F}|C_n) \sim MVN(\underline{\mu}_n, \underline{\underline{\Sigma}}_n)$
– Its density is : $g_n(\underline{f}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\underline{\underline{\Sigma}}_n|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{f}-\underline{\mu}_n)^T \underline{\underline{\Sigma}}^{-1}(\underline{f}-\underline{\mu}_n)}$
– $\underline{\mu}_n$ = mean vector of class n
– $\underline{\underline{\Sigma}}_n$ = covariance matrix of class n

Bayes' rule becomes :

$$P(C_n = n|\underline{F} = \underline{f}) = \frac{g_n(\underline{f})\pi_n}{P(\underline{F} = \underline{f})} \tag{3.3}$$

Because the denominator does not depends on n, it can be written as a constant D :

$$P(C_n = n|\underline{F} = \underline{f}) = D * g_n(\underline{f})\pi_n \tag{3.4}$$

Expanding $g_n(\underline{f})$:

$$P(C_n = n|\underline{F} = \underline{f}) = \frac{D\pi_n}{(2\pi)^{\frac{p}{2}}|\underline{\underline{\Sigma}}_n|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{f}-\underline{\mu}_n)^T \underline{\underline{\Sigma}}_n^{-1}(\underline{f}-\underline{\mu}_n)} \tag{3.5}$$

The LDA assumes that the covariance matrix is the same for every class that is $\underline{\underline{\Sigma}}_n = \underline{\underline{\Sigma}}$. Absorbing now everything that does not depend on n in a constant D' equation 3.5 becomes :

$$P(C_n = n|\underline{F} = \underline{f}) = D'\pi_n e^{-\frac{1}{2}(\underline{f}-\underline{\mu}_n)^T \underline{\underline{\Sigma}}_n^{-1}(\underline{f}-\underline{\mu}_n)} \tag{3.6}$$

Taking the logarithm of both sides :

$$\ln\,[P(C_n = n|\underline{F}) = \underline{f}] = \ln\, D' + \ln\, \pi_n - \frac{1}{2}(\underline{f} - \underline{\mu}_n)^T \underline{\underline{\Sigma}}^{-1}(\underline{f} - \underline{\mu}_n) \tag{3.7}$$

The first term does not depend on n so it is the same for every class and can be ignored.

$$\begin{aligned} \ln\,[P(C_n = n|\underline{F} = \underline{f})] &= \ln\, \pi_n - \frac{1}{2}(\underline{f} - \underline{\mu}_n)^T \underline{\underline{\Sigma}}^{-1}(\underline{f} - \underline{\mu}_n) \\ &= D'' + \ln\, \pi_n - \frac{1}{2}\underline{\mu}_n^T \underline{\underline{\Sigma}}^{-1}\underline{\mu}_n + \underline{f}^T \underline{\underline{\Sigma}}^{-1}\underline{\mu}_n \end{aligned} \tag{3.8}$$

The goal is then to find the class n that maximizes $\delta_n(\underline{f})$. That is, in equation 3.1 :

$$\tilde{C}_n = \arg\,[\max\, \delta_n(\underline{f})]$$

$$\delta_n(\underline{f}) = \ln\, \pi_n - \frac{1}{2}\underline{\mu}_n^T \underline{\underline{\Sigma}}^{-1}\underline{\mu}_n + \underline{f}^T \underline{\underline{\Sigma}}^{-1}\underline{\mu}_n \tag{3.9}$$

The mean vector and the covariance matrix are defined during a training phase. Several training samples can be recorded for one class, this number of training observations will be designed by k. Each observation of class n gives a feature vector for class n designed by $\underline{f}_{C_n}$ and the number of observations made in class n is designed by K_n. These variables being set, the mean vector and covariance matrix are calculated as follows.

$$\underline{\mu}_n = \frac{1}{K_n}\sum_{k=1}^{K_n} \underline{f}_{C_n,k} \tag{3.10}$$

$$\underline{\underline{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}[\frac{1}{K_n - 1}(\underline{\underline{F}}_n - \underline{\underline{M}}_n)(\underline{\underline{F}}_n - \underline{\underline{M}}_n)^T] \tag{3.11}$$

- $\underline{\underline{F}}_n$ = training feature matrix for class n and is of size [n_features x n_training_samples].
- $\underline{\underline{M}}_n$ = training mean matrix for class n = $[\mu_n, \mu_n, \mu_n, ...]$ and is of same dimension as $\underline{\underline{F}}_n$.

Linear Discriminant Analysis Equations

The prediction of a class n, that is the prediction of the performed task, is defined by the following equations:

$$\tilde{C}_n = \arg\,[\max\,\delta_n(\underline{f})]$$

$$\delta_n(\underline{f}) = \ln\,\pi_n - \frac{1}{2}\underline{\mu}_n^T\underline{\underline{\Sigma}}^{-1}\underline{\mu}_n + \underline{f}^T\underline{\underline{\Sigma}}^{-1}\underline{\mu}_n$$

The mean vector $\underline{\mu}_n$ and the covariance matrix $\underline{\underline{\sum}}$ being computed during a training phase.

This development reveals the significance of the training phase. Indeed the classifier needs to be trained before it can be tested. The training is done the same way as the testing : based on features extracted from time-analysis windows. At first the decisions taken at the end of each window will be used to train the classifier,afterwards, test samples of the different tasks can be predicted.

3.3.3 Scikit-learn Pyhton library

In order to implement the LDA on the microprocessor, a python library called scikit-learn [57] is used. This library provides a LDA class and several functions to perform the needed operations. That is why the ROS nodes (LDA_service and score_service) using the scikit-learn (sklearn) library had to be implemented in python as shown in figure 3.14 and 3.15.

The three functions used out of the library are the following :

1. fit(X,y) : fits the trained samples stored in X to the true tasks labels stored in y.
2. predict(X) : predicts the task label for the tested samples stored in X according to a fitted classifier.
3. score (X,y): returns a score based on the prediction accuracy of the samples stored in X, knowing the true tasks labels of these samples stored in y.

The complete code of the LDA_service.py node can be consulted in the appendix A.3. Figure 3.14 is an overview of the algorithm followed by the code.
Once the features have been computed and stored in a vector as explained in section 3.2.2, the LDA_service can be called. If the system is in training mode, the trained samples are send to the the LDA_service node which fits the LDA classifier using the fit() function of the sklearn library. The true tasks labels of the training samples are defined by the system knowing how much different tasks are performed and how many samples per task were taken.
Once the system changes to the testing mode, a call to the LDA_service node returns a predicted task label of the current tested sample using the predict() function provided by the sklearn library. In the end, all the predicted tasks of all the testing samples are stored in a vector.

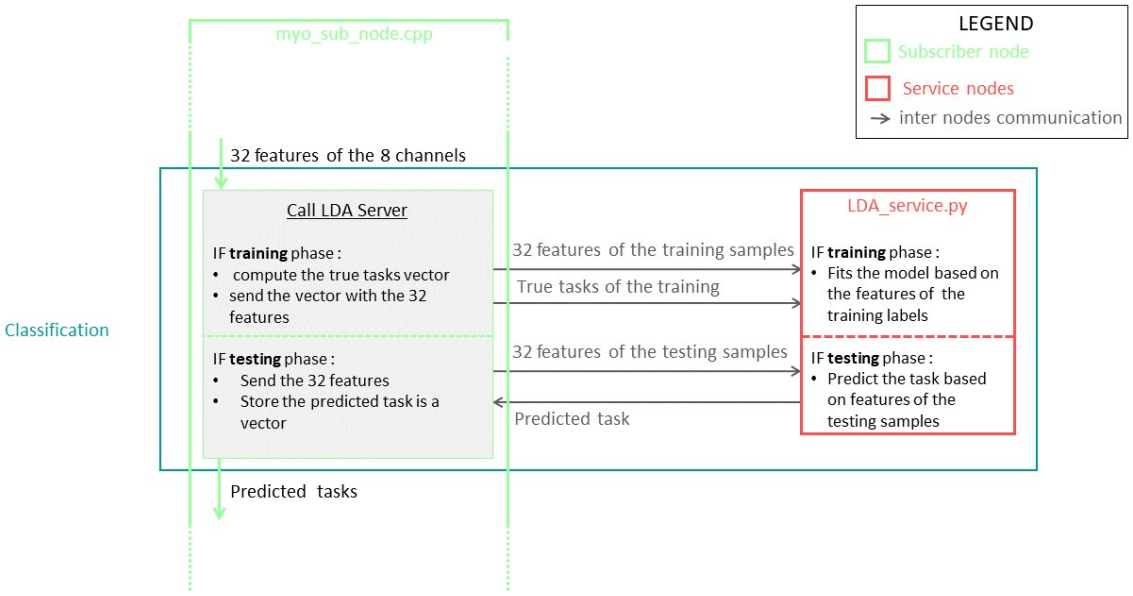


Figure 3.14: Zoom into the classification part of the ROS nodes structure shown in figure 3.8

This way, at the end of the experiment the score_service.py shown in figure 3.15 is called and returns a score using the score() function. The true task labels of the testing samples are introduced by the user.

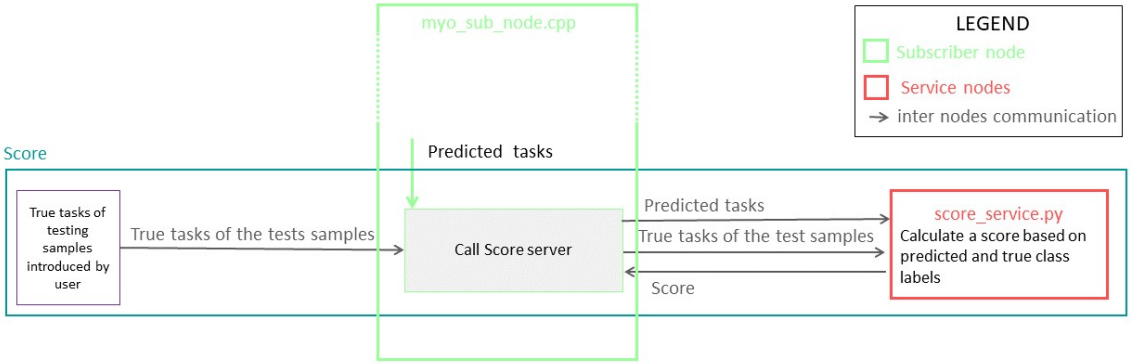


Figure 3.15: Zoom into the score part of the ROS nodes structure shown in figure 3.8

3.4 System Evaluation

This section presents the methods used to evaluate the performance of the system. As proposed by the article of He Huang et al.[39], two types of errors are differentiated:

- **Steady-state errors** The errors that occur in steady-state mode, that is, when the classifier misclassifies one task during a running task.

- **Transitional errors** The errors that occur during transition mode, that is, when the classifier misclassifies one task during a transition from one task to another.

The next subsections explain how to calculate these errors.

3.4.1 Steady-state errors

Two parameters are used to evaluate the errors in steady-state mode, the classification accuracy and the confusion matrix.

1. The **classification accuracy**, CA, is the number of correctly classified tested samples over the total number of tested samples.

$$\text{CA} = \frac{\text{\# of correctly classified tested samples}}{\text{\# of total tested samples}}$$

2. The **confusion matrix**, M, is the relationship between the targeted tasks and the predicted tasks. It is a square matrix of dimension [TxT], T being the number of target tasks.

$$M = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1T} \\ a_{21} & a_{22} & \dots & a_{2T} \\ a_{31} & a_{32} & \dots & a_{3T} \\ \dots & \dots & \dots & \dots \\ a_{T1} & a_{T2} & \dots & a_{TT} \end{vmatrix}$$

$$a_{ij} = \frac{\text{\# of tested samples} \in \text{task i predicted as task j}}{\text{\# of total tested samples} \in \text{task i}}$$

Therefore, if the classifier is perfect without any errors, the confusion matrix would be a diagonal matrix. An example of a perfect imaginary experiment with four different tasks to differentiate is shown in figure 3.16.

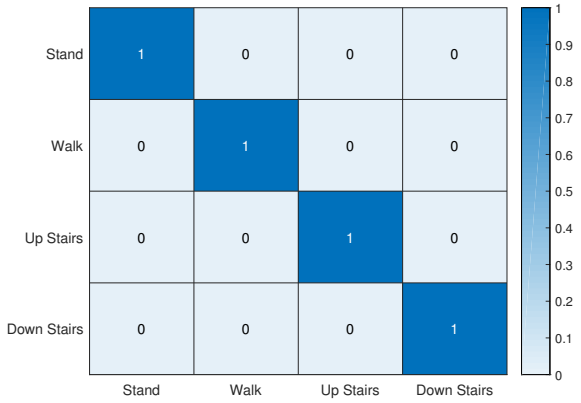


Figure 3.16: Example of a perfect confusion matrix for 4 different performed tasks

3.4.2 Transitional errors

Two parameters are used to evaluate the transitional mode.

1. The **transitional accuracy** is the number of correctly classified tested samples occurring at a transition :

$$\text{TA} = \frac{\text{\# of well classified transitional samples}}{\text{\# of total transitional samples}}$$

2. The **transitional time** is the difference between the prediction and the observation time. The time zero corresponds to the toe-off of the leading leg. The observation time is the time the user writes down when he observes the subject making the transition step (at the toe-off). The prediction time, that is the time at which the predicted transition occurs, is assumed to be the time between the two decisions that mark the transition. For instance, if the five first decisions predict task 1 and the five next decisions predict task 2, the predicted transition time is assumed to be the time equally between the fifth and the sixth decision.
   Hence, a positive value means that the system is in advance with respect to the observed movement.

3.4.3 Features graphs

The system has one classifier which is based on the 32 features computed from the eight muscles. Because it does not have eight different classifiers, one for each muscle, it can not be precisely known which muscle classifies the best. However, the tendency of a muscle to induce errors could be determined by analysing the features graphs. Indeed, in one feature graph, as shown in figure 3.17, the different tasks are compared and per task, the trained and tested samples are shown. It means that if the two graphs representing the tested and trained data overlap, it is likely that the taken decisions helped the classifier to predict the true task. If, on the other hand, for one decision, the tested sample of one task is more close to the trained sample of another class, then, according to this decision, it is likely that the classifier will misclassify the tested task with the closer trained task.

For instance, figure 3.18 shows the mean absolute value feature (MAV) for the sartorius (SAR), the gracilis (GRA) and the adductor magnus (AM). If the stair descent (SD) task is analysed (in green on the graphs), according to the MAV of the SAR, it is likely that all the decisions will predict correctly the SD except for the last one where the SD could be confused with the walking (WA). According to the GRA, the 1st, 2nd, 3rd and 6th decision could confuse SD with WA, as for the AM the two last decisions could again misclassify the SD as WA.

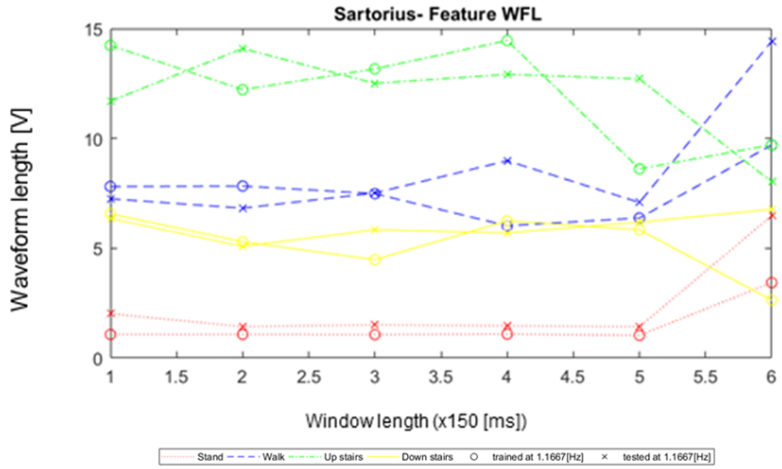


Figure 3.17: Example of a feature graph : Waveform length computed for the sartorius EMG signals

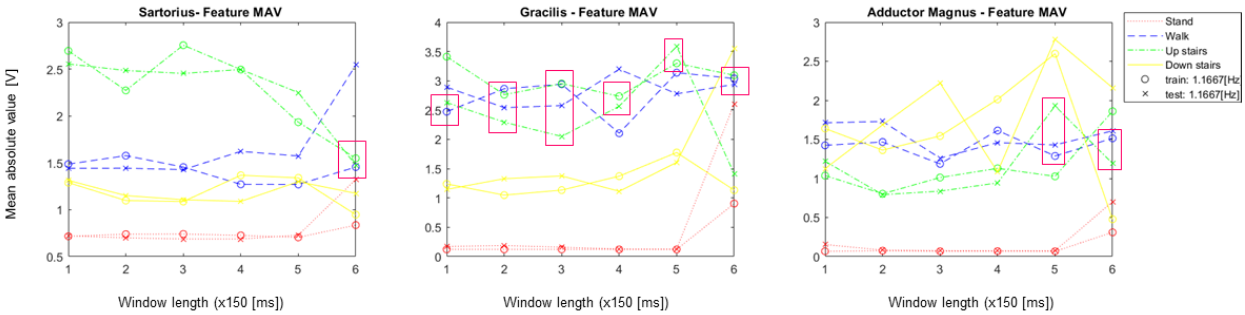


Figure 3.18: Illustration of a feature graph reading

This way, the possible errors made by the classifier can be computed and an idea of the muscles efficiency can be obtained.

3.5 Experimental Protocol

This section introduces to the followed protocol to conduct the experiments. It begins with the devices placement recommendations followed by an explanation of the circuits with the training and testing phase. It ends with the presentation of the varied parameters during the experiments. As a reminder, the purpose is to differentiate six different locomotion modes: level-ground Walking (WA), still standing (ST), stair ascent (SA), stair descent(SD), ramp ascent (RA) and ramp descent (RD).

3.5.1 Devices placement

The first thing that have to be done is the electrodes placement on the subject. The eight MyoWare's^{TM} are placed on the eight chosen muscles of the thigh : The rectus femoris, the vastus lateralis, the biceps femoris, the semitendinosus, the sartorius, the adductor magnus, the tensor fascia latae and the gracilis. This choice has been justified in section 2.2.

Here follows some recommendations for a good placement of the MyoWare's^{TM} [33]:

1. Clean the intended area with soap
2. Snap the electrodes to the sensor's snap connectors as shown in figure 3.19
3. Place the sensor on the desired muscle so one of the connected electrodes is in the middle of the muscle body. The other electrode should line up in the direction of the muscle length.
4. Place the reference electrode on a bony or nonadjacent muscular part of the body near the targeted muscles

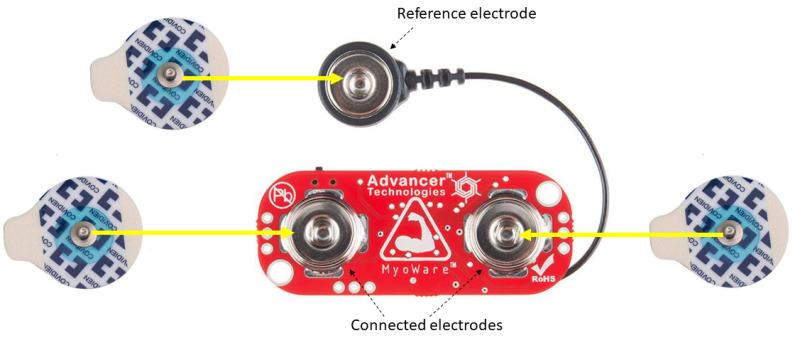


Figure 3.19: Illustration of the electrodes snapped to the connectors of a MyoWare^{TM}

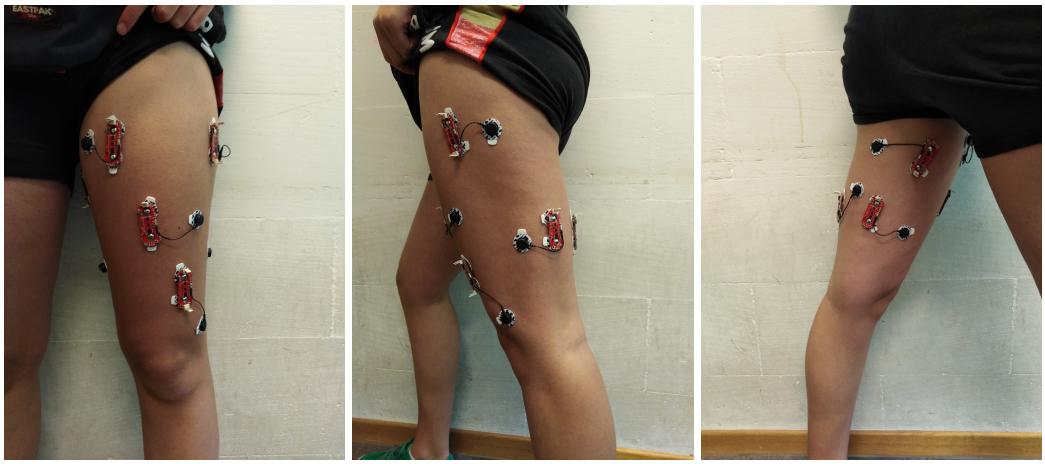


Figure 3.20: Pictures of a subject with correctly placed electrodes

The placement has to be done correctly to avoid to much noise and cross talking as was shown in figure 3.3. However, finding the middle of the good muscles is not an easy task. In appendix B, some guidelines to find and palpate the eight thigh muscles are proposed. But it is better if a specialist, for instance a physiotherapist, helps palpate the muscles. Figure 3.20 shows a subject with the MyoWare's^{TM} correctly placed.

Once the electrodes are placed, the Raspberry Pi 3B+ along with the power bank are put in a bumbag that the subject carries around the waist. The MyoWare's^{TM} can then be wired to the ADC. Figure 3.21 shows a subject ready for an experiment.

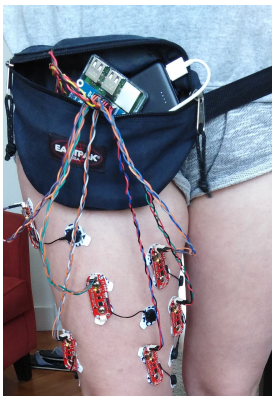


Figure 3.21: Picture of a subject ready for an experiment

3.5.2 Varied Parameters

In the conducted experiments the following parameters were changed:

1. The **subjects** : when a change of subject is made, the results can differ due to the different morphology of the different subjects but also due to the placement of the electrodes which is made by a different person for each subject. These are the informations about these subjects :
   - Subject 1 : Female aged 21. The surface electrodes were placed by a physiotherapist member of the *Laboratoire d'anaylse du mouvements, Cliniques Universitaires Saint-Luc*.
   - Subject 2 : Female aged 22. The surface electrodes were placed by a non-specialist with the help of the guidelines presented in appendix B.
   - Subject 3 : Male aged 18. The electrodes were placed by a physiotherapist graduated in June 2019.
2. The **gait cadence**: The subject had to perform the different task following the cadence set by a metronome. three different cadences were used :
   - 0.8333 [Hz] : slow
   - 1.1667 [Hz]: normal
   - 1.5 [Hz]: fast
3. The **number of muscles** involved : the number of muscles for which the activity was recorded varied from four to eight. Starting with the 8 muscles that were chosen

in section 2.2 , the biceps femoris and the vastus lateralis are then removed from the measurements. This choice is made because these two muscles are part of a group muscle , the hamstrings and the quadriceps respectively. Indeed, the RF and the VL are part of the quadriceps and hence are responsible for the same actions as shown on figure 3.22. The same applies for the biceps femoris and the semitendinosus. The goal of this experiment is to see if only one muscle of the same muscles group is sufficient to detect the different tasks. Regarding the experiment with four muscles, the tensor fascia latae and the gracilis were removed to consider the influence of the hip adduction and abduction as shown in figure 3.23.

- 8 muscles : Rectus femoris, Vastus lateralis, Biceps femoris, Semitendinosus, Sartorius, Tensor fascia latae, Gracilis and the Adductor magnus.
- 6 muscles : Rectus femoris, Semitendinosus, Sartorius, Tensor fascia latae, Gracilis and the Adductor magnus.
- 4 muscles : Rectus femoris, Semitendinosus, Sartorius and Adductor magnus.

Knee Extensors	**Hip Extensors**	**Hip Abductors**
• Rectus Femoris • Vastus Lateralis • Vastus Medialis — Quadriceps	• Gluteus Maximus • Biceps Femoris • Semitendinosus • Semimembranosus	• Gluteus Maximus • Tensor Fascia Latae
Knee Flexors	**Hip Flexors**	**Hip Adductors**
• Biceps Femoris • Semitendinosus • Semimembranosus — Hamstrings • Sartorius (aids) • Gracilis (aids)	• Rectus Femoris • Vastus Lateralis • Vastus Medialis • Sartorius	• Adductor Magnus • Gracilis

Figure 3.22: Choice of the muscles based on their function for an experiment with 6 recorded muscles

Knee Extensors	**Hip Extensors**	**Hip Abductors**
• Rectus Femoris • Vastus Lateralis • Vastus Medialis — Quadriceps	• Gluteus Maximus • Biceps Femoris • Semitendinosus • Semimembranosus	• Gluteus Maximus • Tensor Fascia Latae
Knee Flexors	**Hip Flexors**	**Hip Adductors**
• Biceps Femoris • Semitendinosus • Semimembranosus — Hamstrings • Sartorius (aids) • Gracilis (aids)	• Rectus Femoris • Vastus Lateralis • Vastus Medialis • Sartorius	• Adductor Magnus • Gracilis

Figure 3.23: Choice of the muscles based on their function for an experiment with 4 recorded muscles

4. The **number of tasks**: At first, four tasks with patterns significantly different from

each other are differentiated: still standing, level-ground walking, stair ascent and descent. Afterwards, ramp ascent and ramp descent are added. Indeed, their patterns will be more similar to walking for light inclinations and more similar to stair ascent and descent for strong inclinations. To analyse the possible difference in accuracy between theses two cases, some experiments are conducted with 4 tasks and others with 6 tasks to differentiate.

- 4 tasks : ST, WA, SA, SD
- 6 tasks : ST, WA, SA, SD, RA, RD

3.5.3 Circuits

In order to classify the different tasks, one experiment must be divided is two phases : the training and the testing phase. Indeed, as explained in section 3.3, the LDA classifier used by the system needs to be trained.

3.5.3.1 Training phase

During the training, the features calculated over a time-window are used to build the data set of the task that is being performed. The system knows which tasks is trained when and in which order. Hence, for each task, a set of training samples will be assigned.

In order to assure a good training of the tasks, the number of performed gait cycles per task must be determined. To have a good capture of the movement this number is set to 5. If the cadence at which the subject is moving is known, an estimation of the needed recording time for one task can be done.
This time amount must be expressed in a window length in order to be introduced into the system and besides, one window length correspond to one taken decision. So the question is to find the number of decisions taken during 5 performed gait cycles. However, the cadence of the subject is one of the varied parameter during the experiments. In order to have a fixed value to input in the system, the slowest cadence is taken into consideration, 0.8333 [Hz], that is 5 cycles are performed in 12 [s]. As was explained in section 3.2.1, the decision stream rate is of 0.69 [Hz], that is one decision is taken in 1.45 [s] and hence 9 decisions have to taken to capture the 12 [s] of 5 gait cycle. To avoid the inaccuracies at the beginning and the end of the movement the first and last decisions are not taken into account for the classification.

The system is made such as the training of the different tasks does not have to be continuous. The subject can interrupt between each task in order, for instance, to move from the stairs to the ramp.

To sum up, the subject begins the experiment with a training phase. He/she begins by performing the first task and once the system has made 9 decisions, the system pauses and the subject can get ready for the next task to perform. The user send then a signal to the system to indicates that it can begin recording the next task and so on until all the tasks are trained.

3.5.3.2 Testing phase

Once the training phase is done, the system enters the testing mode. The features computed over each window are then used by the classifier to compare the recorded signals to the trained samples of each task and return a predicted task. As explained in section 3.4, the system is evaluated by determining two types of errors: the steady-state errors and the transitional errors. In order to easily evaluate these errors, two modes will be distinguished to conduct the experiments, the steady-state mode and the transitional mode.

Steady-state mode The goal of this mode is to determine the errors that occurs when the classifier misclassifies one task during a performed tested class. This errors take not into account the transitional period. Therefore, during the steady-state mode, the circuit is not performed in continuous. The system is paused between each tested task. This way, the transitional errors do not interfere with the steady-state errors.
This mode is very alike the training phase. Indeed, in the training phase, the system also pauses between each trained task. Moreover, the rule determined for the training phase that ensures to record 5 cycles to have a good capture of the signal can also be applied for the testing phase. That is why, to simplify the protocol, the system will wait for 9 decisions to be taken per task as was done for the training. Indeed, it allows the system to use the same algorithm for the training phase and the testing phase in steady-state mode. But instead of using the computed features over a window to fit the classifier, the features are used to predict the performed task.

At the end of the number of desired tests (one test states for one task performed in order to be tested), the system calculates a score for the experiment and provides two vectors with the targeted and predicted tasks to be able to calculate the confusion matrix off-line.

Transitional mode The purpose of this mode is to compute the transitional errors, that is, the misclassified tasks during a transition from one task to another. Therefore, a continuous circuit is predefined. Attention must be paid to the practical feasibility of this circuit. Once the training is done, the system launches a continuous recording and the subject executes the circuit. When the circuit is completed, the user can stop the system and it returns two vectors with the predicted tasks and the corresponding time of the decisions.

By comparing the actual performed tasks during the circuit and the predicted tasks returned by the system, the transitional errors can be computed. These experiments were completed at a fixed cadence of 1.1667 [Hz]. Figure 3.24 and 3.25 show the actual circuits performed for experiments conducted with subject 3 and 6 and 4 differentiated tasks. Figure 3.26 shows the circuit performed during an experiment with subject 1 and 4 different tasks to classify.

Figure 3.24: Picture of the circuit performed for a transitional experiment conducted with subject 3 and 6 different tasks

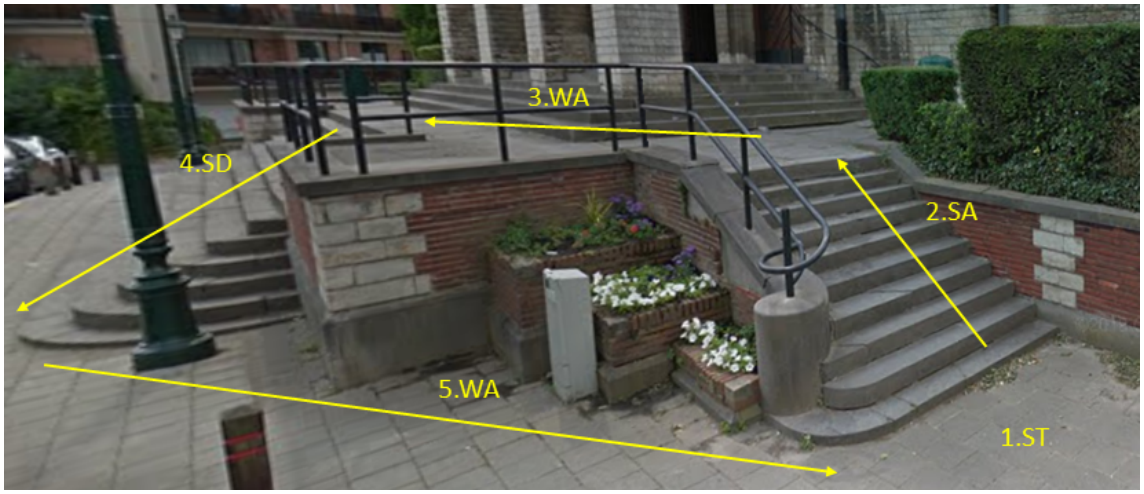


Figure 3.25: Picture of the circuit performed for a transitional experiment conducted with subject 3 and 4 different tasks

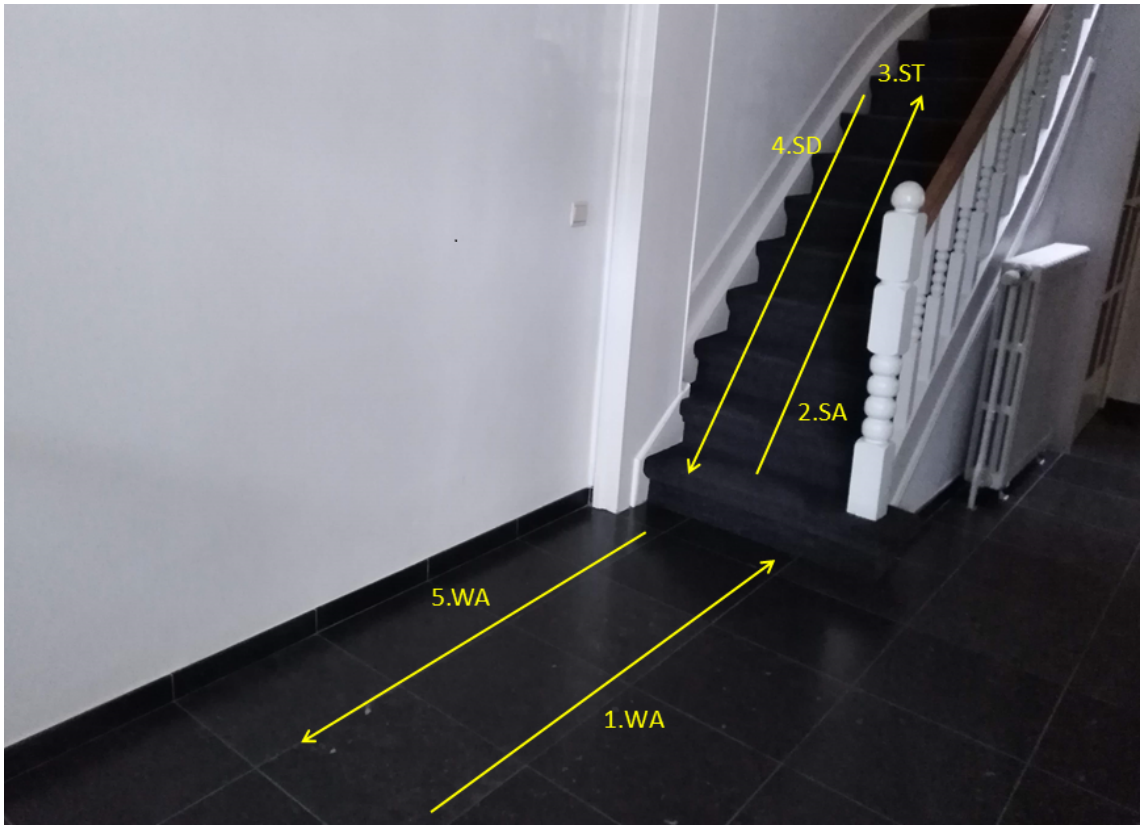


Figure 3.26: Picture of the circuit performed for a transitional experiment conducted with subject 1 and 4 different tasks

Chapter 4

Results

This chapter presents the results obtained after the conducted experiments. It is divided in two sections. The first one presents the results of the experiments performed to determine the steady-state error, that is, the error that occurs when the system misclassifies one task in the middle of a running task. The second section presents the results of the experiments conducted to determine the transitional error, that is, when the system misclassifies one task during a transition from one task to another. These two types of experiments are respectively called steady-state experiments and transitional experiments.

As a reminder from section 3.5.2, the varied parameters were:

1. The subjects: in this chapter, references will be made to subject 1, 2 and 3. The information about these subjects are the following:
   - Subject 1 : Female aged 21. The surface electrodes were placed by an experimented physiotherapist, member of the *Laboratoire d'anaylse du mouvement, Cliniques Universitaires Saint-Luc*.
   - Subject 2 : Female aged 22. The surface electrodes were placed by a non-specialist with the help of the guidelines presented in appendix B.
   - Subject 3 : Male aged 18. The electrodes were placed by a physiotherapist just graduated in June 2019.
2. The cadence of the gait:
   - 0.8333 [Hz] : slow
   - 1.1667 [Hz]: normal
   - 1.5 [Hz]: fast
3. The number of muscles involved :
   - 8 muscles : Rectus femoris (RF), Vastus lateralis (VL), Biceps femoris (BF), Semitendinous (SMT), Sartorius (SAR), Tensor fascia latae (TFL), Gracilis (GRA) and the Adductor magnus (AM).
   - 6 muscles : Rectus femoris, Semitendinous, Sartorius, Tensor fascia latae, Gracilis and the Adductor magnus.
   - 4 muscles: Rectus femoris, Semitendinous, Sartorius and Adductor magnus.

4. The number of tasks:

   – 4 tasks : Still standing (ST), level-ground Walking (WA), Stair ascent (SA), Stair descent (SD)
   – 6 tasks : Still standing (ST), level-ground Walking (WA), Stair ascent (SA), Stair descent (SD), Ramp ascent (RA) and ramp descent (RD).

4.1 Steady-state experiments

As explained in section 3.4, the system in steady-state mode is evaluated based on the classification accuracy and the confusion matrix. Moreover, the feature graphs can also be analysed to get an idea of the efficiency of the muscles and the features. The relevant graphs are shown in this chapter and the feature graphs of all the experiments can be consulted in appendix D.

The recorded EMG signals without processing are not directly used for the system evaluation. However it is still interesting to visualise the pattern of the EMG's. To obtain a smoother graph, an moving-average filter of 10 samples is applied on the recorded signal. As an example, the EMG signals of one experiment during the training of all the different tasks are shown in figure 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6. The EMG recordings for all the experiments are available in appendix C.

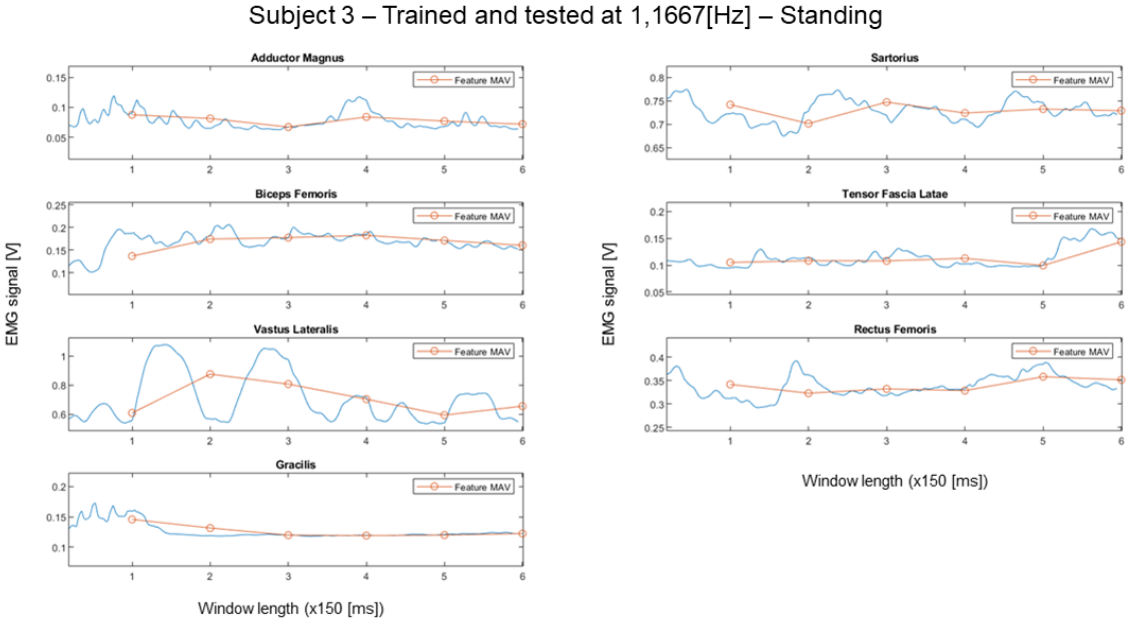


Figure 4.1: EMG signals of the different muscles recorded during the training phase of an experiment with subject 3 performing the standing task at a cadence of 1.1667[Hz]

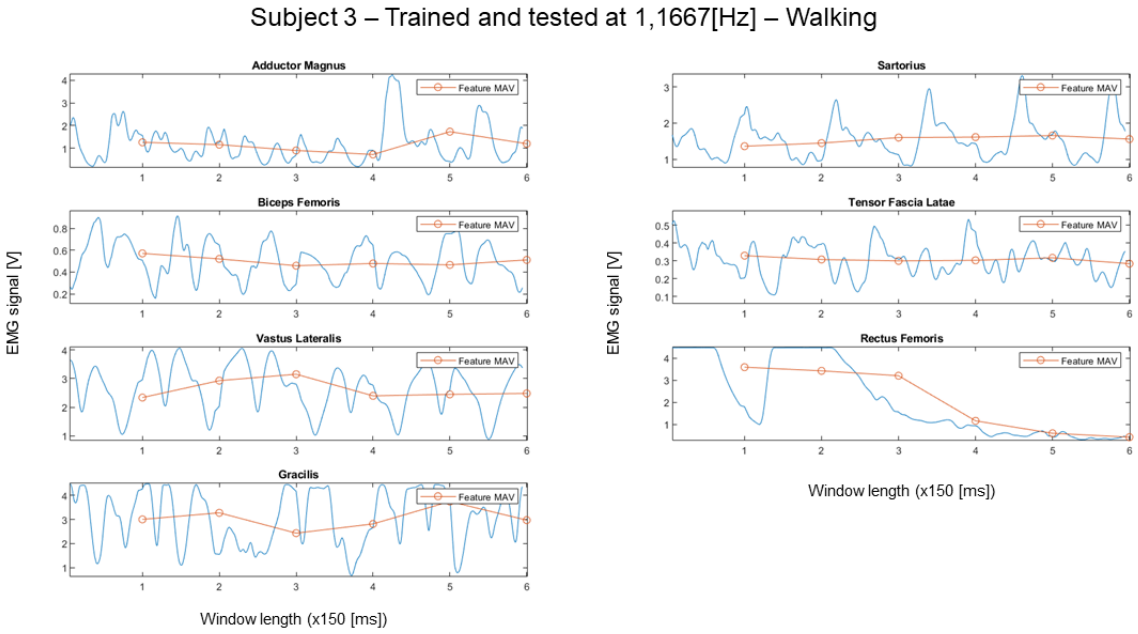


Figure 4.2: EMG signals of the different muscles recorded during the training phase of an experiment conducted with subject 3, performing the walking task at a cadence of 1.1667[Hz]

Figure 4.3: EMG signals of the different muscles recorded during the training phase of an experiment conducted with subject 3, performing the stair ascent task at a cadence of 1.1667[Hz]

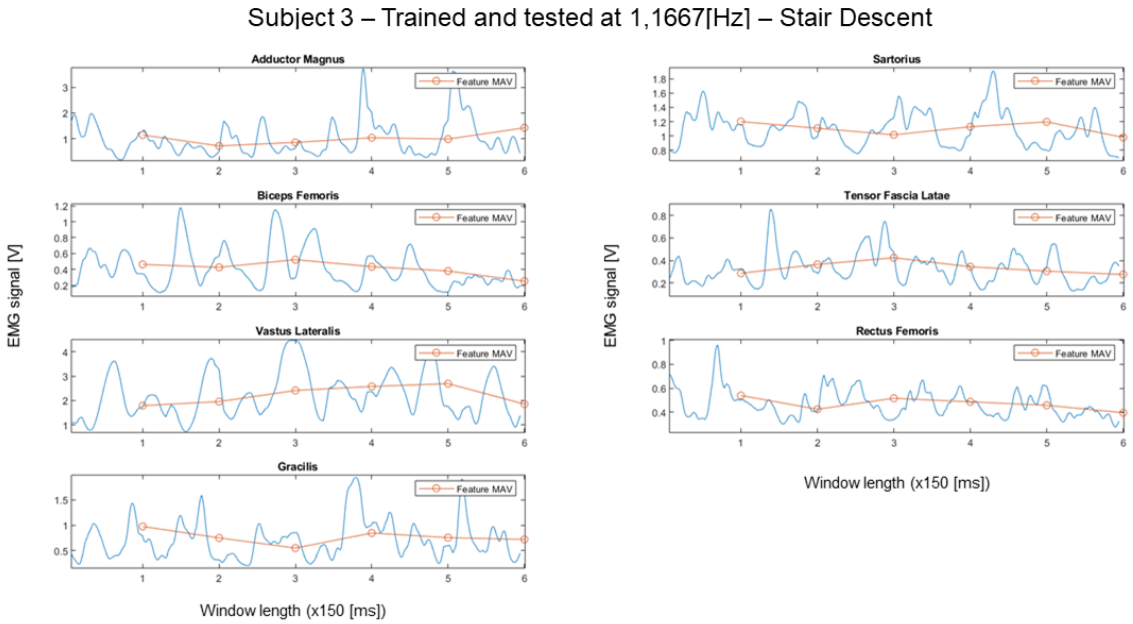


Figure 4.4: EMG signals of the different muscles recorded during the training phase of an experiment conducted with subject 3, performing the stair descent task at a cadence of 1.1667[Hz]

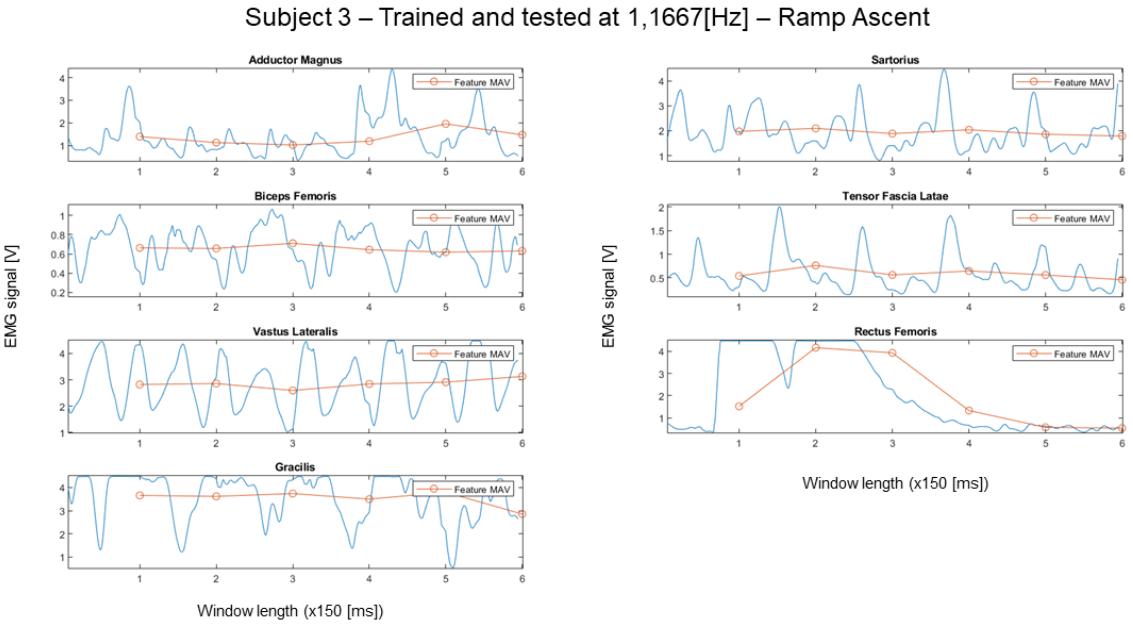


Figure 4.5: EMG signals of the different muscles recorded during the training phase of an experiment conducted with subject 3, performing the ramp ascent task at cadence of 1.1667[Hz]

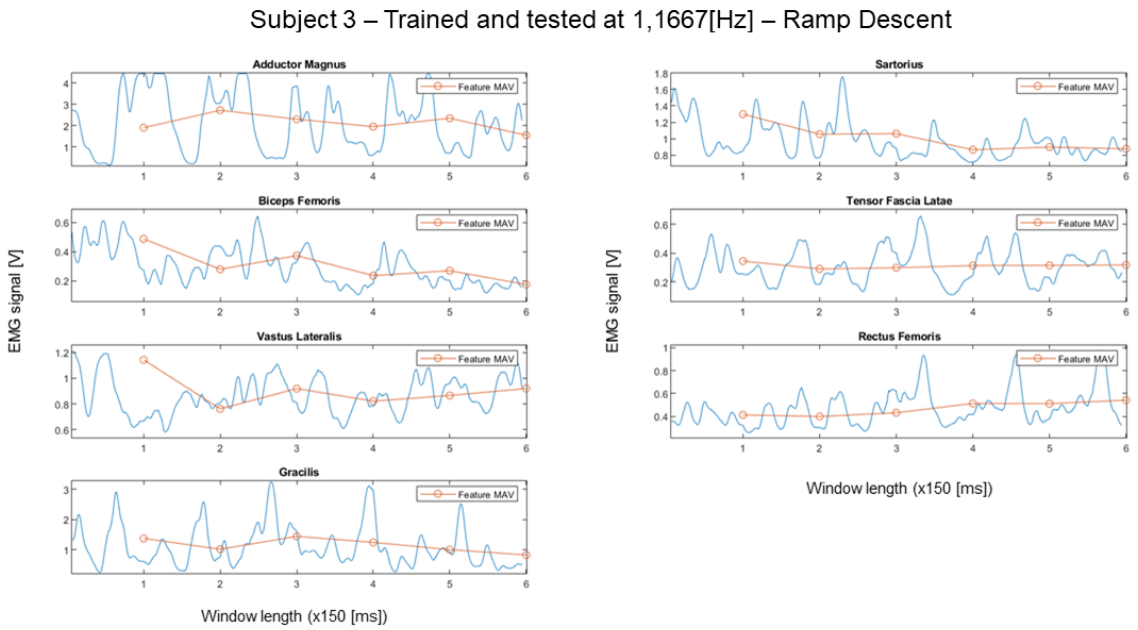


Figure 4.6: EMG signals of the different muscles recorded during the training phase of an experiment conducted with subject 3, performing the ramp descent task at a cadence of 1.1667[Hz]

4.1.1 Effect of the cadence variation

The experiments were conducted with two different subjects. With subject 1, the classifier was first trained at a slow cadence, 0.833 [Hz], and then tested at 0.833 [Hz], 1.1667 [Hz] and 1.5 [Hz]. Afterwards, the classifier was trained at a normal cadence, 1.1667 [Hz], and tested with the same cadence variation. With subject 2, the classier was trained and tested at 1.1667 [Hz] and afterwards, it was trained and tested at 1.5 [Hz].

Classification accuracy The classification accuracy of the experiments with subject 1 are shown in figure 4.7. The best scores are obtained when the testing is made at the same cadence as the training: 96.43% of corrected predicted tasks for the training and testing at 1.1667[Hz] and 83.93% for 0.833 [Hz]. There is a huge reduction of accuracy when the testing is done at a difference cadence as the training cadence. Reduction of about 34% and 37.5% when the testing was made at 0.833[Hz] and 1.5[Hz] respectively but trained at 1.1667 [Hz] and a reduction of about 12.5% and 37.5% when the testing was made at 1.1667[Hz] and 1.5[Hz] respectively but trained at 0.833 [Hz].

The goal of the experiments with subject 2 was to compare a fast cadence and a normal cadence of gait. As shown in figure 4.8, the best score is obtained for the normal cadence, 1.1667[Hz] with a classification accuracy of 92% and with a fast cadence there is an accuracy reduction of 12%.

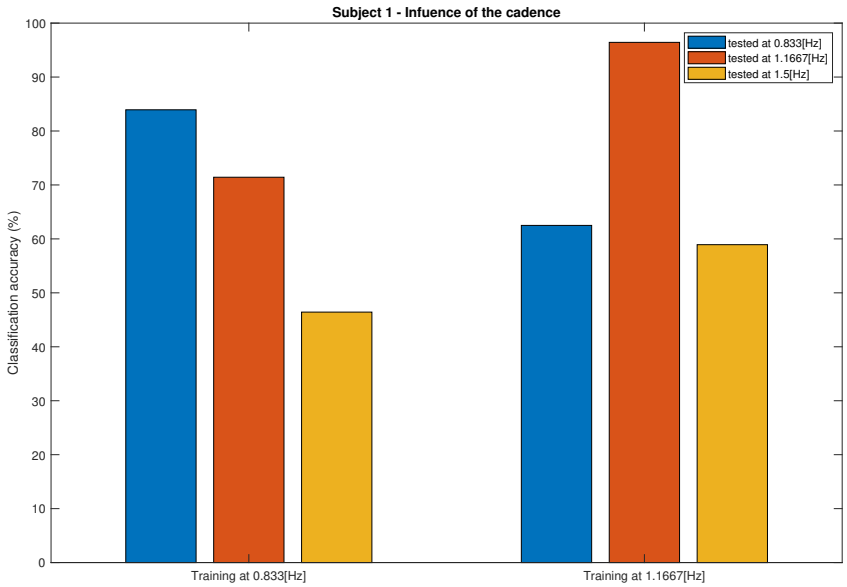


Figure 4.7: Classification accuracy obtained after the experiments conducted with subject 1 performing the 4 tasks at different cadences

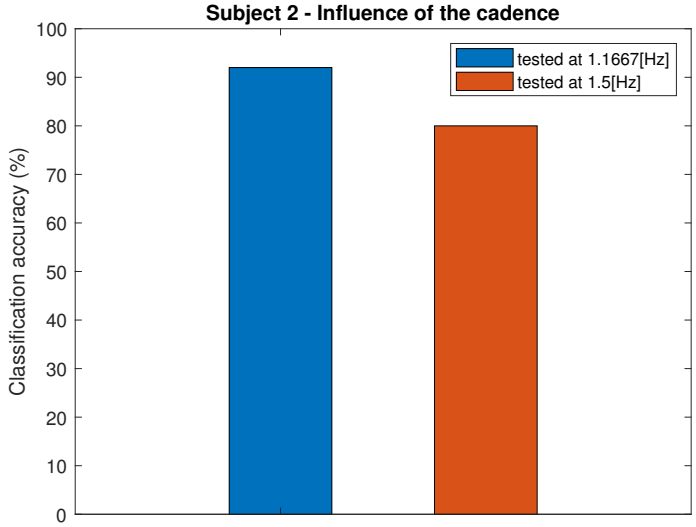


Figure 4.8: Classification accuracy obtained after the experiments conducted with subject 2 performing the 4 tasks at different cadences

Confusion matrices When analysing the confusion matrices, the confusion between the different tasks can be analysed. The confusion matrices of the experiments with subject 1 are shown in figure 4.9. For the experiments made at the same cadence during the training

and testing phase, which had the best scores, the little error is more disparate. However for the others experiments, the tasks the most misclassified are more clearly defined. For a training at 0.833 [Hz] combined with a testing at 1.1667 [Hz], 75% of the error is a confusion between the WA and SD tasks. The same tasks are confused with a training at 0.833 [Hz] and a testing at 1.5 [Hz]: 40% of the error is a confusion between WA and SD. For the experiments with a training at 1.1667 [Hz], 57.2% of the error is a confusion between SA and SD for the testing at 0.833 [Hz] and 56.5% of the error is the SD task misclassified as WA for the testing at 1.5 [Hz].

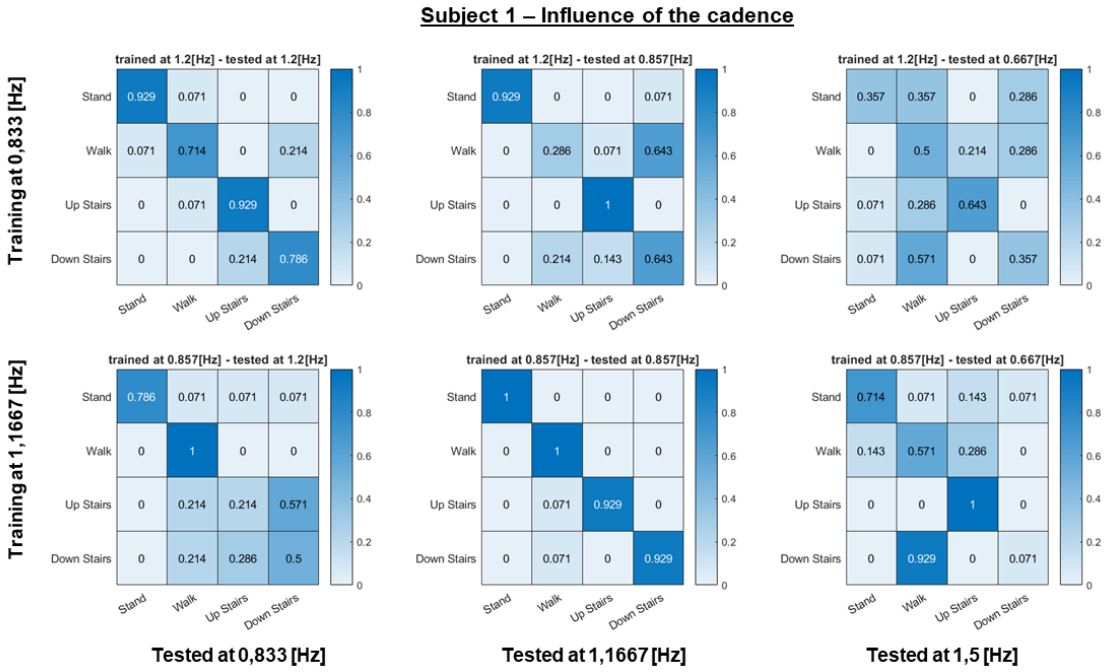


Figure 4.9: Confusion matrices obtained after the experiments conducted with subject 1 performing the 4 tasks at different cadences

During the experiment with subject 2 at a cadence of 1.1667 [Hz], the error occurs only for the classification of the WA task. And during the one with a fast cadence, 1.5 [Hz], 85.7% of the error is a confusion between the SA and the SD task.

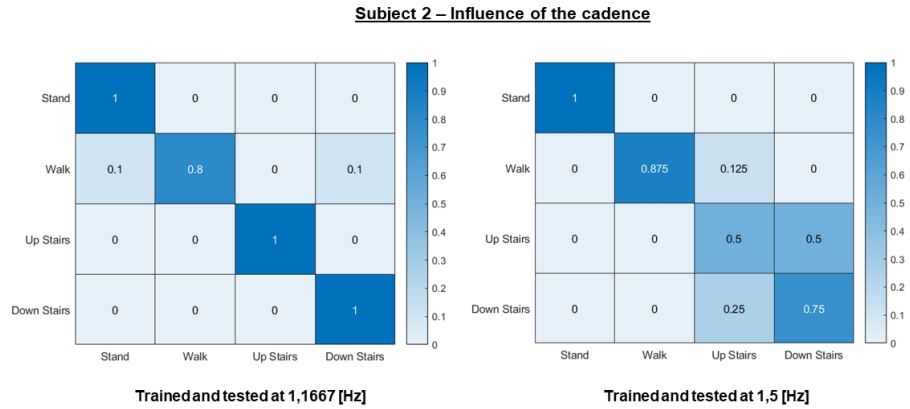


Figure 4.10: Confusion matrices obtained after the experiments conducted with subject 2 performing the 4 tasks at different cadences

4.1.2 Effect of the number of tasks variation

These experiments were conducted with subject 3 and are a comparison between 4 and 6 differentiated tasks. The subject followed a cadence of 1.1667 [Hz] during the training and the testing phase.

Classification accuracy The best result is obtained when only four tasks are differentiated : 90% of well predicted tasks. When adding the RA and RD tasks, a reduction in accuracy of 11% occurs.

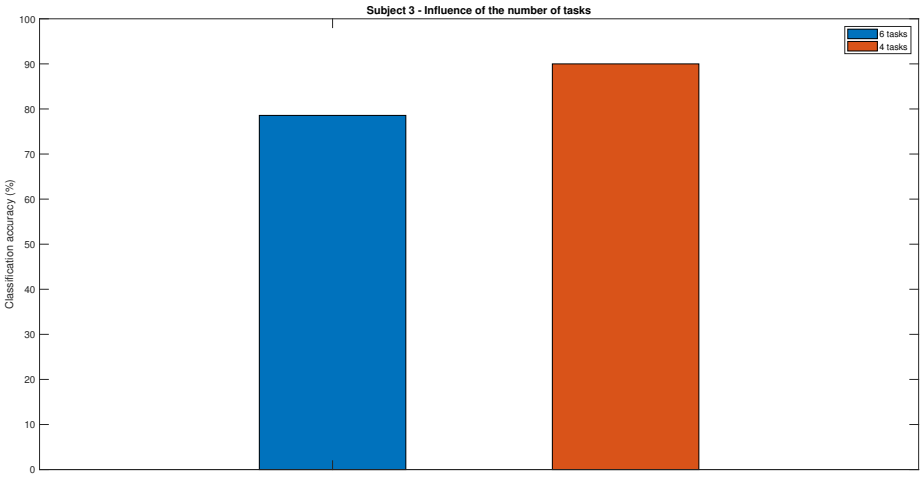


Figure 4.11: Classification accuracy obtained after the experiments conducted with 4 and 6 different tasks

Confusion matrices During the experiment with four tasks, the error only occurs when classifying the SA task. When the RA and the RD are added, the first thing that meets the eye when looking at the matrix shown in figure 4.12 is that all the added errors involves the RA and RD tasks. Indeed, 40% of the error is SA misclassifiedd as RA, 26.66% of the error is SD misclassified as RD and 26.67% of the error is a confusion between RD and WA.

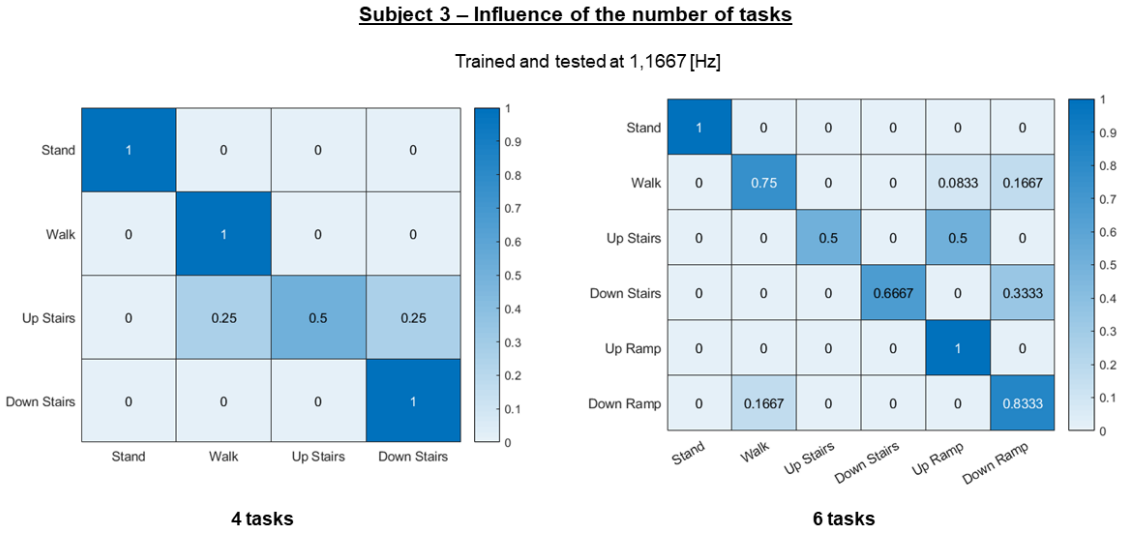


Figure 4.12: Confusion matrices obtained after the experiments conducted with 4 and 6 different tasks

4.1.3 Effect of the number of muscles variation

Initially, eight muscles were chosen to be recorded, in order to see the influence of the number of measured muscles, an experiment was done with two muscles removed and afterwards, a last experiment was completed with only 4 measured muscles remaining. During the conduction of these experiments, a loose contact occurred in the Myoware™ device that was recording the signals of the semitendinosus, these measurements were thus useless for the classification. Hence, it can be said that the experiments were conducted with 7, 5 and 3 muscles. The subject followed a cadence of 1.1667[Hz] and 4 tasks were differentiated (ST, WA, SA, SD).

Classification accuracy Figure 4.13 shows the classification accuracy of these experiments. As expected, the experiment with seven muscles has the highest accuracy with 90%. An accuracy reduction of 10% occurs when removing two muscles and when only three recorded muscles remain, an additional reduction of 20% occurs lowering the classification accuracy at 60%.

Confusion matrices With seven measured muscles, it can be seen in figure 4.14 that only the SA task was misclassified. Indeed 50% of the total error is the SA misclassified as

WA, and the other 50% is the SA misclassified as SD. The experiments with five measured muscles had the greatest difficulties classifying the SD : 66.67% of the error is SD predicted as WA, 22.22% is SD predicted as SA and 11.11% is WA predicted as SA. Finally, with three measured muscles, in addition to the SD, the WA was also difficult to classify: 50% of the error is SD as predicted ad SA, 25% is WA predicted as SA, 16.67% is SA as SD and 8.3% is WA predicted as SD.

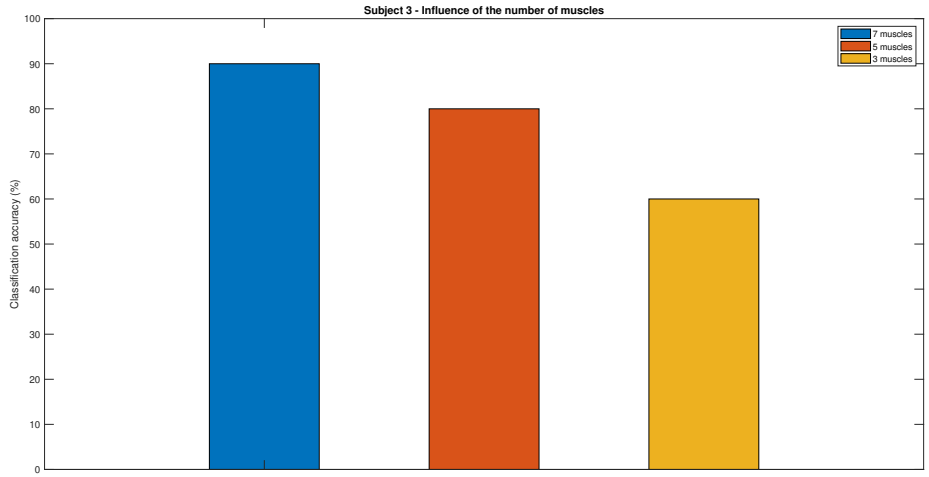


Figure 4.13: Classification accuracy of the experiments conducted with 7, 5 and 3 muscles

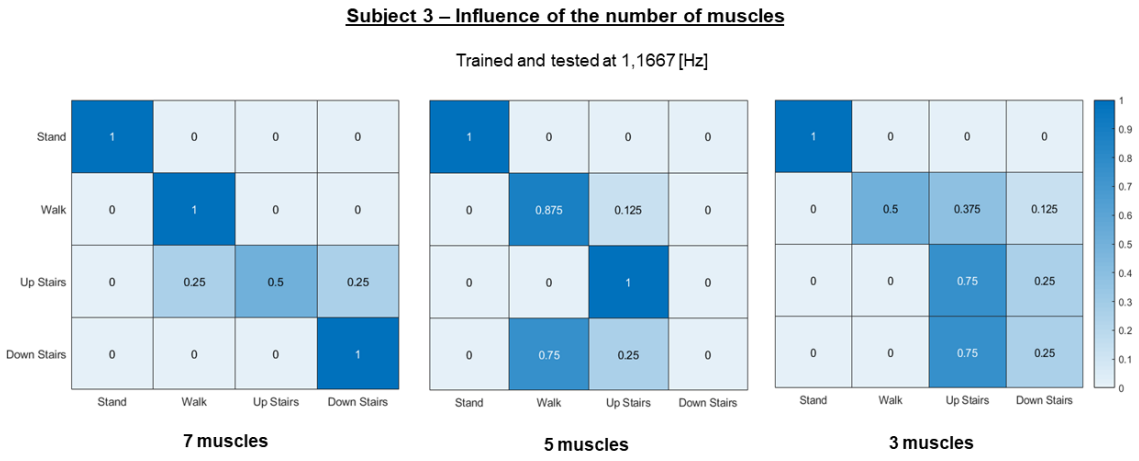


Figure 4.14: Confusion matrices of the experiments conducted with 7, 5 and 3 muscles

Features graph analysing The confusion between tasks shown by the confusion matrices can also be visualized when analysing the features graphs. Indeed, this graphs show the values of the features at each taken decision for each muscle. If the graphs are analysed as explained in section 3.4.3, it will allow to highlight the muscles that were the most likely

to induce errors. Regarding the number of muscles comparison, it could be interesting to analyse if the removed muscles induced a lot of errors or not.

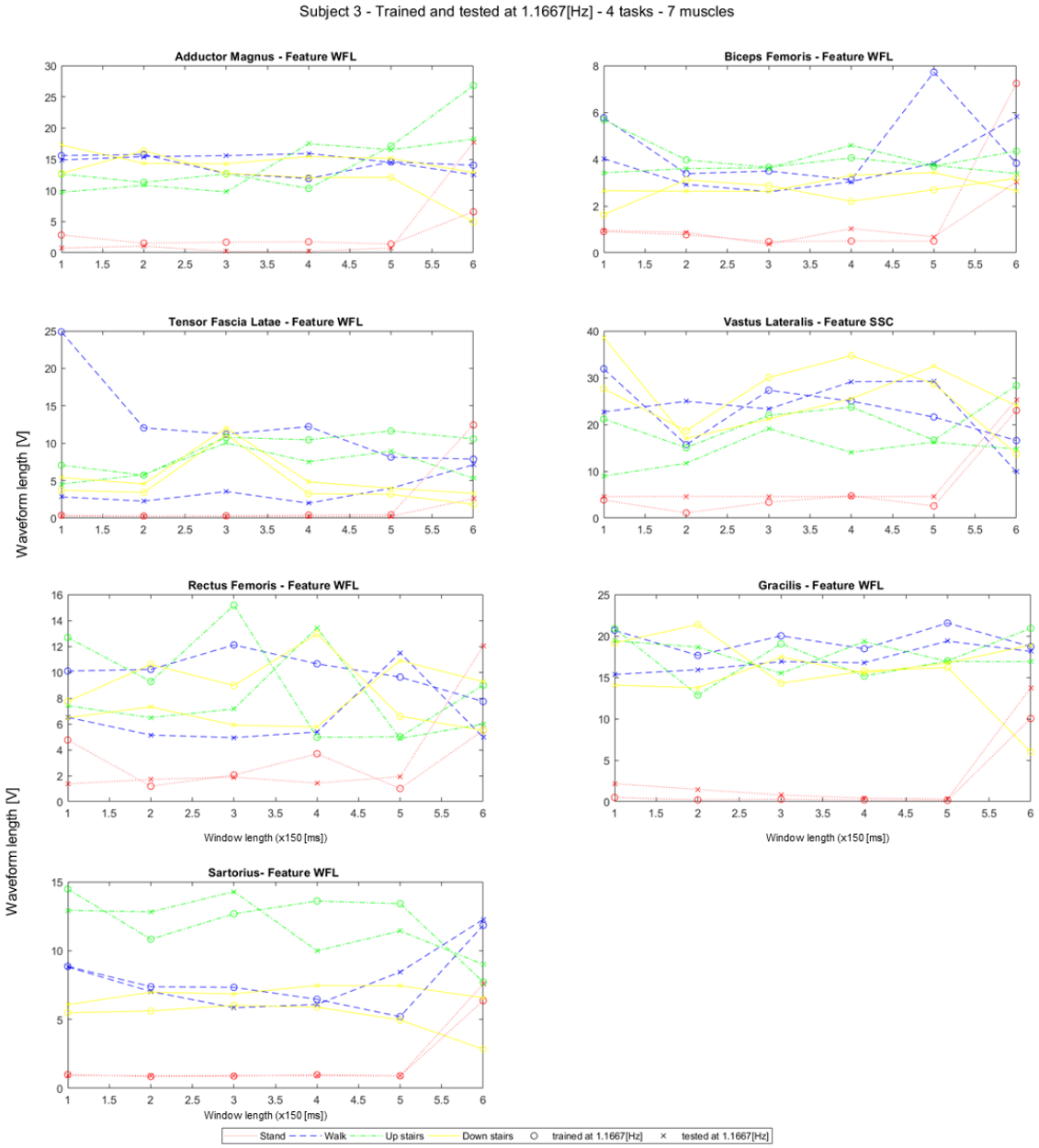


Figure 4.15: Waveform length computed during an experiment conducted at a cadence of 1.1667 [Hz], with subject3, 7 muscles and tasks

For instance figure 4.15 shows the waveform length computed during the experiment for the seven different muscles. One can visualize that it is likely that the sartorius were more efficient than the rectus femoris for which the tested and trained lines in one same task are more apart from one another.

Figure 4.16 shows the percentage of the total error induced by each muscle. As a reminder of section 3.4.3, these percentages are not values obtained directly from the classifier but are computed indirectly based on the feature graphs. Hence, the results should be interpreted as a general tendency. According to the features graph, the rectus femoris is the muscle that induced the more errors unlike the sartorius which induced the least errors. The same analyse was made for the 5-muscles experiment.

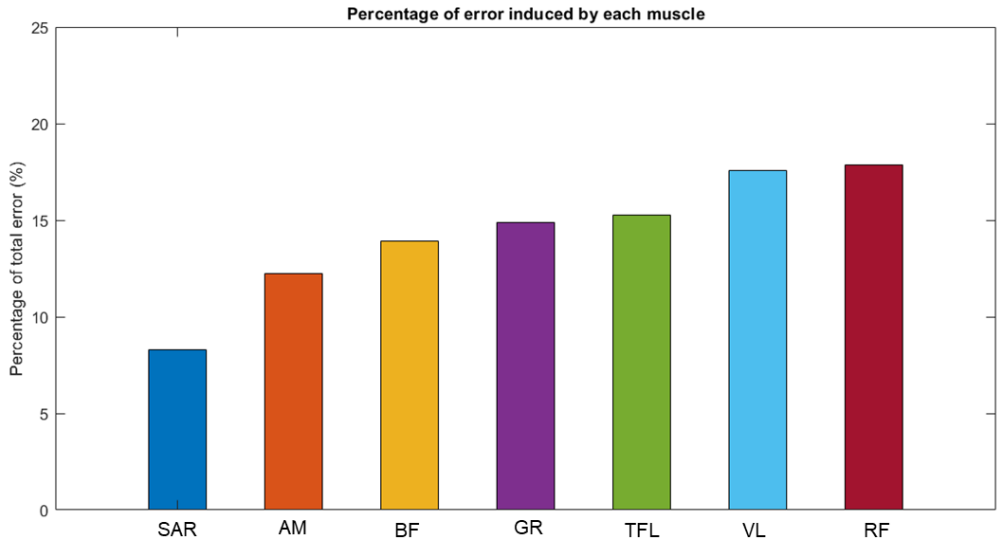


Figure 4.16: Error induced by each muscle for an experiment conducted at 1.1667[Hz] with subject 3, 4 tasks and 7 muscles

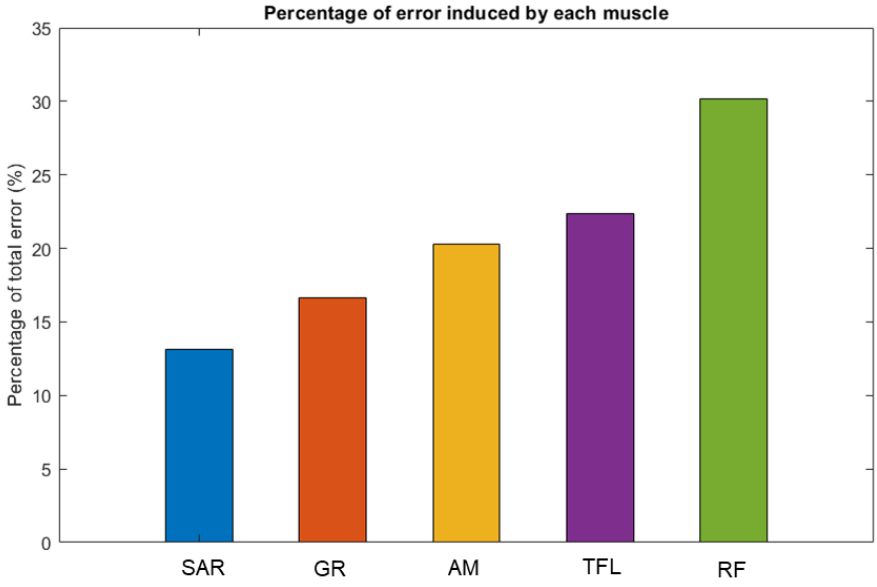


Figure 4.17: Error induced by each muscle for an experiment conducted at 1.1667[Hz] with subject 3, 4 tasks and 5 muscles

Figure 4.17 shows that the same tendency is kept with the sartorius having the best score and the rectus femoris the worst. Fortunately, a muscle that induced a lot of errors was removed, the vastus lateralis and the two best muscles (SAR and AM) were kept.

4.1.4 Effect of the subject variation

These experiments are conducted with a subject's cadence of 1.1667[Hz] during the training and testing phase. Four tasks had to be differentiated. Information about the subjects was mentioned at the beginning of this chapter. Figure 4.18 shows an example of the differences that can occur in the pattern of the EMG signals between two different subjects.

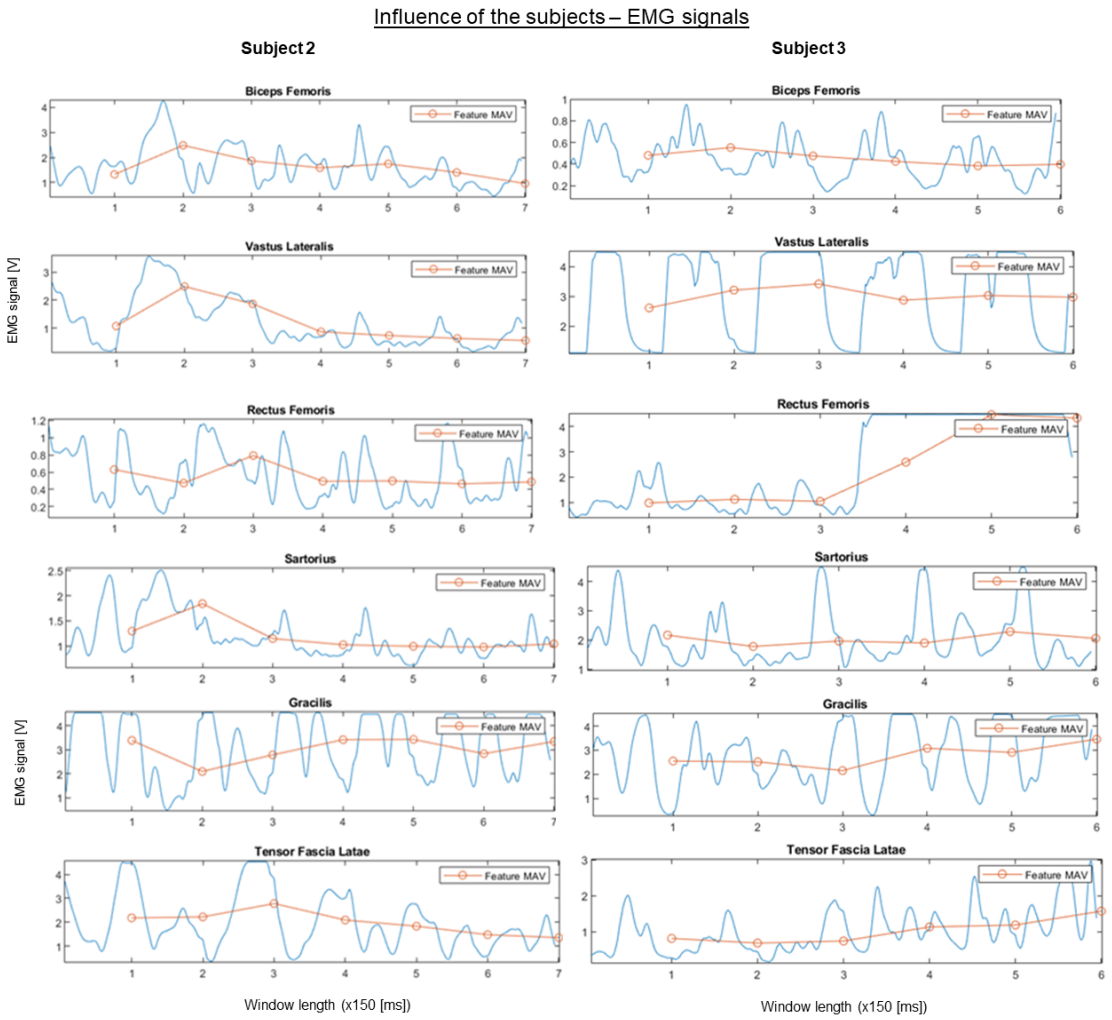


Figure 4.18: EMg signals of the different muscles recorded during an experiment conducted with subject 2 and another experiment conducted with subject 3

Classifiaction accuracy The three experiments conducted with subject 1, subject 2 and subject 3, respectively score 96.43%, 92% and 90%. This is shown in figure 4.19.

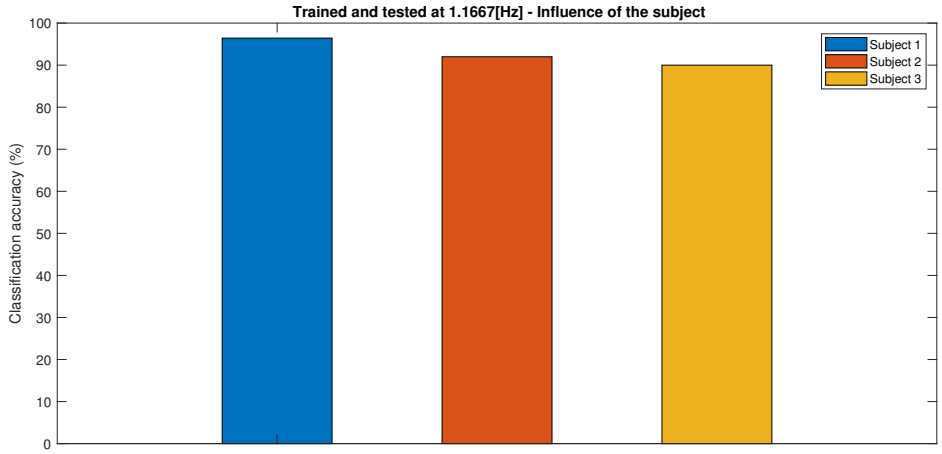


Figure 4.19: Classification accuracy of the experiments conducted with subject 1, 2 and 3

Confusion matrices Regarding the experiment completed With subject 1, the error is divided between a confusion of SA with WA and a confusion of SD with WA. With subject 2, the WA task is misclassified once as ST and once as SD. The last experiment with subject 3 has 50% of the error that occur because SA is predicted as WA an the other 50% because SA is predicted as SD.

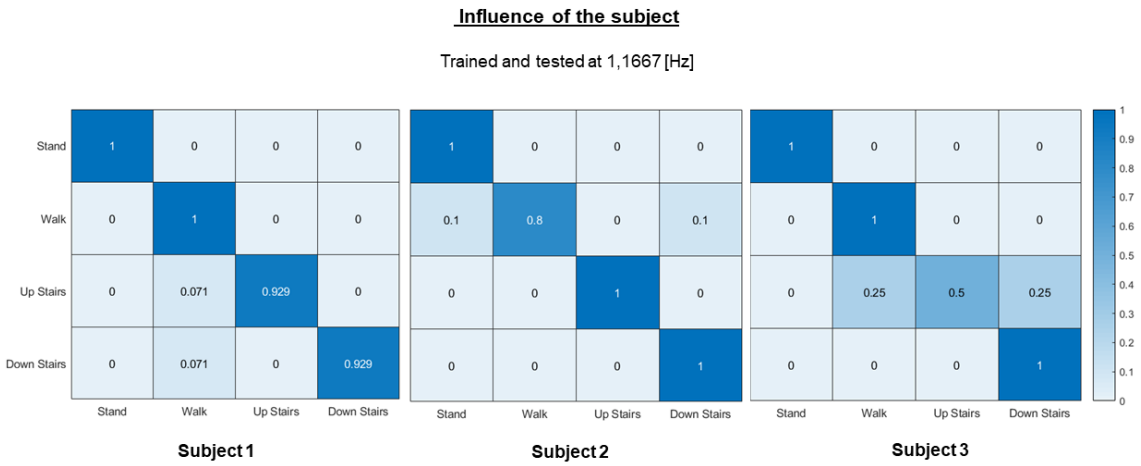


Figure 4.20: Confusion matrices of the experiments conducted with subject 1, 2 and 3

4.2 Transitional experiments

The transitional experiments are evaluated based on the transitional accuracy and the transitional time as explained in section 3.4. The experiments were conducted in transitional mode (section 3.5.3) at a fixed normal cadence of 1.1667 [Hz]. The circuits performed with the different subjects are shown in section 3.5.3.2.

Transitional accuracy The best score, 90%, is obtained for the experiment with subject 3 and 4 different tasks. When adding the RA and RD tasks, a reduction in accuracy of 18.57% occurs. Meanwhile, the experiment with subject 1 and 4 tasks only scores 75%.

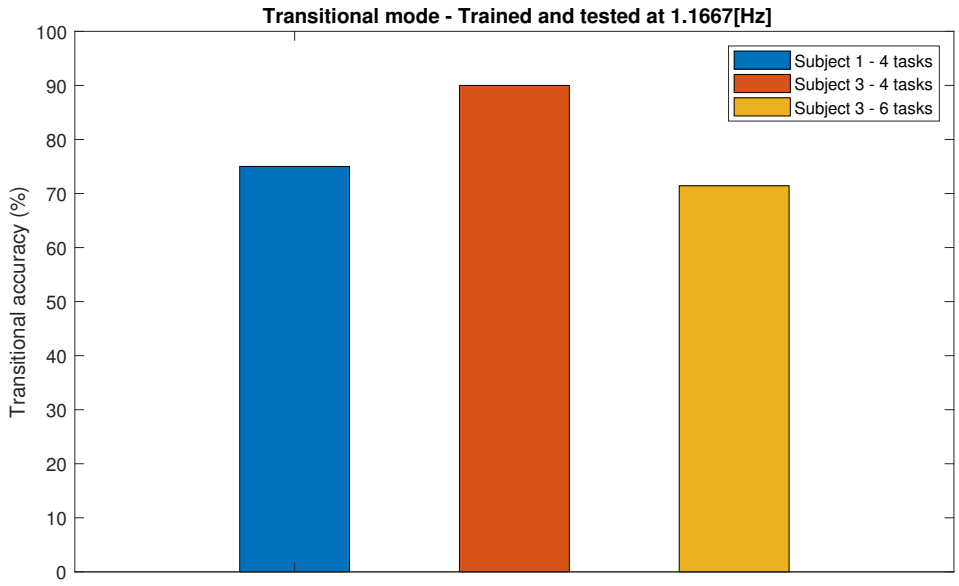


Figure 4.21: Transitional accuracy obtained after the experiments conducted with subject 1 and 4 tasks; subject 3 and 4 tasks; subject 3 and 6 tasks

Transitional time The transitional times are computed following the method explained in section 3.4. It represents the difference in time between the predicted transition and the actual transition step. A positive value means that the system is in advance with respect to the actual movement. Here follows the transitional times for the three performed experiments:

1. Subject 1 - 4 tasks:
   - WA $\rightarrow$ SA : 293 [ms]
   - SA $\rightarrow$ ST : 134 [ms]
   - ST $\rightarrow$ SD : - 413 [ms]
   - SD $\rightarrow$ WA : - 298 [ms]
   - WA $\rightarrow$ ST : - 183 [ms]
2. Subject 3 - 4 tasks:
   - ST $\rightarrow$ SA : 412 [ms]
   - SA $\rightarrow$ WA : 516 [ms]
   - WA $\rightarrow$ SD : - 318 [ms]

- SD → WA : 978 [ms]
- WA → ST : - 374 [ms]

3. Subject 3 - 6 tasks:
   - ST → SA : 464 [ms]
   - SA → WA : 574 [ms]
   - WA → SD : 272 [ms]
   - SD → WA : 85 [ms]
   - WA → RA : 411 [ms]
   - RA → ST : 107 [ms]
   - ST → RD : 990 [ms]

Chapter 5

Discussion

This chapter first proposes a discussion of the results obtained in the previous chapter. For each parameter variation, hypotheses will be given about the why of these results and solutions will be proposed when needed. Afterwards, some methodical considerations are presented and the chapter ends with propositions to enhance the system and go further into the intention detection of locomotion modes for a high-level control of lower limb prostheses.

5.1 Steady-state experiments: effects of the different varied parameters

Effect of the cadence variation The first conclusion that comes out of the results obtained after the experiments conducted at a varied cadence, is that the testing must be done at the same cadence as the training. Otherwise, the system is not capable of providing an accurate classification of the different tasks. Indeed, the system is sensitive to the speed of locomotion of the subject.
Moreover, when it comes to train and test the system at the same speed, a normal cadence of gait (1.1667 [Hz]) provided better results than a slower (0.8333 [Hz]) or a faster (1.5 [Hz]) cadence. The inaccuracies that occurred during the slower cadence could be explained by the fact that it was difficult for the subject to keep a smooth and normal gait when he/she had to follow the metronome at an abnormally slow speed.
Regarding the faster cadence, it is likely that the system could not be fast enough and hence induced more errors. In fact, one decision was obtained at a rate of 0.69[Hz] that is in 1.45 [s] and one gait cycle at the fastest cadence is performed in 1.333 [s] whereas a gait cycle for the slowest and the normal cadence is performed in 1.71 [s] and 2.4 [s] respectively. That means that, for a faster cadence, the system is not able to take at least one decision per gait cycle. Hence, the signal captured in one time-window will be more dense and complex to classify.

Ideally, one gait cycle should be divided into phases and one decision should be taken for each phase. For instance, the toe-off and the heel-contact could be detected with a instrumented insole, dividing the gait cycle in two phases: the swing phase and the stance phase (see section 2.2). This way, the information to process in one time-window would

be less dense and hence easier to classify. It is likely then that the system becomes less sensitive to the cadence variations of the subjects.

Effect of the muscles number variation The initial choice of the muscles to be removed was justified in section 3.5.2. The first goal was to see if the classification could still be done accurately when removing one out of the two muscles belonging to the same group muscles (the hamstrings and the quadriceps). However, as explained in section 4.1.3, the recordings of the semitendinosus were useless and therefore the classifier could use the information of 5 muscles instead of 6 as shown in figure 5.1. Hence, all the hamstrings were removed and the classifier had no information about the hip extension. The reduction in accuracy from the 7-muscles experiment to the 5-muscles experiment was of 11% and the majority of the error was induced by a misclassification of the SD task. Indeed, when looking at the major muscle activity during a stair descent gait cycle that was shown in figure 2.11, it is the task that uses the hamstrings the most. However, the accuracy reduction is not tremendous even with the absence of the hamstrings. This could be explained by the fact that the vastus lateralis, which was removed, was one of the muscle that induced the most errors during the 7-muscles experiment. Indeed, it was shown in the results chapter in figure 4.17 that it was likely that the vastus lateralis and the rectus femoris were the worst muscles in terms of induced errors. Therefore, it is likely that the accuracy reduction is due to the fact that no more hip extensors are recorded whereas they are important for the SD, while this reduction was compensate by the fact that the vastus lateralis, which induced a lot of errors during the 7-muscles experiment, was removed.

Knee Extensors • Rectus Femoris • Vastus Lateralis • Vastus Medialis (Quadriceps)	**Hip Extensors** • Gluteus Maximus • Biceps Femoris • Semitendinosus • Semimembranosus	**Hip Abductors** • Gluteus Maximus • Tensor Fascia Latae
Knee Flexors • Biceps Femoris • Semitendinosus • Semimembranosus (Hamstrings) • Sartorius (aids) • Gracilis (aids)	**Hip Flexors** • Rectus Femoris • Vastus Lateralis • Vastus Medialis • Sartorius	**Hip Adductors** • Adductor Magnus • Gracilis

Figure 5.1: 5 measured muscles during the experiment with subject 3 , 4 tasks and a cadence of 1.1667 [Hz] and the function of these muscles

Between the 5- and 3-muscles experiment, the reduction accuracy doubled compared to the 7- to the 5-muscles experiment. This 20% of accuracy reduction was still mostly due to the misclassification of the SD class. But, in this case, it was the most confused with the SA task. Moreover, the WA task was also significantly confused with the SA task. Indeed, when looking at the functions of the remaining muscles shown in figure 5.2, in addition to the absence of the hip extensors, there is an absence of hip abductors and the knee flexors are weakened with only the sartorius remaining (which is only an aid to the knee flexion).

The knee flexion is more pronounced during the SA than during the WA but the classifier does not get a clear information about the knee flexion anymore which could induce the confusion between WA and SA.
Moreover, the rectus femoris which is the muscle that is likely to have induced the more errors during the 5-muscles experiment was not removed. The second worst (TFL) and the second best (GR) muscles were not recorded which could not incite a compensation.

Knee Extensors • Rectus Femoris • Vastus Lateralis • Vastus Medialis (Quadriceps)	**Hip Extensors** • Gluteus Maximus • Biceps Femoris • Semitendinosus • Semimembranosus	**Hip Abductors** • Gluteus Maximus • Tensor Fascia Latae
Knee Flexors • Biceps Femoris • Semitendinosus • Semimembranosus (Hamstrings) • Sartorius (aids) • Gracilis (aids)	**Hip Flexors** • Rectus Femoris • Vastus Lateralis • Vastus Medialis • Sartorius	**Hip Adductors** • Adductor Magnus • Gracilis

Figure 5.2: 3 measured muscles during the experiment with subject 3 , 4 tasks and a cadence of 1.1667 [Hz] and the function of these muscles

To sum up, in order to have a good accuracy, it is best to ensure that per knee- and hip-joint movement, at least one principal responsible muscle for this function is recorded.

Effect of the tasks number variation The results showed an accuracy reduction of 11% when adding the RA and RD tasks and it was observed that this additional error was only induced by misclassification of the RA and the RD tasks. Indeed, depending on the slope inclination, the signal patterns of ramp ascent and descend could be confused with walking patterns or stair ascent and descent patterns. During the experiments, the RA and RD tasks were the most confused with the SA and SD task respectively.
In order to avoid this confusion, sensors capturing the environment could be added to the system. This way, the RD and RA tasks could be trained as to be SD and SA while the system would ba able to recognise if the subject is on a stair or a ramp and hence differentiate the tasks. Another way to avoid the confusion with the walking task could be to add mechanical sensors on the foot to compute the ground inclination [26][27][28].

Effect of the subject variation The EMG signals recorded during the experiments show that the pattern of one gait cycle can be very different from one subject to another and even from one cycle to another for the same subject. Indeed, the activity in the muscles is something very complex and nobody has the exact same anatomy or has the same physiological reactions. Moreover, the electrodes placement could also be the cause of inaccuracies, but within the framework of this thesis, it is not possible to separate the errors induced by the misplacement of the electrodes and the ones induced by the subject physiological differences. Hence the importance of the training phase thanks to which the

accuracy of the results can be maintained above 90% for the experiments conducted at a cadence of 1.1667[Hz] and with 4 tasks to differentiate.

However, this mandatory training phase is not very practical for frequent use. Therefore, some studies try to find alternatives to this training phase. For instance, Aaron J. Young et al. [41], [58] propose to train one classifier with a pool of subjects and then analyse if it would work on a new subject without re-training.

5.2 Transitional experiments

Classification accuracy during transitional steps The numbers of misclassified samples are more important during the transitional steps than during the steady-sate mode. For instance, there is a 21% difference in accuracy between the steady-state mode and the transitional mode for experiments with subject 1 and 4 different tasks and a 8% difference for experiments with subject 3 and 6 tasks.
As was proposed earlier in this section regarding the cadence variation, adding an sensorized insole to detect the toe-off and the heel-strike so to be able to divide one gait cycle in phases, could be a solution to assure better classification at the transitional steps. Indeed, it would allow a better capture of the signal at the transition time and hence, a less complicated signal and a better classification.

Transitional timing Regarding the transitional time, the system have an advance of 214.7 [ms] in average on the actual movement. In fact, E.C. Wentink et al. [42][59] affirm an advance of 130-260 [ms] on mechanical sensors when the prosthetic leg leads and according to [9] and [10] the electromechanical delay between the neural activity and the actual force production in the muscles is between 30 and 100 [ms]. However, the results show a maximum of 990 [ms] advance and 413 [ms] delay with respect to the movement. These values are quit surprising.
Indeed, with the system slowness, it is not possible to determine the predication time with enough accuracy. This can be explained by the fact that it is assumed that the transition occurs in between the two decisions that mark the transition (section 3.4). But the time lapse between two decisions is of 1.45 [s] (section 3.2.1) and hence is to great to be able to determine the prediction time accurately.
This problem is illustrated in figure 5.3 which shows an example where the five first decisions of the system predict the standing task and the five next decisions predict the walking task. The toe-off must have occurred between the 5^{th} decision and the 6^{th} decision but it is not known precisely when it happened. It was assumed that the prediction time is the time equally between the two transition decisions. therefore, with the time between two decisions being of 1.45 [s], the inaccuracy of the predicted time can go up till 0.725 [ms] which is a lot considering the values of E.C. Wentink,et al.[42][59].

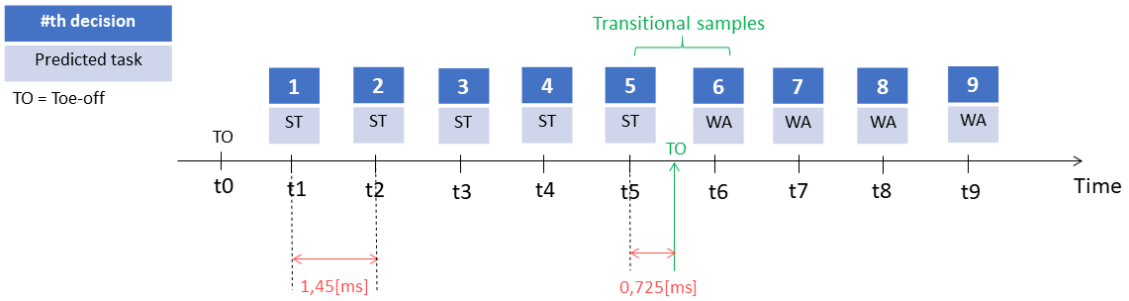


Figure 5.3: Illustration of the timing during a transition step

As a result, due to the fact that the system is to slow, the transitional time results cannot be taken into consideration. To allow accurate results, the system should be able to take decisions at a much faster rate.

5.3 Methodical considerations

During the *Method* chapter several issues were encountered. Some were resolved during the process and some persist and are presented below followed by a reflexion of possible solutions.

One of the main weakness of system is its slowness. Indeed, the decisions are taken at a rate of 0.69[Hz] which allow to take only one to two decisions per gait cycle at a normal cadence. One of the main reason of this delay is the fact that the Analog-to-digital converter is unable to perform the sampling of the eight channels at the same time as was explained in section 3.1.2. If the eight signals could be sampled at the same time, with a cycling throughput of 837 [Hz] and the current values of window length and increment, the decision stream rate could be of 5.58 [Hz] which is eight time faster. It would then be possible to take more than 2 decisions over one gait cycle and hence to predict with more accuracy. Unfortunately, an ADC with a Raspberry Pi interface and 8 channels that can be sampled at exactly the same time is difficult to find. Two solutions could then be considered:

1. Each Myoware^{TM} has its own ADC integrated circuit. That could have been done by designing a PCB (printed circuit board) proper to this application with one ADC chip for each measured muscle.

2. Find another device than the Myoware^{TM} to measure the EMG signals that includes an ADC. For instance the LPEMMG-B2 PLUS [60] which communicates by Bluetooth.

The first solution will be more difficult to implement but more cheaper than the second one.

The biggest issue encountered with the EMG method is the fact that the signals are strongly non-stationary. Indeed, from one subject to another and especially from one cycle to another, the EMG recordings can present strong differences. This non-stationarity is partly due to the noise sensitivity : motion artifacts, crosstalk between adjacent muscles, electrical noise. But above all, the muscle activity will differ because everybody has a different anatomy, physiology and reacts in different ways to external loads or events. Moreover, the muscles can be subjected to fatigue which can also alter the EMG signals. Therefore, even if the noise is perfectly filtered, some issues will persist to occur due this non-stationarity. Some possible way to compensate this drawback would be to consider a fusion with mechanical sensors.

5.4 Going further

All along this chapters, issues encountered during the system evaluation or the system implementation were mentioned. Some solutions were proposed for each one of them to enhance the system and allow to get closer to a system accuracy that would be clinically accepted. That is, that ensures the safety of the users. Based on these issues, this last section presents several steps that could be done to go further into the intention detection of locomotion modes with the aim of a prosthesis control for lower-limb amputees.

1. **Instrumented insole** Adding an insole with pressure sensors included would allow to detect the important gait events such as the toe-off and the heel-strike and thus to divide the gait cycle in different phases. Each phase could then have its proper classifier which would render the signals to classify less dense and hence, easier to predict. This could allow a reduction in sensitivity to the cadence of the subject because the system becomes phase-dependent and in addition, it could make the classification at the transitional steps easier. He Huang et al.[11] propose a phase-dependent classifier by dividing one gait cycle in four phases thanks to an insole pressure sensor.

2. **Neuro-mechanical fusion** In order to compensate the sensitivity of the EMG techniques and the non-stationarity of the signals, several studies propose a fusion between EMG techniques and the use of mechanical sensors. The addition of a sensorized insole was already an example of such methods.
   Some studies go further and propose a fusion of EMG recordings and mechanical sensors such as IMU's. Indeed, features of the mechanical signals could be fused with the features of the EMG signals to provide a more complete set of features used for the classification. For instance, another work of He Huang[39] proposes a neuromuscular-mechanical fusion as shown in figure 5.4. They compared this fusion method with techniques using only EMG and only mechanical sensors and state that the fusion method presents the best accuracy. Domen Novak and Robert Riener [61] propose a review of the sensor fusion methods in wearable robotics and all the following articles investigate the fusion between EMG and mechanical sensors for gait intention detection for lower-limb amputees : [38], [40], [42], [50], [59], [44],

[45] and [46].

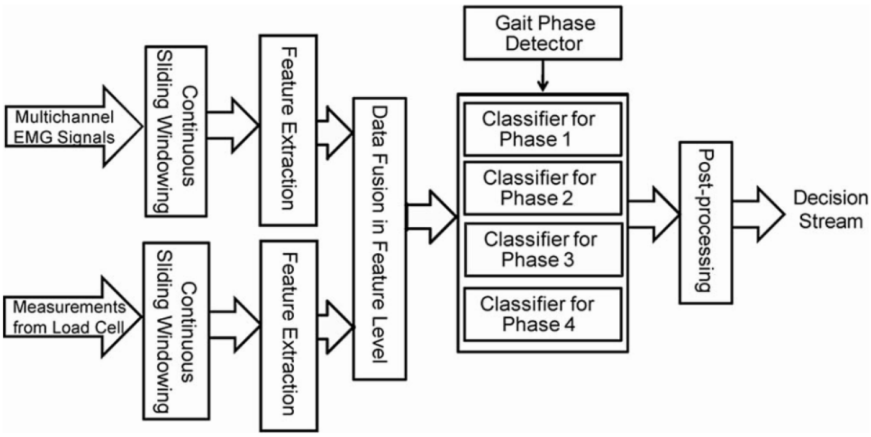


Figure 5.4: Architecture of the classifier proposed by He Huang[39] based on neuromuscular–mechanical fusion

3. **Training phase** One practical issue is the training phase that have to be done each time the device is put on. With the perspective of using this methods for lower-limb prosthesis, it would not be convenient for a daily-use. Aaron J. Young et al. [41][41] investigate an user-independent classifier by training the classifier on a pool of lower limb amputees before testing it on a novel user.

4. **Amputated subjects** One of the next steps is of course to test the system on lower limb amputated subjects. Indeed, their gait is altered and new parameters might be taken into account.

5. **Environment detection** In order to allow the system to work even in complicated environment such as a crowd, the system could include an environment detection. In addition to allow obstacles detection, it could especially help in the discrimination of the ramp ascent and descent task. Indeed, depending on the inclination of the ramp, these task are easily confused with walking or with stair ascent and descent respectively. If sensors could detect the presence of a stair or a ramp in the environment (through cameras for instance) or compute the foot inclination, the confusion between theses task might be reduced. The following articles investigate such methods [26][27][28][29][30].

Conclusion

Passive prostheses for lower-limb amputees are nowadays widely used but yet do not satisfy the user needs. Active powered prostheses have been developed over the last decades and provide the necessary energy to perform different locomotion modes in a comfortable way. To ensure that the prosthesis follow the user's intent, a high-level control based on the detection of the locomotion modes intention have to be designed. In the aim of helping to achieve this objective, this thesis investigated a way to detect the gait intention of healthy subjects. The differentiated locomotion modes were still standing, level-ground walking, stair ascent, stair descent, ramp ascent and ramp descent.

After a review of the literature, the electromyography (EMG) method was chosen based on its good timing performances and its integration of voluntary control. This technique records the neuromuscular activity of skeletal muscles via surfaces electrodes. The analysis of the biomechanics of human gait allowed to select the eight muscles to be measured : the rectus femoris, the vastus lateralis, the biceps femoris, the semitendinosus, the sartorius, the tensor fascia latae, the adductor magnus and the gracilis.

The EMG signals were recorded thanks to the *Myoware*^{TM} which is a device that filters, amplifies, rectifies and integrates the raw EMG. The data was then sampled at 1000 [Hz] by the *High-Precision AD/DA board* and send via SPI to the microcontroller, the Raspberry PI 3B+. A sliding window algorithm was implemented to capture the signal in time-dependent windows and to allow the extraction of the signal features over each window: the mean absolute value, the variance, the slope sign changes and the waveform length. Based on the computed features of the eight recorded muscles, a linear discriminant analysis (LDA) was implemented to classify the signals. The classifier required a training phase before the samples could be tested. The obtained decision stream rate was of 0.69 [Hz].

The experiments were conducted according to two modes, the steady-state experiments aimed to determine the classification accuracy for each performed task and the transitional experiments aimed to evaluate the system at the transition steps between two different tasks. During the steady-state experiments, the following parameters were varied : the subjects, the cadence of the subject, the number of recorded muscles and the number of differentiated tasks.

The best results were acquired for the experiments conducted at a normal gait cadence, with four differentiated tasks and eight recorded muscles. Indeed, adding the ramp ascent and descent tasks introduced confusions with the walking, the stair ascent and the stair descent tasks. The system was also sensitive to the subject's cadence as the testing phase had to be done at the same cadence as the training phase to avoid misclassification. Moreover, the decision stream rate was too slow to keep up with a faster gait cadence.

Finally, reducing the number of measured muscles induced an increasing error at an extend depending on which muscles were removed. In order to ensure a good classification, the primarily responsible muscle for each knee- and hip- joint movement should be recorded.

The biggest drawback of the EMG method apart from its sensitivity to noise which can be enhanced with a good treatment, is its non-stationarity. In fact, from one subject to another or even from one gait cycle to another the EMG recordings can be completely different. Therefore, the acquired results are not sufficient for clinical application which requires to ensure the user's safety. However, more and more studies propose neuromuscular-mechanical fusion methods to compensate this drawback of the EMG method while keeping its advantageous timing.

Therefore, to go further, an instrumented insole dividing the gait cycle in different phases should be added to the system to allow a phase-dependent classifier implementation. This way, the system could be less sensitive to the subject's cadence and the transitional steps could be classified with less errors. Moreover, IMU's measurements should also be fused with the EMG signals to provide a more complete set of features and hence a higher classification accuracy. These methods come of course with a device synchronisation challenge but current studies showed promising results, encouraging the further investigation of these methods and allowing to get closer to a locomotion modes intention detection that could be used by a high-level controller of a lower limb prosthesis.

Bibliography

[1] [Online]. Available: http://uf-mi.u-bordeaux.fr/ter-2016/mazouffre-figus/quelques-chiffres/statistiques/

[2] J.-M. Vanmarsenille, "Amputations : types d'appareillages," *Service d'orthopédie des Cliniques universitaires Saint-Luc*, 2013.

[3] [Online]. Available: https://mcopro.com/prosthetics/technology/iwalk-biom/

[4] A. P. H. B. M. B. O. L. J. d. R. M. R. R. H. V. Michael R Tucker, Jeremy Olivier and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: a review," *Journal of NeuroEngineering and Rehabilitation*, vol. 12(1), 2015.

[5] M. Nilsson, "Medical engineering basic course, lecture 6, clinical neurophysiology," Royal Institute of Technology Stockholm, 2018.

[6] J. d. R. Millan, R. Rupp, G. Mueller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kubler, R. Leeb, C. Neuper, K. Mueller, and D. Mattia, "Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges," *Frontiers in Neuroscience*, vol. 4, p. 161, 2010.

[7] L. N. T. P. G. L. B. D. Rea M., Rana M., "Lower limb movement preparation in chronic stroke: a pilot study toward an fnirs-bci for gait rehabilitation," *Neurorehabilitation Neural Repair*, vol. 28(6), p. 564–75, 2014.

[8] M. L. Bensmaia SJ, "Restoring sensorimotor function through intracortical interfaces: progress and looming challenges," *Nat Rev Neuroscience*, vol. 15(5), p. 313–25, 2014.

[9] K. P. Cavanagh PR., "Electromechanical delay in human skeletal muscle under concentric and eccentric contractions," *Eur J Appl Physiol Occup Physiol.*, vol. 42(3), pp. 159–63, 1979.

[10] M. L. L. G. L. A. Daniel M. Corcos, Gerald L. Gottlieb and G. C. Agarwal, "Electromechanical delay: An experimental artifact," *Journal of Electromyography and Kinesiology*, vol. 2, pp. 59–68, 1992.

[11] T. A. K. He Huang and R. D. Lipschutz, "A strategy for identifying locomotion modes using surface electromyography," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 56, no. 1, 2009.

[12] C.-Y. C. Deok-Hwan Kim and J. Ryu, "Real-time locomotion mode recognition employing correlation feature analysis using emg pattern," *ETRI Journal*, vol. 36, no. 1, 2014.

[13] M. S. B. Jason D. Miller and M. E. Hahn, "Myoelectric walking mode classification for transtibial amputees," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 60, no. 10, 2013.

[14] M. E. H. Deepak Joshi, Bryson H.Nakamura, "High energy spectrogram with integrated prior knowledge for emg-based locomotion classification," *Medical EngineeringandPhysics*, vol. 37, p. 518–524, 2015.

[15] J. A. D. George D. Fulk and K. B. Fite, "Emg control of a bionic knee prosthesis: Exploiting muscle co-contractions for improved locomotor function," *IEEE International Conference on Rehabilitation Robotics*, 2013.

[16] P. B. Samuel K. Au and H. Herr, "An emg-position controlled system for an active ankle-foot prosthesis: An initial experimental study," *IEEE 9th International Conference on Rehabilitation Robotics*, 2005.

[17] O.-C. M. Lendaro E, "Classification of non-weight bearing lower limb movements: towards a potential treatment for phantom limb pain based on myoelectric pattern recognition." *Conf Proc IEEE Eng Med Biol Soc.*, 2016.

[18] E. J. P. John A. Spanias and L. J. Hargrove, "Detection of and compensation for emg disturbances for powered lower limb prosthesis control," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 24, no. 2, 2016.

[19] M. F. M. A. S. W. A. A. D.-S. B. C. K. Ashwin P. H. Needham, Filip Pascal Paszkiewicz and R. Vaidyanathan, "Subject-independent data pooling in classification of gait intent using mechanomyography on a transtibial amputee," *IEEE International Conference on Robotics and Automation*, 2018.

[20] R. K. B. Abdul Hadi Abdul Razak, Aladin Zayegh and Y. Wahab, "Foot plantar pressure measurement system: A review," *Sensors journal*, vol. 12, 2012.

[21] A. M. S. Aaron J. Young and L. J. Hargrove, "A training method for locomotion mode prediction using powered lower limb prostheses," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 22, no. 3, 2014.

[22] A. J. Young and L. J. Hargrove, "A classification method for user-independent intent recognition for transfemoral amputees using powered lower limb prostheses," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 24, no. 2, 2016.

[23] L. A. N. V. D. L. G. P. Maja Gorsic, Roman Kamnik and M. Munih, "Online phase detection usingwearable sensors forwalking with a robotic prosthesis," *Sensors journal*, vol. 14, 2014.

[24] S. M. M. D. R. M. D. J. P. T. B. T. L. N. V. M. C. C. M. M. Domen Novak, Peter Rebersek, "Automated detection of gait initiation and termination usingwearable sensors," *Medical Engineering Physics*, vol. 25, p. 1713– 1720, 2013.

[25] J. M. Dongfang Xu, Yanggang Feng and Q. Wang, "Real-time on-board recognition of continuous locomotion modes for amputees with robotic transtibial prostheses," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 26, no. 10, 2018.

[26] N. V. D. J. Li Q., Young M., "Walking speed and slope estimation using shank-mounted inertial measurement units," *IEEE international conference on Rehabilitation robotics*, p. 839–44, 2009.

[27] Z. R. W. R. Z. J. JIN Dewen, YANG Jiankun, "Terrain identification for prosthetic knees based on electromyographic signal features," *TSINGHUA SCIENCE AND TECHNOLOGY*, vol. 11, no. 1, 2006.

[28] D. JOSH and M. E. HAHN, "Terrain and direction classification of locomotion transitions using neuromuscular and mechanical input," *Annals of Biomedical Engineering*, vol. 44, no. 4, 2016.

[29] L. M. H. H. Zhang F., Fang Z., "Preliminary design of a terrain recognition system," *Engineering in medicine and biology society, EMBC*, p. 5452–55, 2011.

[30] d. R. M. Carlson T., "Brain-controlled wheelchairs: a robotic architecture," *Robot Automation Mag IEEE*, vol. 20, no. 1, 2013.

[31] [Online]. Available: https://courses.lumenlearning.com/wm-biology2/chapter/action-potential/

[32] J. Hamill, K. M. Knutzen, and T. R. Derrick, *Biomechanical Basis of Human Movement*, 4th ed. Wolters Kluwer, 2015.

[33] *3-lead Muscle/Electromyography Sensor for Microcontroller Applications Datasheet*, Advancer Technologies, 2015. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MyowareUserManualAT-04-001.pdf

[34] J. E. C. S. P. A. L. J. H. G. A. D. Jason M. Souza, Nicholas P. Fey, "Advances in transfemoral amputee rehabilitation: Early experience with targeted muscle reinnervation," *Curr Surg Rep*, vol. 2, no. 51, 2014.

[35] [Online]. Available: https://www.verywellfit.com/rectus-femoris-definition-3120373

[36] N. G. Harper, J. M. Wilken, and R. R. Neptune, "Muscle function and coordination of stair ascent," *Journal of Biomechanical Engineering*, vol. 140, 2018.

[37] [Online]. Available: https://fr.slideshare.net/dr_hetvi/biomechanics-of-stair-climbing

[38] A. J. Young, A. Simon, and L. J. Hargrove, "An intent recognition strategy for transfemoral amputee ambulation across different locomotion modes," *35th Annual International Conference of the IEEE EMBS*, 2013.

[39] H. Huang, F. Zhang, R. D. Lipschutz, Z. Dou, D. R. Rogers, and K. B. Englehart, "Continuous locomotion-mode identification for prosthetics legs based on neuromuscular-mechanical fusion," *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, vol. 58, no. 10, 2011.

[40] A. M. Simon, N. P. Fey, K. A. Ingraham, and A. J. L. J. Hargrove, "Powered prosthesis control during walking, sitting, standing, and non-weight bearing activities using neural and mechanical inputs," *6th Annual International IEEE EMBS Conference on Neural Engineering*, 2013.

[41] A. J. Young, A. M. Simon, N. P. Fey, and L. J. Hargrove, "Classifying the intent of novel users during human locomotion using powered lower limb prostheses," *6th Annual International IEEE EMBS Conference on Neural Engineering*, 2013.

[42] E. Wentink, S. Beijen, H. Hermens, J. Rietman, and P. Veltink, "Intention detection of gait initiation using emg and kinematic data," *Gait Posture*, vol. 37, pp. 223–228, 2013.

[43] Z. Jie, *High-Precision AD/DA Board User Manual*, Waveshare, October 2015, revision 1.2. [Online]. Available: https://www.waveshare.com/w/upload/b/b7/High-Precision-AD-DA-User-Manual.pdf

[44] J. A. S. et al., "Online adaptive neural control of a robotic lower limb prosthesis," *Journal of Neural Engineering*, vol. 15, 2018.

[45] A. J. Y. et al., "Analysis of using emg and mechanical sensors to enhance intent recognition in powered lower limb prostheses," *Journal of Neural Engineering*, vol. 11, 2014.

[46] D. Tkach and L. Hargrove, "Neuromechanical sensor fusion yields highest accuracies in predicting ambulation mode transitions for trans-tibial amputees," *35th Annual International Conference of the IEEE EMBS*, 2013.

[47] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "Emg feature evaluation for improving myoelectric pattern recognition robustness," *Expert Systems with Applic ations*, vol. 40, p. 4832–4840, 2013.

[48] *Very Low Noise 24-Bit Analog-to-Digital Converter*, Texas Instruments, September 2013. [Online]. Available: http://www.ti.com/lit/ds/sbas288j/sbas288j.pdf

[49] [Online]. Available: http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes

[50] O. Bai, R. Atri, J. S. Marquez, and D.-Y. Fei, "Characterization of lower limb activity during gait using wearable, multi-channel surface emg and imu sensors," *5th International Electrical Engineering Congress, Pattaya, Thailand*, 2017.

[51] D. Tkach, H. Huang, and T. A. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Tkach et al. Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 21, 2010.

[52] E. Scheme, K. Englehart, and B. Hudgins, "Selective classification for improved robustness of myoelectric control under non-ideal conditions," *IEEE Transactions on Biomedical Engineering*, vol. 58, pp. 1698–1705, 2011.

[53] [Online]. Available: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

[54] S. Raschka, "Linear discriminant analysis," *Bit by Bit*, 2014.

[55] J. Brownlee, "Linear discriminant analysis for machine learning," *Understand Machine Learning Algorithms*, 2016.

[56] J. Taylor, "Stats 202: Data mining and analysis, lecture 9: Classification, lda," Stanford University, 2018.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[58] A. J. Young and L. J. Hargrove, "A classification method for user-independent intent recognition for transfemoral amputees using powered lower limb prostheses," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, vol. 24, no. 2.

[59] E. Wentink, V. Schut, E. Prinsen, J. Rietman, and P. Veltink, "Detection of the onset of gait initiation using kinematic sensors and emg in transfemoral amputees," *Gait Posture*, vol. 39, pp. 391–396, 2014.

[60] [Online]. Available: https://lp-research.com/lpemmg-b2-plus/

[61] R. R. Domen Novak, "A survey of sensor fusion methods in wearable robotics," *Robotics and Autonomous Systems*, vol. 73, pp. 155–170, 2015.

Appendix A

ROS nodes implementation

A.1 Code of the publisher node and the feature service node

```
#include "ros/ros.h"
#include <stdlib.h>
#include <signal.h>
#include <time.h>
#include <stdio.h>
#include <string.h>
#include <wiringPi.h>
#include <wiringPiSPI.h>
#include <myoware/Myo_wipi.h>
#include "myoware/MyoFeatures.h"
#include <cmath>
using namespace std;

/* Subscriber and client node
 * Author: Virginie Gillis
   Date: 10.04.2019

*/


bool feat(myoware::MyoFeatures::Request &req, myoware::MyoFeatures::
    ↪ Response &res)
{

    int n = req.window.size();
    vector<float_t> win;
    win = req.window;

    //mean absolute value
    float_t sum = 0;
    for (int i =0; i<n; ++i)
    {
        sum = sum + win[i];
    }
    res.features.push_back(sum/n);

```

```
36      //variance
37      float_t var = 0;
38      for (int i =0; i<n; ++i)
39      {
40          sum = sum + (win[i]^2);
41      }
42      res.features.push_back(var/(n-1));
43
44      //slope sign changes
45      int sc_count = 0;
46      for (int p = 1; p<n; ++p)
47      {
48          if ( (win[p]>win[p-1] && win[p]>win[p+1]) || (win[p]<win[p-1]
            ↪ && win[p]<win[p+1]))
49          {
50              sc_count++;
51          }
52      }
53      res.features.push_back(sc_count);
54
55      //waveform length
56      float_t wave_l= 0;
57      for ( int q=0; q<n; ++q)
58      {
59          float diff = win[q+1]-win[q];
60          wave_l=wave_l + abs(diff);
61      }
62      res.features.push_back(wave_l);
63
64      return true;
65  }
66
67  void  Handler(int signo)
68  {
69      //System Exit
70      printf("\r\nEND                    \r\n");
71      endSPI();
72      exit(0);
73  }
74
75  int main(int argc, char **argv)
76  {
77      ros::init(argc, argv, "Myoware2");//name of node
78      ros::NodeHandle n;
79      ros::Publisher myo_pub = n.advertise<myoware::Myo_wipi>("/
            ↪ Prosthesis/myoware", 1);
80      ros::ServiceServer service = n.advertiseService("myoware_features",
            ↪ feat);
81      ros::Rate loop_rate(500);
82      printf("Publishing data + features server\r\n");
83
84      signal(SIGINT, Handler);
85
```

```
86
87      if (!initializeSPI()) {
88          return 1;}
89          setBuffer(1);
90          setPGA(PGA_GAIN1);
91          setDataRate(DRATE_1000);
92      while (ros::ok())
93      {
94
95          myoware::Myo_wipi::Ptr msg(new myoware::Myo_wipi);
96          uint32_t values_SE [8];
97          uint8_t channels_SE [8]= {AIN0, AIN1, AIN2, AIN3, AIN4, AIN5,
                ↪ AIN6, AIN7};
98
99          scanSEChannels(channels_SE, 8, values_SE);
100         for (int ch=0; ch<8; ++ch)
101         {
102             msg->emg_ch0.push_back((float_t)(values_SE[ch]*5.0/0x7fffff
                    ↪ ));
103         }
104         double t_ros = ros::Time::now().toSec();
105         int tdiv = (int) t_ros/1000;
106         double t = t_ros-(tdiv*1000);
107         msg->time =t;
108         myo_pub.publish(msg) ;
109
110         ros::spinOnce();
111         loop_rate.sleep();
112
113     }
114     return 0;
115 }
```

A.2 Code of the subscriber node

```
1  #include "ros/ros.h"
2  #include "std_msgs/String.h"
3  #include <myoware/Myo_wipi.h>
4  #include <myoware/Myo_train.h>
5  #include <myoware/Myo_labels.h>
6  #include <myoware/Myo_emgbag.h>
7  #include <myoware/Myo_featbag.h>
8  #include <myoware/MyoFeatures.h>
9  #include <myoware/MyoLDA.h>
10 #include <myoware/LDAscore.h>
11 #include <ros/console.h>
12 #include <time.h>
13 #include <cstdlib>
14
15 /* Subscriber and client node
16  * Author: Virginie Gillis
17    Date: 10.04.2019
```

```
18
19 */
20
21 #define N_TRAIN 9 // = N_train/class
22 #define WIND_LENGTH 150
23 #define WIND_INCR 12
24 #define N_CLASS 4
25
26
27 using namespace std;
28 vector<vector<float_t>> window;
29 vector<int> predict_labs;
30 vector<double_t> time_pred;
31 vector<int> true_labs;
32 ros::ServiceClient *featclientPtr; //pointer for client
33 ros::ServiceClient *ldaclientPtr;
34 ros::ServiceClient *scoreclientPtr;
35 ros::Publisher *bagpubPtr;
36 ros::Publisher *emgbagpubPtr;
37 ros::Publisher *featbagpubPtr;
38 int count_win_all = 0;
39 int count_win_class=0;
40 int ntest = 0;
41 int nset =1;
42 int rdy =0;
43 int NTEST;
44
45 void myocallback(const myoware::Myo_wipi::Ptr& msg)
46 {
47     myoware::MyoFeatures featsrv;
48     myoware::MyoLDA ldasrv;
49     myoware::LDAscore scoresrv;
50     myoware::Myo_labels::Ptr msglab(new myoware::Myo_labels);
51     myoware::Myo_emgbag::Ptr msgbag(new myoware::Myo_emgbag);
52     myoware::Myo_featbag::Ptr msgfeat(new myoware::Myo_featbag);
53
54     int n_feat;
55     if ( rdy ==0)
56     {
57         ros::Publisher emgbagpub = (ros::Publisher)*emgbagpubPtr;
58         msgbag->emg = msg->emg_ch0;
59         msgbag->nset= nset;
60         msgbag->time = msg->time;
61         emgbagpub.publish(msgbag);
62
63         window.push_back(msg->emg_ch0); //store each sample in a window
64
65         if (window.size() == WIND_LENGTH) //one window of 150 samples
               ↪ is completed
66         {
67             vector<float_t> features; //to store features of the 8
                   ↪ channels in one vector
68
```

```
69              for (int i=0; i<8; ++i) // to separate each channel
70              {
71                  vector<float_t> column(window.size(),0);
72                  for (int j=0; j<window.size(); ++j)
73                  {
74                      column[j] = window[j][i];
75                  }
76
77                  //call feature service
78                  featsrv.request.window = column; // input of feature
                        ↪ server= vector window of size n_samp
79                  ros::ServiceClient featclient = (ros::ServiceClient)*
                        ↪ featclientPtr; //dereference the clientptr
80                  if (featclient.call(featsrv))//request service from
                        ↪ client -> calculate features over window
81                  {
82                      n_feat = featsrv.response.features.size();
83
84                      for ( int j =0; j<n_feat; ++j){
85                          features.push_back(featsrv.response.features[j
                                ↪ ]);
86
87                      }
88                  }
89                  else{
90                      ROS_ERROR("Failed to call feature service from sub
                            ↪ node");
91                  }
92              }
93
94              double tros = ros::Time::now().toSec();
95              int t_div = (int) tros/1000;
96              ros::Publisher featbagpub = (ros::Publisher)*featbagpubPtr;
97              msgfeat->time = (tros-(t_div*1000));
98              msgfeat->feat = features;
99              featbagpub.publish(msgfeat);
100
101             //  call LDA service
102             ldasrv.request.data_sample = features;
103             vector<int> lab;
104             for ( int i = 1; i <= N_CLASS; i++) // construct y for fit
                    ↪ function in lda server
105             {
106                 for (int j =0; j<(N_TRAIN-2); j++)
107                 {
108                     lab.push_back(i);
109                 }
110             }
111             ldasrv.request.labels = lab;
112             ldasrv.request.n_win = count_win_all;
113             ldasrv.request.n_train = N_TRAIN;
114             ldasrv.request.n_class = N_CLASS;
115             ros::ServiceClient ldaclient = (ros::ServiceClient)*
```

```
                ↪ ldaclientPtr;
116             if (ldaclient.call(ldasrv))
117             {
118                 if (count_win_all >= (N_TRAIN * N_CLASS ) )
119                 {
120                     ROS_INFO("model predicted: %d", ldasrv.response.
                        ↪ classlab);
121
122                     predict_labs.push_back(ldasrv.response.classlab);
123                     double time_ros = ros::Time::now().toSec();
124                     int timediv = (int) time_ros/1000;
125                     time_pred.push_back(time_ros-(timediv*1000));
126
127                     // decomment the message publishing for transition
                        ↪ mode
128                     //ros::Publisher bagpub = (ros::Publisher)*
                        ↪ bagpubPtr;
129                      //msglab->pred_lab = predict_labs;
130                      //msglab->pred_time= time_pred;
131                     //bagpub.publish(msglab);
132
133                     count_win_all ++;
134                 }
135                 else {
136                     ROS_INFO("training %d",count_win_all);
137                     count_win_all++;
138                 }
139             }
140             else{
141                 ROS_ERROR(" Failed to call lda service from server node
                    ↪ ");
142             }
143
144             window.erase(window.begin(),window.begin()+WIND_LENGTH-
                ↪ WIND_INCR); //erase window to start a new one but
                ↪ keep wind_incr number of samples
145
146             if ( count_win_class ==( N_TRAIN -1) ){  // one class is
                ↪ trained
147                 rdy=1;
148                 nset++;
149                 if ( count_win_all >= (N_TRAIN * N_CLASS))  // test
                    ↪ part
150                 {
151                     //to test without interruption -> for transition
                        ↪ mode
152                     //count_win_class ++;
153                     // printf("testing \n");
154
155                       // to test with interruption -> for static mode
                          ↪ decomment all if statement
156                      if ( ntest >= NTEST ) //end of experience
157                      {
```

```
158                     printf("end of experience, num test = %d\n",
                            ↪ ntest);
159
160                     //call score service
161                     predict_labs.erase(predict_labs.begin()+
                            ↪ predict_labs.size() -(N_TRAIN-1));
162                     predict_labs.erase(predict_labs.begin() +
                            ↪ predict_labs.size() - 1);
163                     printf("predict_labs = %d\n", predict_labs.
                            ↪ size());
164                     for ( int i = 0 ; i < predict_labs.size(); ++i
                            ↪ )
165                     {
166                         printf("%d", predict_labs[i]);
167                     }
168                     scoresrv.request.predicted = predict_labs;
169                     scoresrv.request.truelab = true_labs;
170                     ros::ServiceClient scoreclient = (ros::
                            ↪ ServiceClient)*scoreclientPtr;
171                     if (scoreclient.call(scoresrv))
172                     {
173                         printf("\n score for this experience = %f
                                ↪ \n", scoresrv.response.score);
174                         //publish message
175                        ros::Publisher bagpub = (ros::Publisher)*
                               ↪ bagpubPtr;
176                        msglab->pred_lab = predict_labs;
177                        msglab->pred_time= time_pred;
178                        msglab->true_lab = true_labs;
179                        msglab->score = scoresrv.response.score;
180                        bagpub.publish(msglab);
181                     }
182                     else{
183                        ROS_ERROR(" Failed to call lda service from
                               ↪  server node");
184                     }
185                     // to continue testing without recalibration
186                     ntest=1;
187                     predict_labs.clear();
188                     count_win_class =0;
189
190                 }
191                 else
192                 {
193                     printf("next class to test \n"); // idem
194                     if ( ntest>0)
195                     {
196                        predict_labs.erase(predict_labs.begin()+
                               ↪ predict_labs.size() -(N_TRAIN-1));
197                        predict_labs.erase(predict_labs.begin() +
                               ↪ predict_labs.size() - 1);
198                     }
199
```

```
200                         count_win_class =0;
201                         ntest++;
202                     }   //comment until here for transition mode
203                 }
204                 else // training part
205                 {
206                     count_win_class=0;
207                     printf("next class to train \n");
208                 }
209
210             }
211             else{
212                 rdy=0;
213                 count_win_class++;
214             }
215         } // end completed window
216
217     }
218     else //rdy =1, have to wait for signal
219     {
220         //printf("waiting for signal\n");
221     }
222
223 }
224 void callback(const myoware::Myo_train::Ptr& msg)
225 {
226     rdy = msg->ready; // get the signal from user
227
228 }
229
230 int main(int argc, char **argv)
231 {
232     ros::init(argc, argv, "Myoware_sub");
233     ros::NodeHandle n;
234
235     ros::ServiceClient featclient = n.serviceClient<myoware::
        ↪ MyoFeatures>("myoware_features"); //create client
236     featclientPtr = &featclient; //give the address of the client to
        ↪ the clientPtr pointer
237
238     ros::ServiceClient ldaclient = n.serviceClient<myoware::MyoLDA>("
        ↪ lda");
239     ldaclientPtr = &ldaclient;
240
241     ros::ServiceClient scoreclient = n.serviceClient<myoware::LDAscore
        ↪ >("score");
242     scoreclientPtr = &scoreclient;
243
244     ros::Publisher bagpub = n.advertise<myoware::Myo_labels>("/
        ↪ Prosthesis/baglab", 1);
245     bagpubPtr = &bagpub;
246
247     ros::Publisher emgbagpub = n.advertise<myoware::Myo_emgbag>("/
```

```
        ↪ Prosthesis/bagemg", 1);
248     emgbagpubPtr = &emgbagpub;
249
250     ros::Publisher featbagpub = n.advertise<myoware::Myo_featbag>("/
        ↪ Prosthesis/bagfeat", 1);
251     featbagpubPtr = &featbagpub;
252
253     NTEST=atoll(argv[1]);
254     for ( int i = 2; i<=NTEST+1; i++)
255     {
256         for ( int j = 0 ; j<N_TRAIN-2; j++)
257         {
258             true_labs.push_back(atoll(argv[i]));
259         }
260     }
261
262     ros::Subscriber sub = n.subscribe ("/Prosthesis/myoware", 1,
        ↪ myocallback);
263     ros::Subscriber sub_train = n.subscribe ("/Prosthesis/train", 1,
        ↪ callback);
264
265     ros::spin();
266
267     return 0;
268 }
```

A.3 Code of the LDA service node

```
1  #!/usr/bin/env python
2  #Author : Virginie Gillis
3  #Date : 20/04/2019
4
5  from myoware.srv import *
6  import rospy
7  import os
8  import numpy as np
9  from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
10
11 X = np.array([])
12 Xnew = np.array([])
13 clf=LinearDiscriminantAnalysis()
14 def lda(req):
15     global X
16     global Xnew
17     global clf
18     if req.n_win < (req.n_train*req.n_class):
19         print ("trainig")
20         if req.n_win == 0:
21             X = np.hstack((X,req.data_sample))
22         else:
23             X = np.vstack((X,req.data_sample))
24
```

```
25          return 0
26      elif req.n_win == (req.n_train*req.n_class) :
27          print ("fit")
28          print("X: ",X)
29          Xnew = np.delete(X,[0,req.n_train-1, req.n_train, (2*req.
                ↪ n_train)-1,2*req.n_train,(3*req.n_train)-1,3*req.n_train
                ↪ ,(4*req.n_train)-1,4*req.n_train,(5*req.n_train)-1,5*req
                ↪ .n_train,(6*req.n_train)-1],axis=0)
30          print ("Xnew :", Xnew)
31          clf.fit(Xnew, req.labels)
32
33          print("predict")
34          lab = clf.predict([req.data_sample])
35          print(lab)
36          return lab
37      else :
38          print("predict")
39          lab = clf.predict([req.data_sample])
40          print(lab)
41          return lab
42
43
44  def lda_server():
45      print ("lda server launched")
46      rospy.init_node('lda_server')
47      s = rospy.Service('lda', MyoLDA, lda)
48      rospy.spin()
49
50  if __name__ == "__main__":
51      lda_server()
```

A.4 Code of the score service node

```
1   #!/usr/bin/env python
2   #Author : Virginie Gillis
3   #Date : 20/04/2019
4
5   from myoware.srv import *
6   import rospy
7   import os
8   import numpy as np
9   from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
10  from sklearn.metrics import accuracy_score
11
12  def score(req):
13      score = accuracy_score(req.truelab, req.predicted)
14      return score
15
16  def lda_score_srv():
17      print ("lda score server launched")
18      rospy.init_node('lda_score_srv')
19      s = rospy.Service('score', LDAscore, score)
```

```
20      rospy.spin()
21
22  if __name__ == "__main__":
23      lda_score_srv()
```

Appendix B

Guidelines to find the muscles of the thigh

In order to correctly find the muscles of the thigh, here are some guidelines to follow. First the movements for which the muscle is responsible have to be identified. The subject is then ask to perform this movements (green arrows on the illustrations) and as aid, a resistance can be applied in opposition to this movement (red arrows on the illustrations).

B.1 Quadriceps

The quadriceps are one of the most easiest group of muscles to find with the hamstrings. Their main function are :

- Knee extension
- Hip flexion

To located them, one can start from the knee patella. The subject is asked to perform a knee extension and a hip flexion at the same time as shown in figure B.1. Starting from the patella, one can feel the rectus femoris activating in the continuity of the patella axe. The vastus lateralis is then the activated muscle on the lateral side with respect to the rectus femoris.

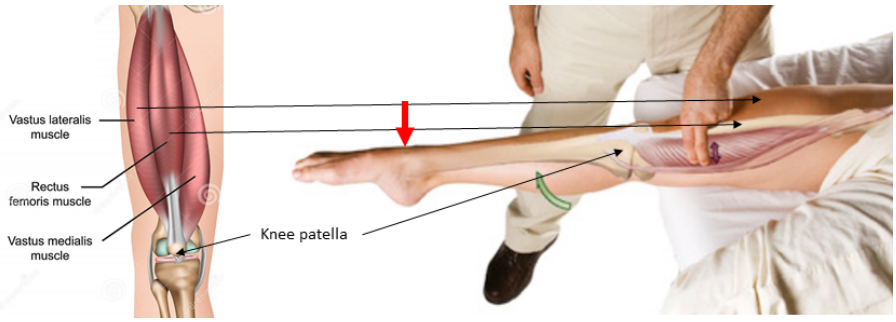


Figure B.1: Palpation of the quadriceps

B.2 Sartorius

The main functions of the sartorius are :

- lateral rotation of the hip and the knee
- hip flexion
- hip abduction
- knee flexion

The subject is asked to perform a hip lateral rotation and a hip flexion as shown in figure B.2 and one can start from the iliac spine anterior superior which is shown by a blue circle in the figure. Another way to find it easily is to sit in a cross-legged position and push on the knee to create a resistance as shown in figure B.3.

Figure B.2: Palpation of the sartorius

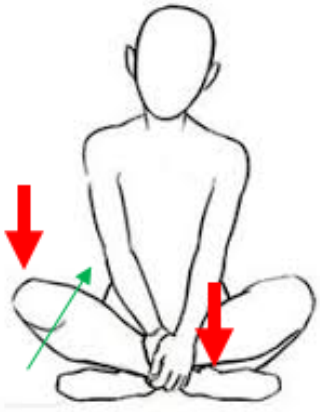


Figure B.3: Cross-legged sitting to find the sartorius

B.3 Tensor fascia latae

The main functions of the TFL are:

- medial rotation of the hip
- hip flexion
- hip abduction

The subject is asked to perform a hip flexion and medial rotation, a resistance on the knee as shown in figure B.4 can be applied to help.

Figure B.4: Palpation of the tensor fascia latae

B.4 Hamstrings

The common functions of the hamstrings are :

- hip extension
- knee flexion

The subject is asked to perform a hip extension together with a knee flexion, the biceps femoris and the semitendinosus will the be activated. They can be differentiated by their location: the BF in located on the lateral side and the SMT on the medial side as shown in figure B.5 and B.6. Moreover, the subject can also be asked to perform a lateral rotation of the knee during which the SMT will not be activated and vice versa, a medial rotation of the knee can be performed to activate only the SMT and not the BF.

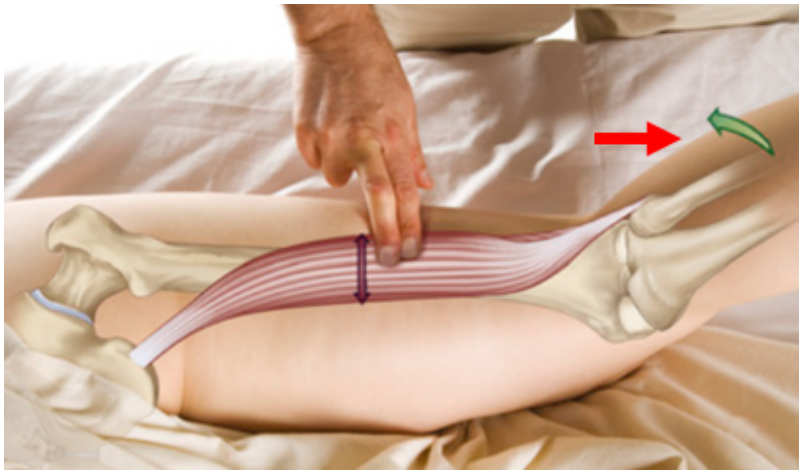


Figure B.5: Palpation of the biceps femoris

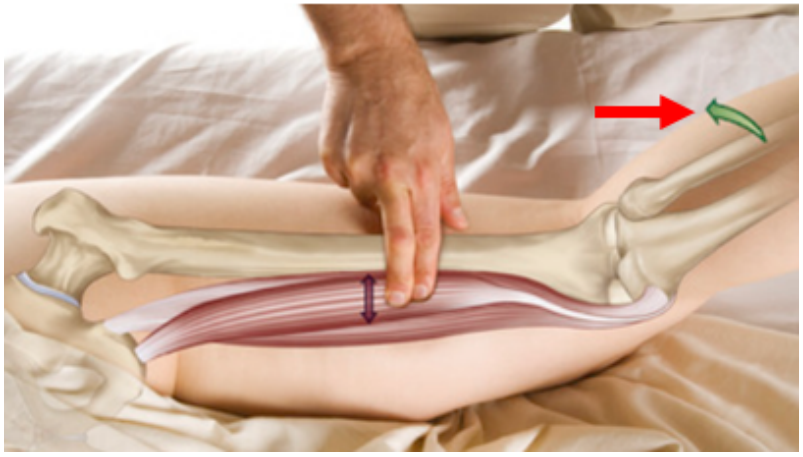


Figure B.6: Palpation of the semitendinosus

B.5 Adductor magnus

The main functions of the AM are:

- hip adduction

Once the SMT and the BF has be found, one can slide more in the medial direction and find the adductor magnus in between the SMT ad the gracilis. The subject can be asked to perform a knee flexion as shown in figure B.7. Indeed this action will not activate the AM but will activate the ST and the GRA that surround the AM. Another way to find it is to ask the subject to perform an adduction of the hip while standing as shown in figure B.8.

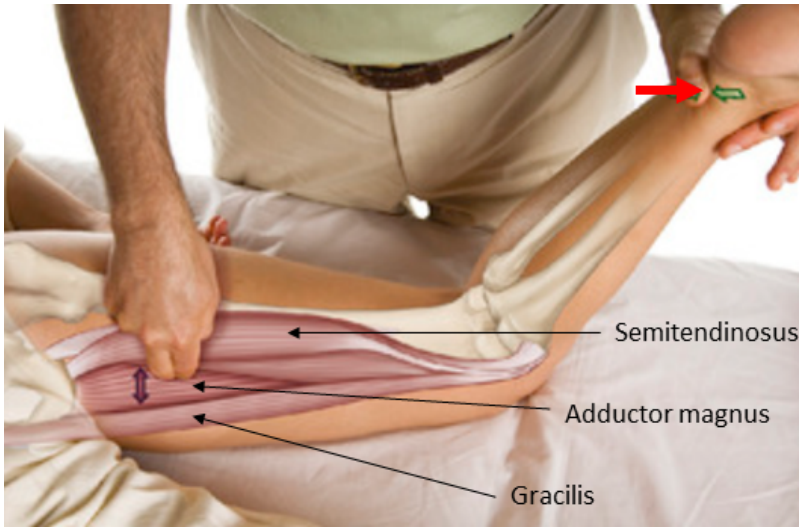


Figure B.7: Palpation of the adductor magnus

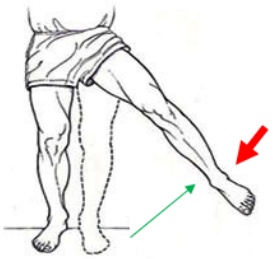


Figure B.8: Palpation of the adductor magnus

B.6 Gracilis

The gracilis is more difficult to find given its thin shape as shown in figure B.7. Its main function are :

- hip abduction
- lateral rotation of the hip
- knee flexion

To activate the gracilis the subject is asked to perform a lateral rotation of the hip and a hip adduction as shown in figure B.9

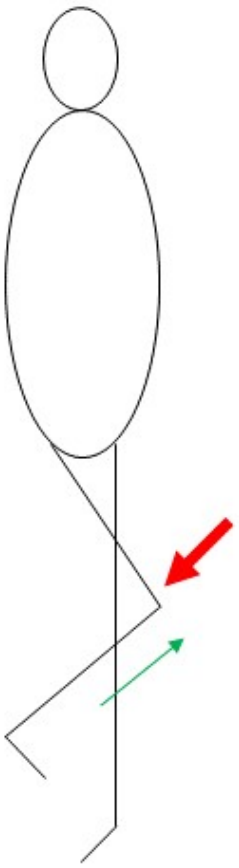


Figure B.9: Palpation of the gracilis

Appendix C

Results - EMG graphs

Subject 1 - Trained and tested at 1.1667[Hz] - 4 tasks - 8 muscles

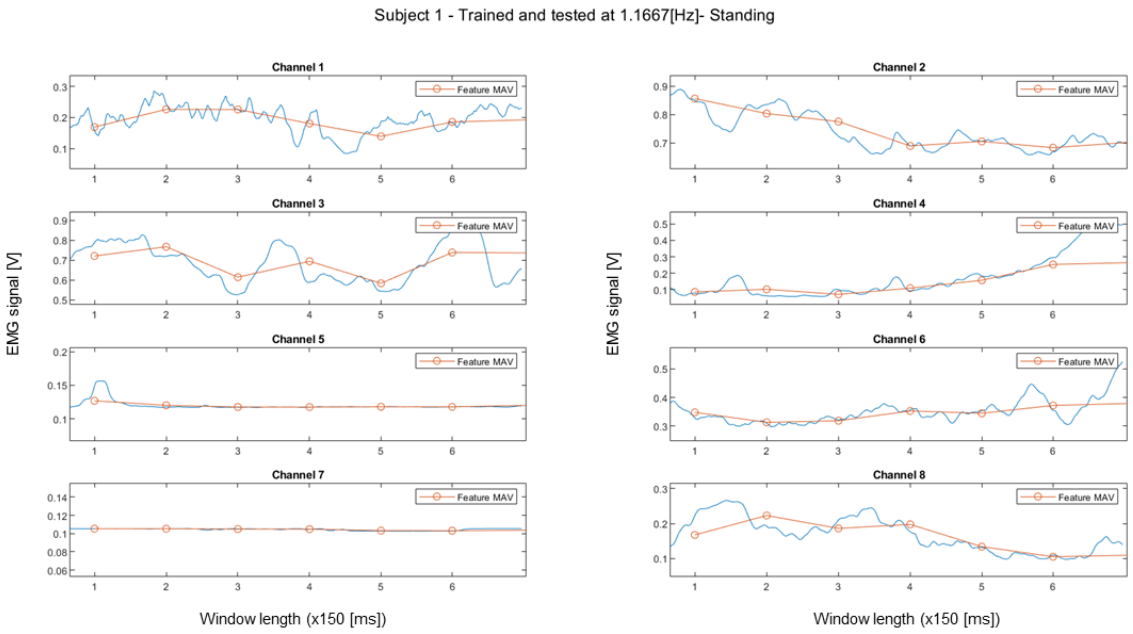


Figure C.1: EMG signals recorded during the experiment with subject 1, trained and tested at 1.1667[Hz], during the standing task

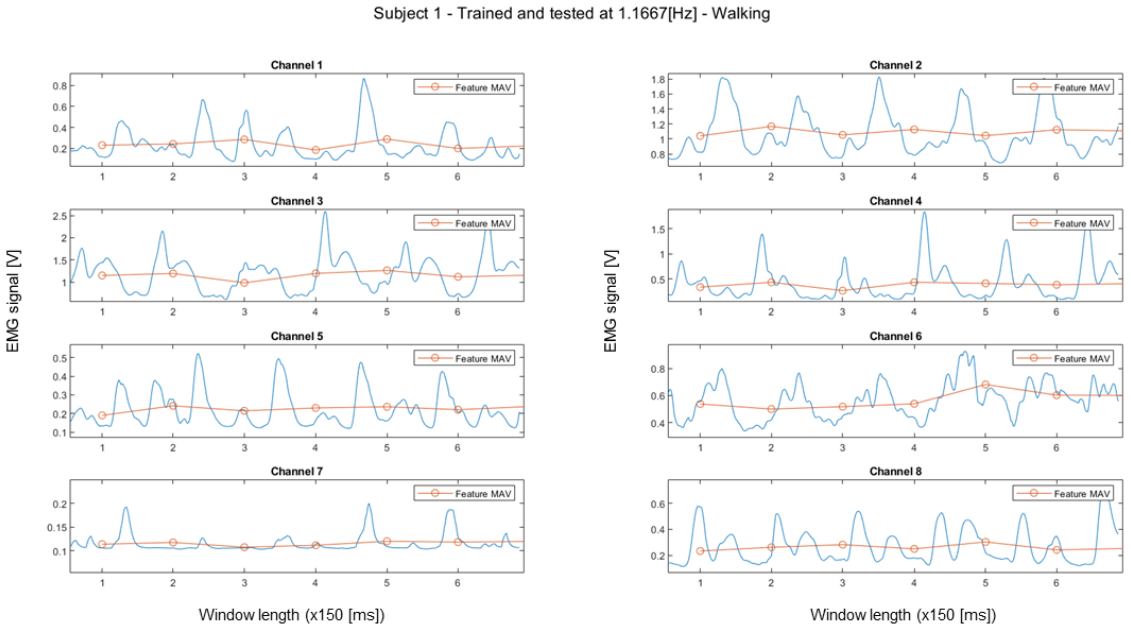


Figure C.2: EMG signals recorded during the experiment with subject 1, trained and tested at 1.1667[Hz], during the walking task

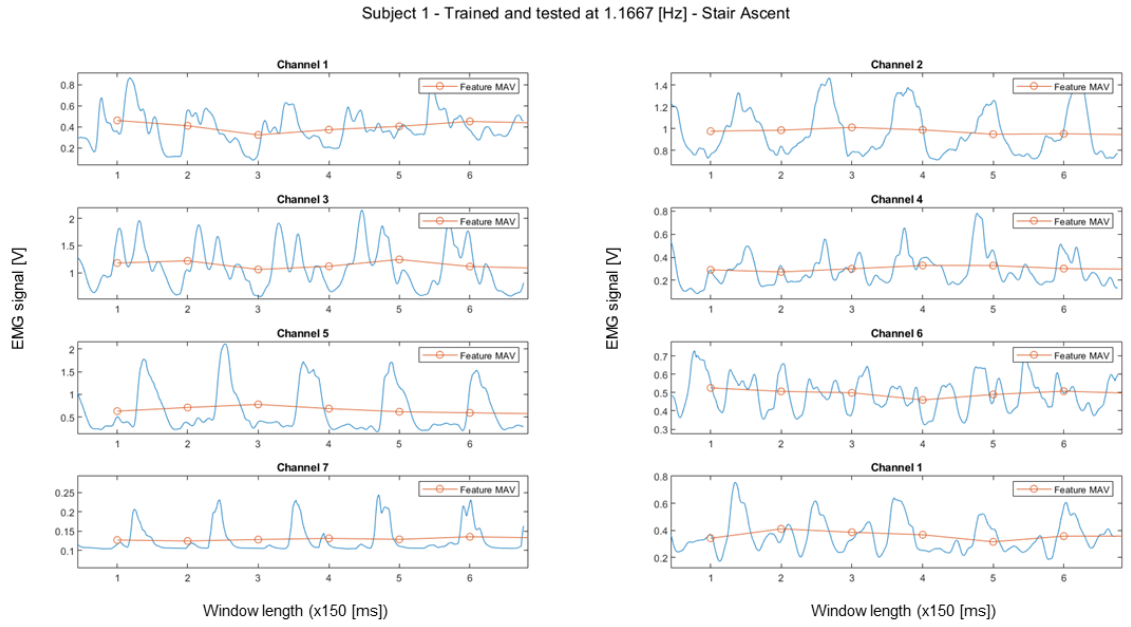


Figure C.3: EMG signals recorded during the experiment with subject 1, trained and tested at 1.1667[Hz], during the stair ascent task

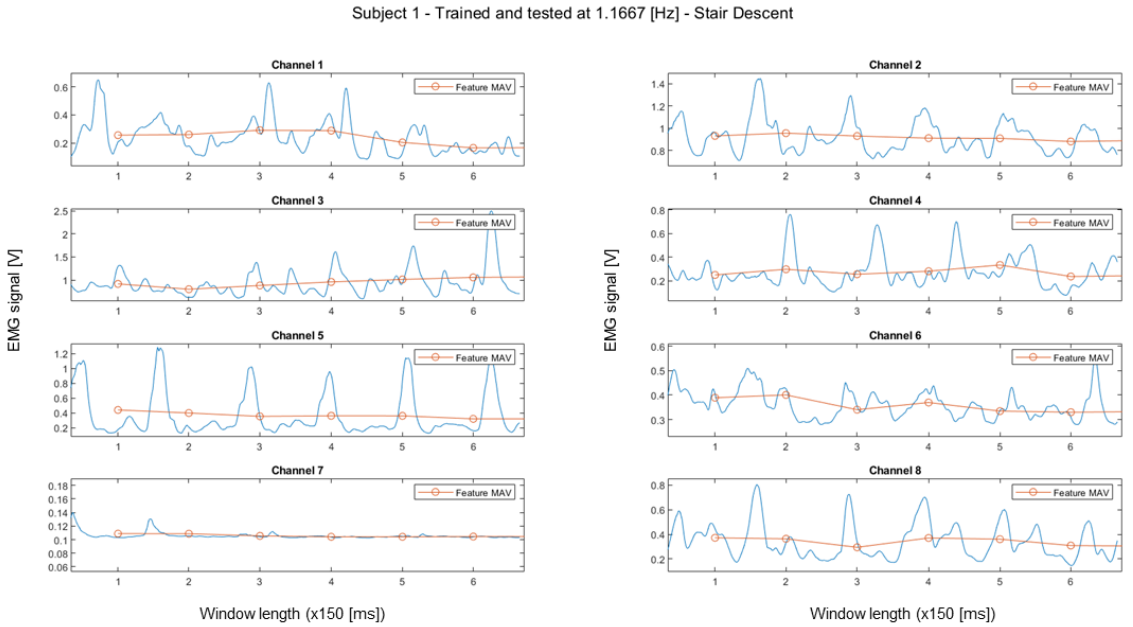


Figure C.4: EMG signals recorded during the experiment with subject 1, trained and tested at 1.1667[Hz], during the stair descent task

Subject 2 - Trained and tested at 1.1667[Hz] - 4 tasks - 7 muscles

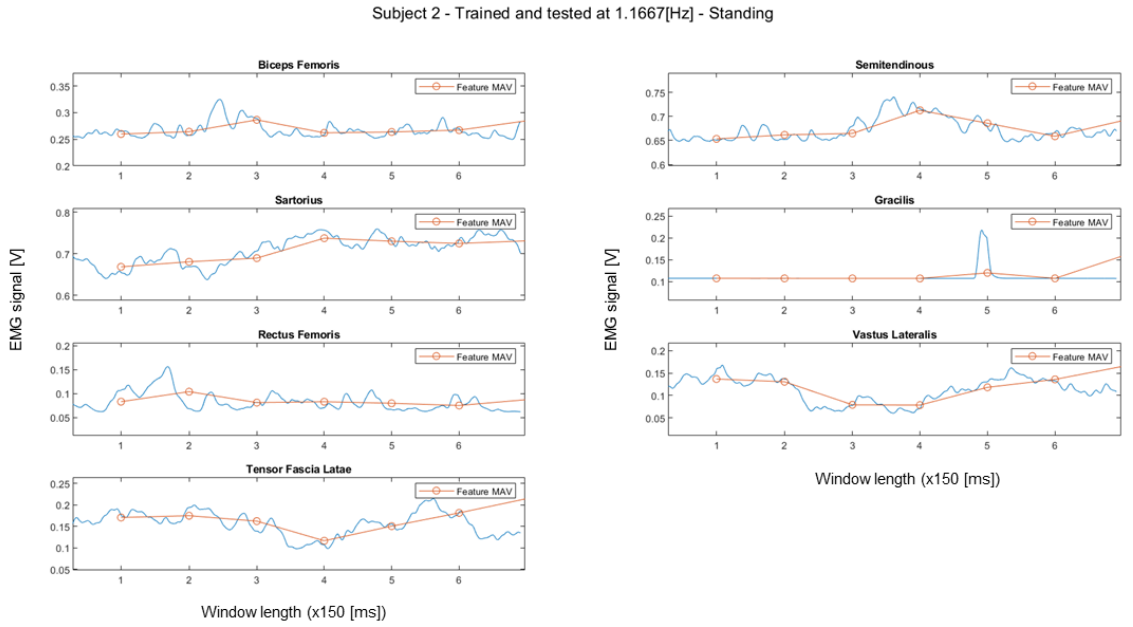


Figure C.5: EMG signals recorded during the experiment with subject 2, trained and tested at 1.1667[Hz], during the standing task

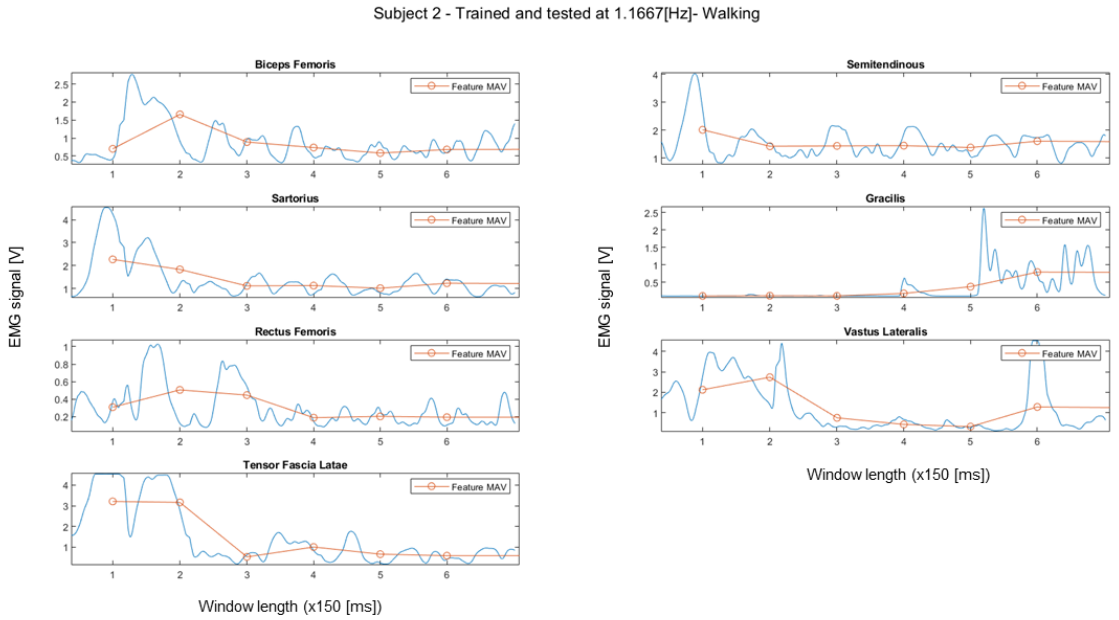


Figure C.6: EMG signals recorded during the experiment with subject 2, trained and tested at 1.1667[Hz], during the walking task

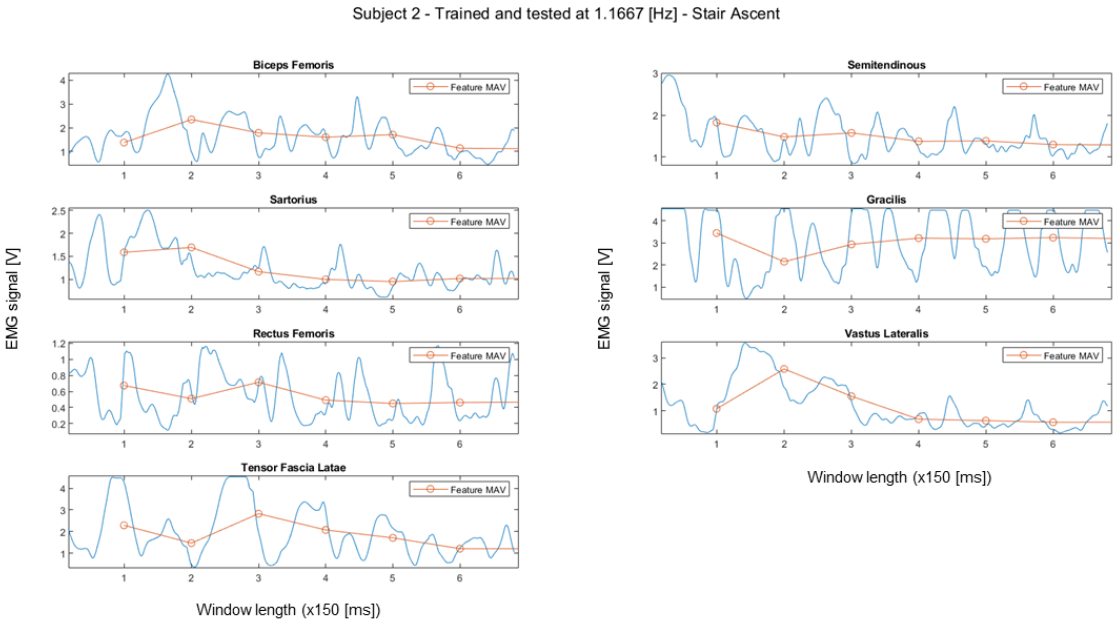


Figure C.7: EMG signals recorded during the experiment with subject 2, trained and tested at 1.1667[Hz], during the stair ascent task

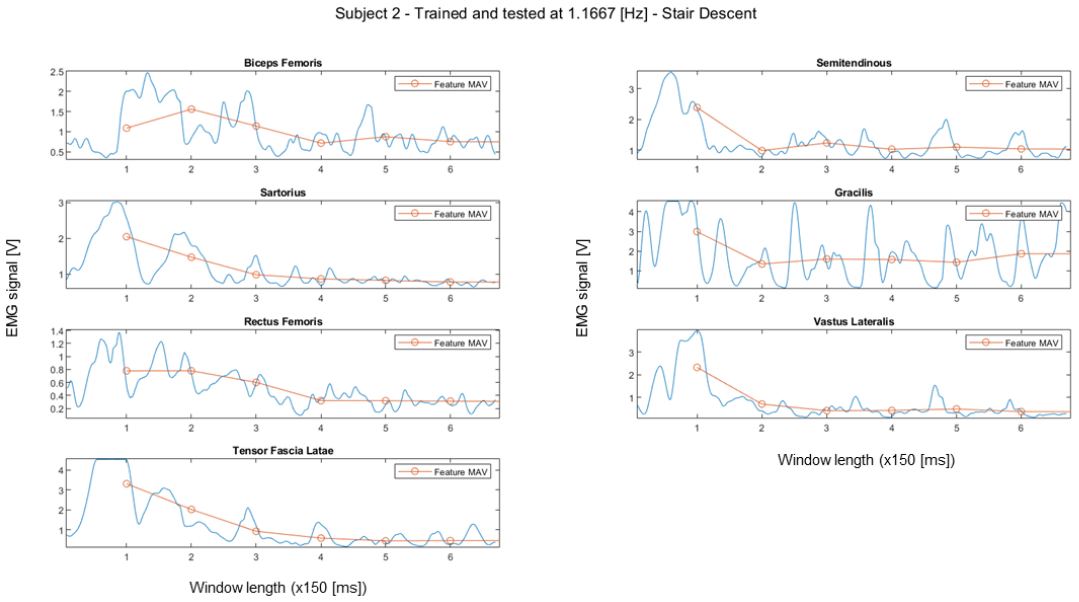


Figure C.8: EMG signals recorded during the experiment with subject 2, trained and tested at 1.1667[Hz], during the stair descent task

Subject 2 - Trained and tested at 1.5[Hz] - 4 tasks - 7 muscles

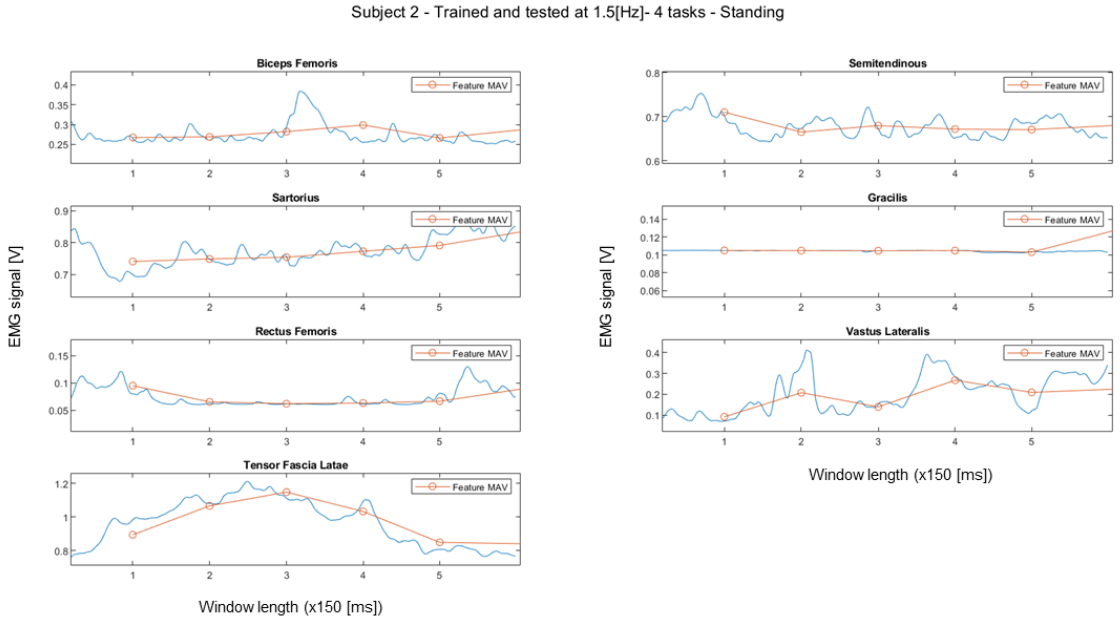


Figure C.9: EMG signals recorded during the experiment with subject 2, trained and tested at 1.5[Hz], during the standing task

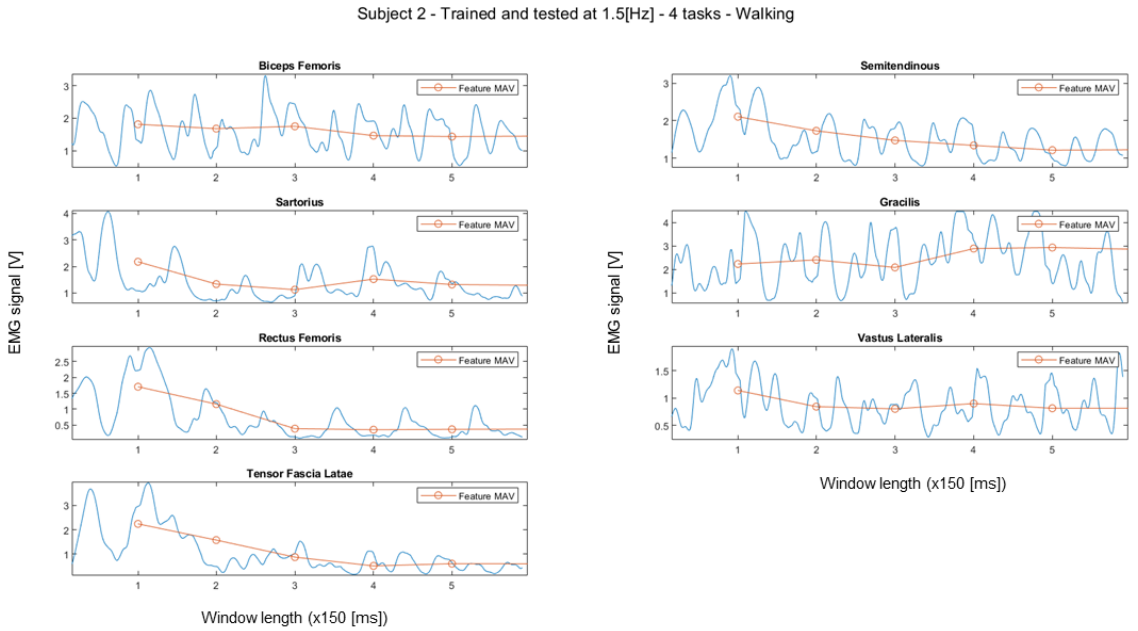


Figure C.10: EMG signals recorded during the experiment with subject 2, trained and tested at 1.5[Hz], during the walking task

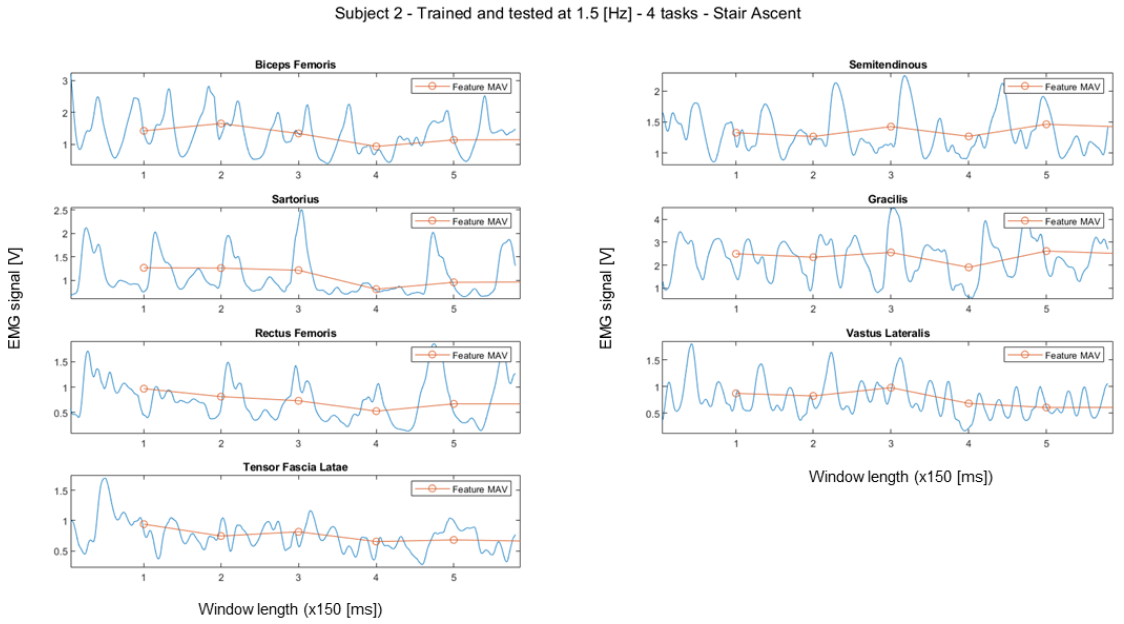


Figure C.11: EMG signals recorded during the experiment with subject 2, trained and tested at 1.5[Hz], during the stair ascent task

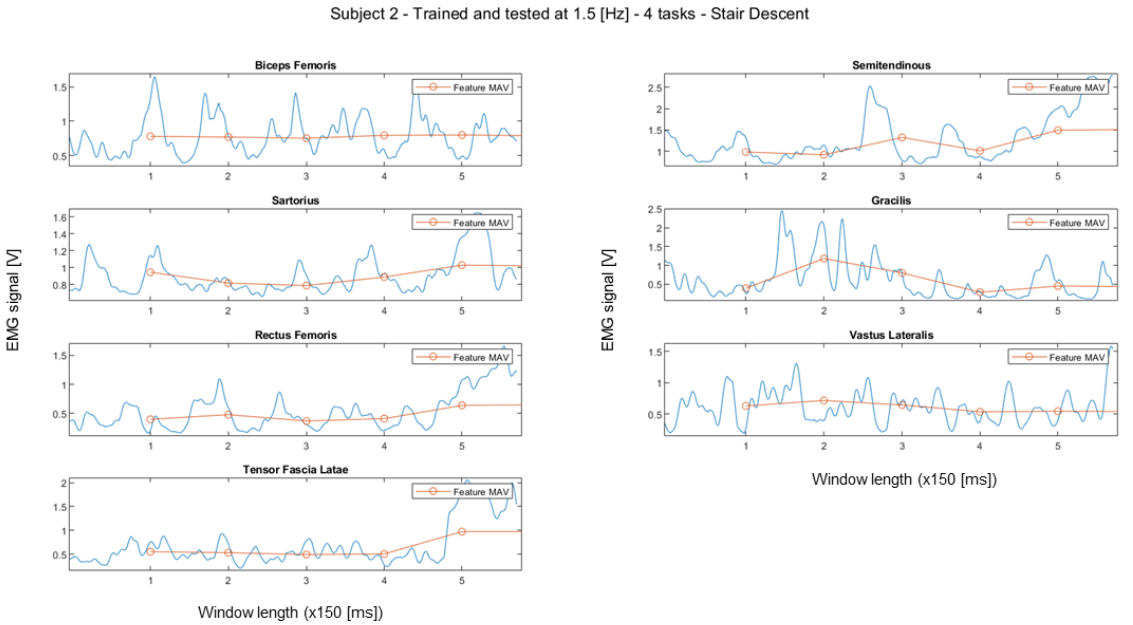


Figure C.12: EMG signals recorded during the experiment with subject 2, trained and tested at 1.5[Hz], during the stair descent task

Subject 3 - Trained and tested at 1.1667[Hz] - 6 tasks - 7 muscles

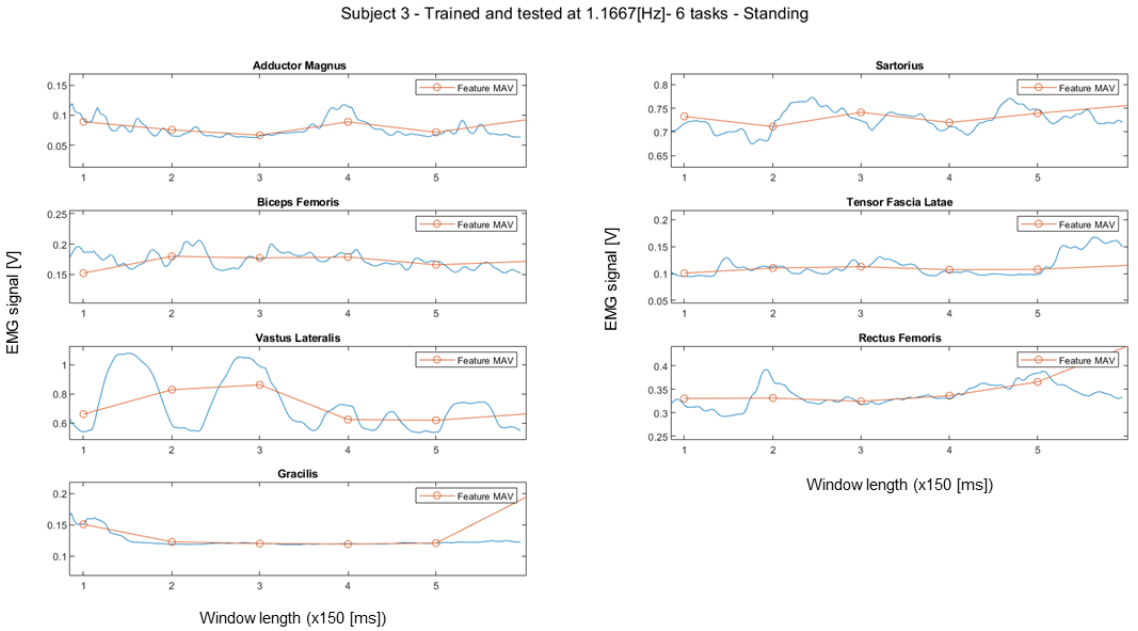


Figure C.13: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the standing task

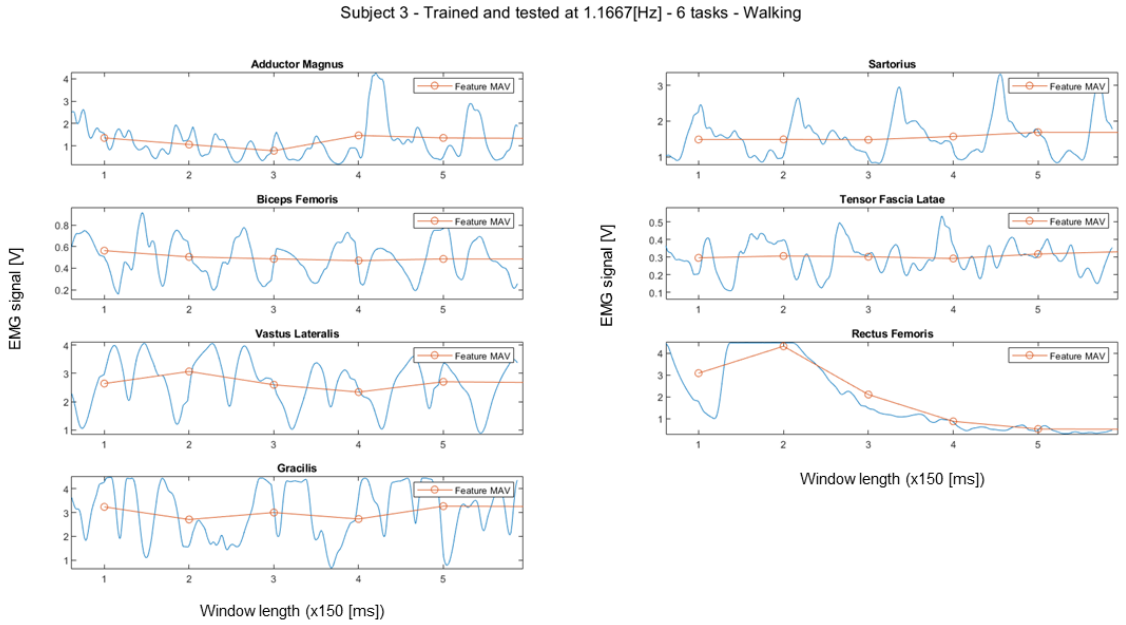


Figure C.14: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the walking task

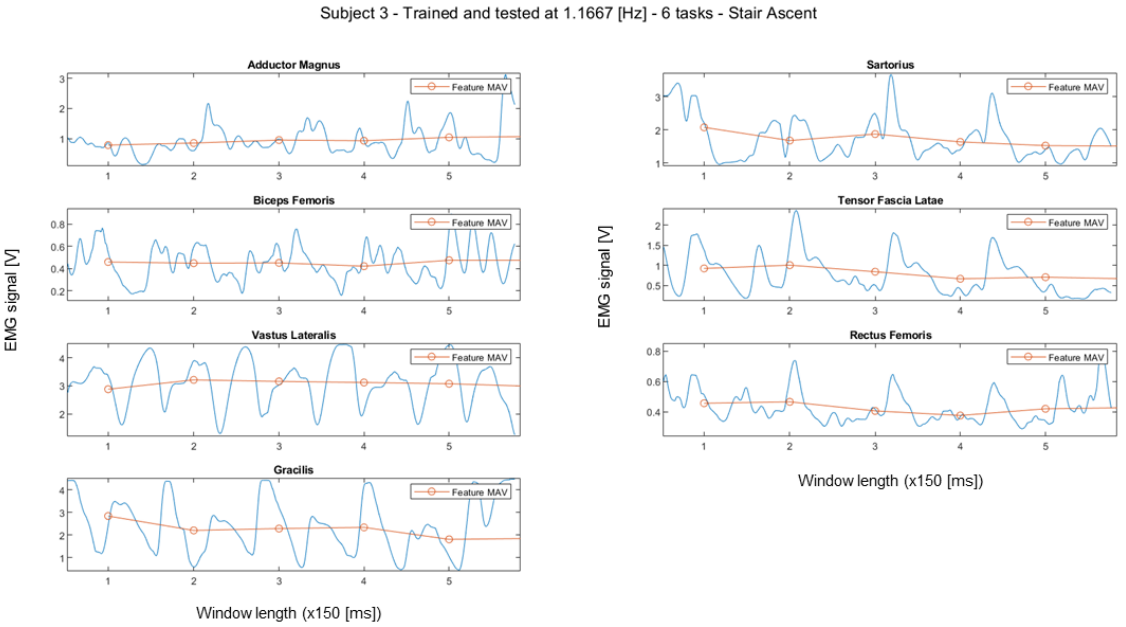


Figure C.15: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the stair ascent task

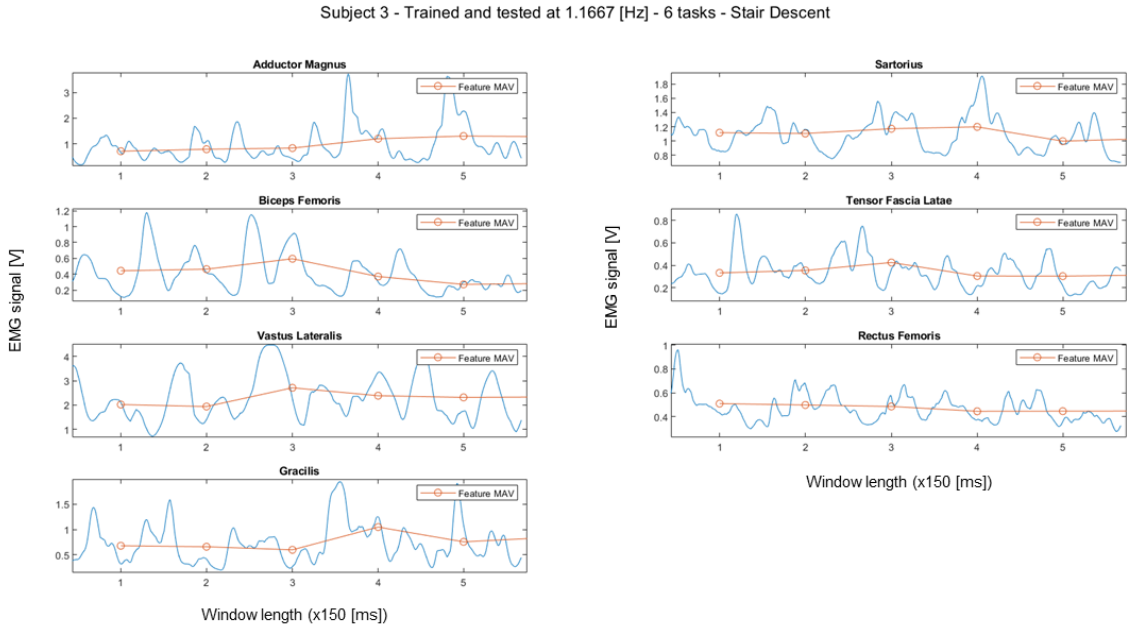


Figure C.16: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the stair descent task

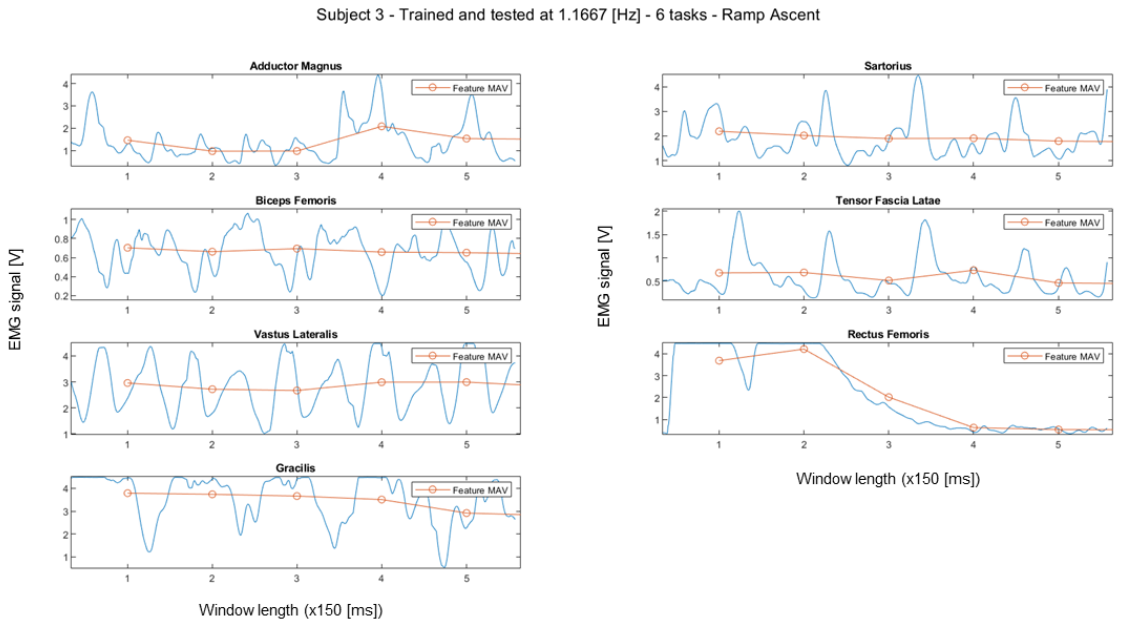


Figure C.17: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the ramp ascent task

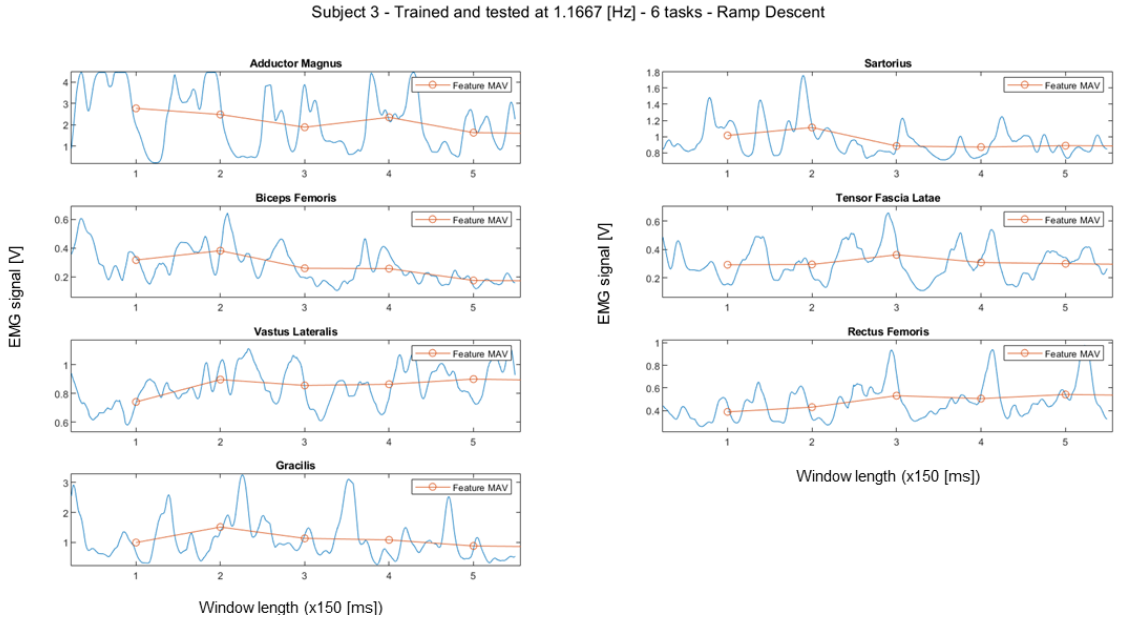


Figure C.18: EMG signals recorded from 7 muscles during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz], during the ramp descent task

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 7 muscles

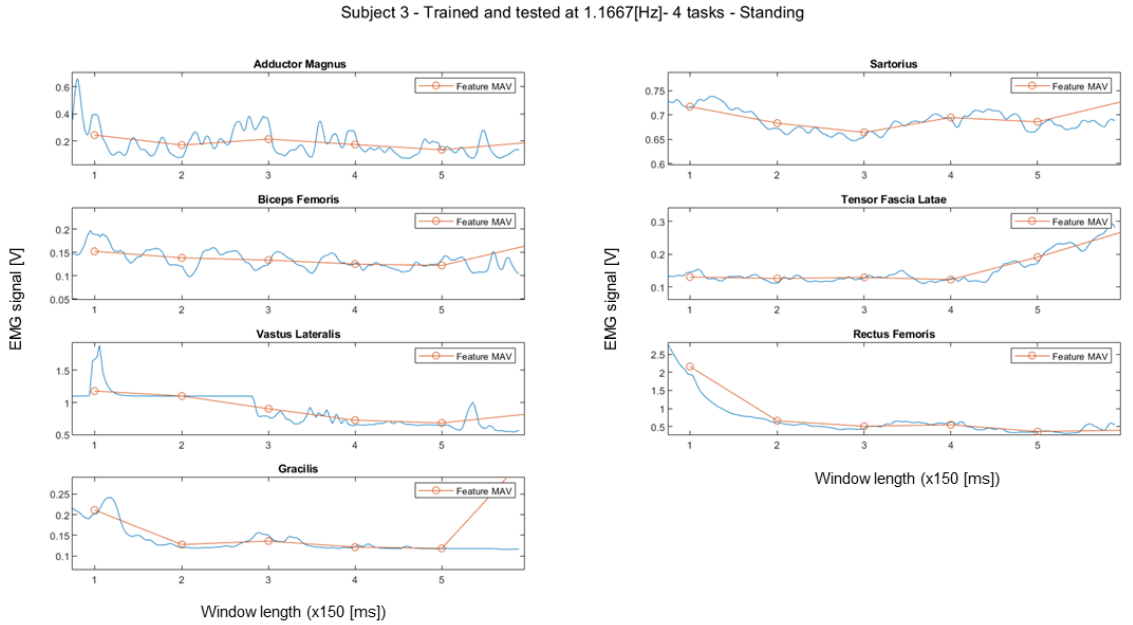


Figure C.19: EMG signals recorded from 7 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the standing task

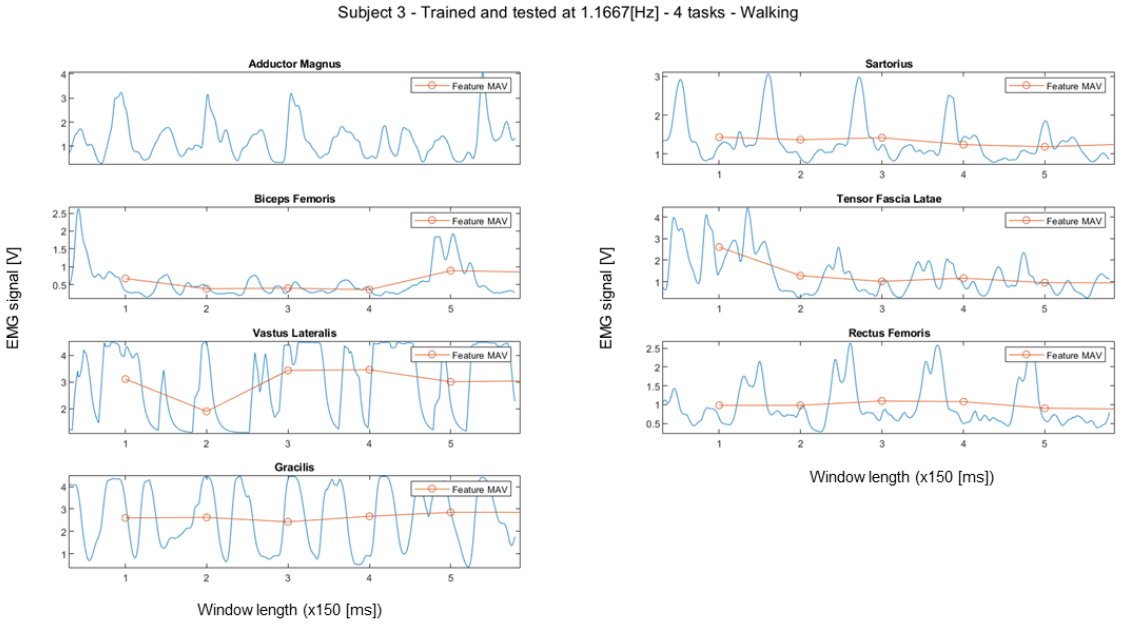


Figure C.20: EMG signals recorded from 7 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the walking task

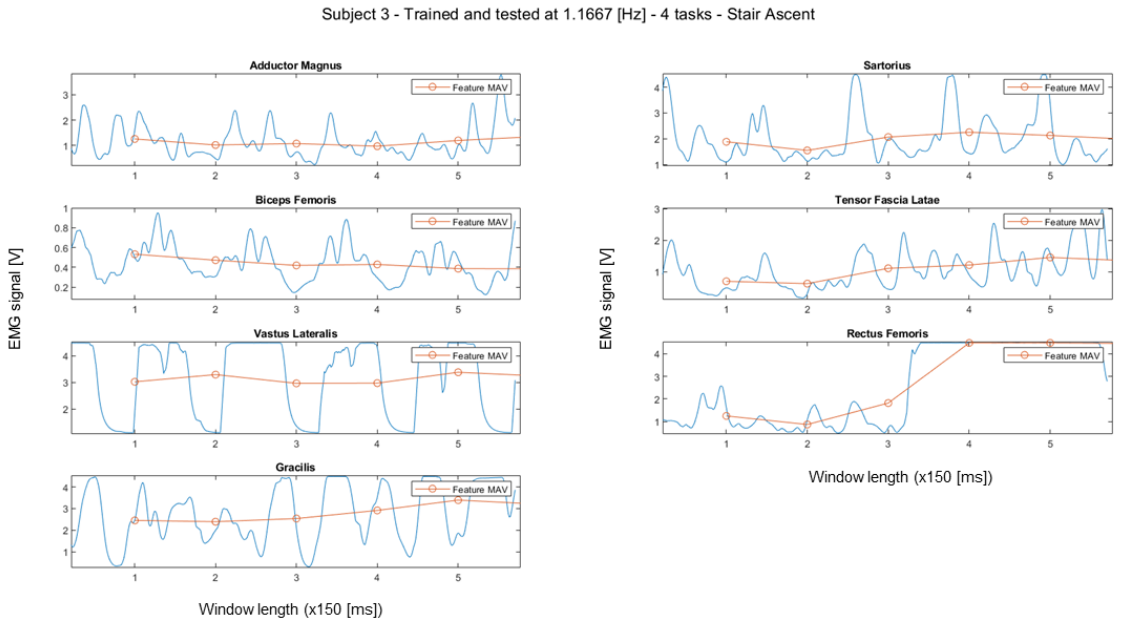


Figure C.21: EMG signals recorded from 7 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair ascent task

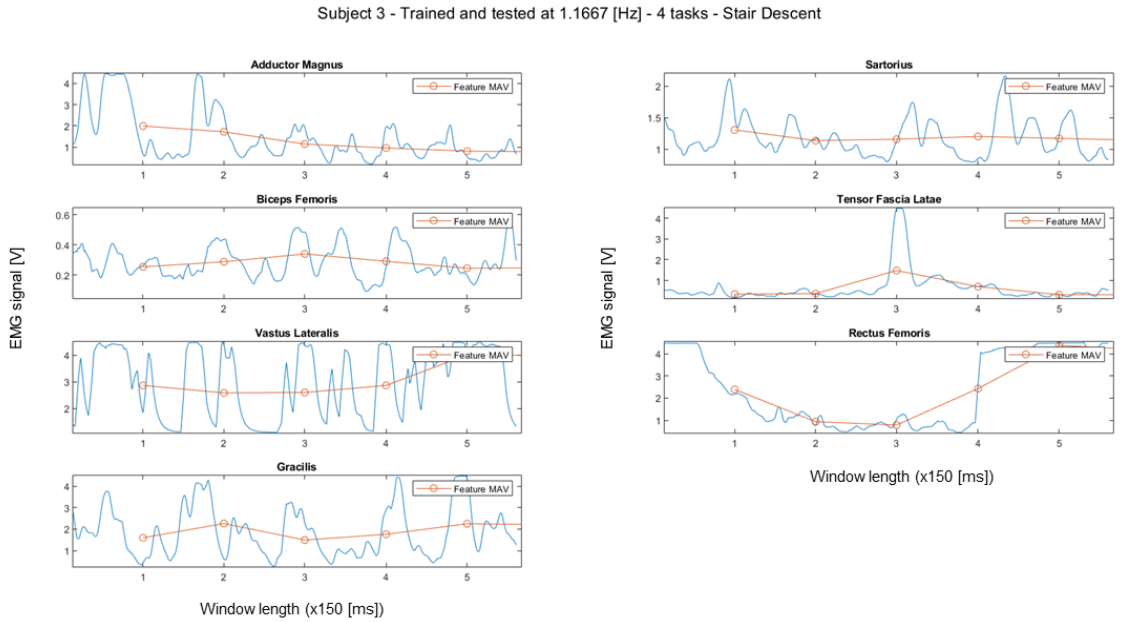


Figure C.22: EMG signals recorded from 7 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair descent task

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 5 muscles

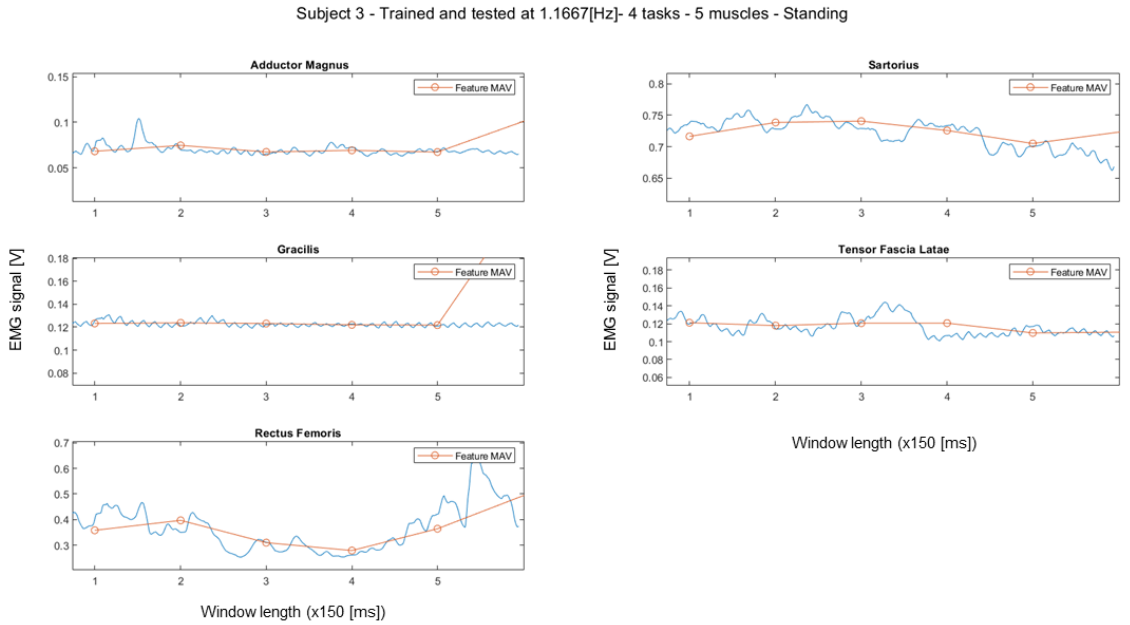


Figure C.23: EMG signals recorded from 5 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the standing task

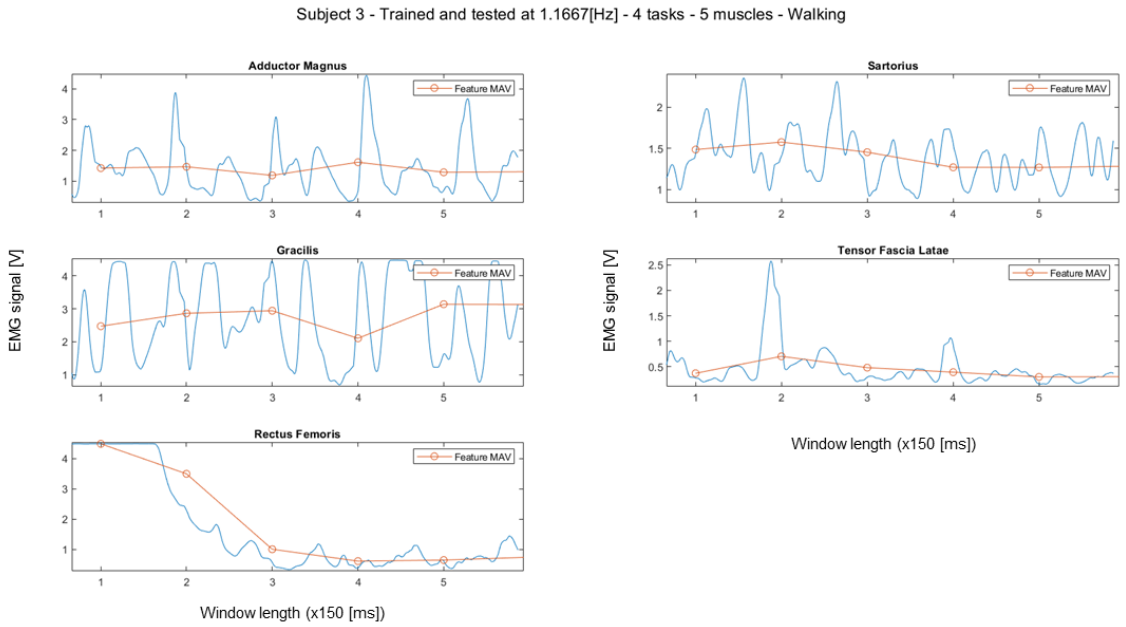


Figure C.24: EMG signals recorded from 5 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the walking task

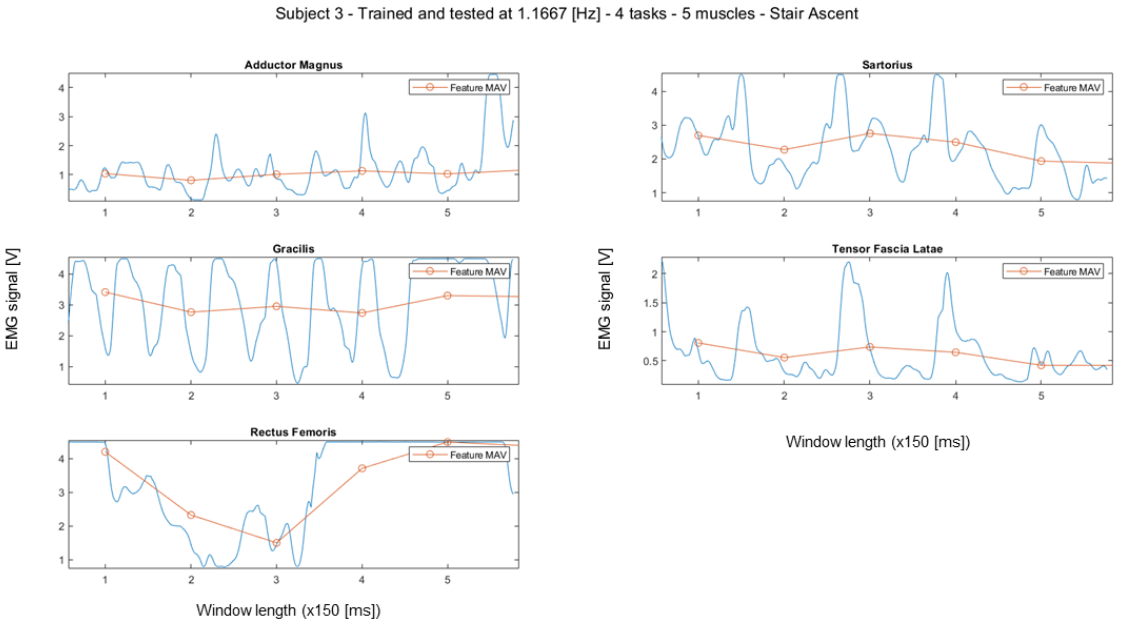


Figure C.25: EMG signals recorded from 5 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair ascent task

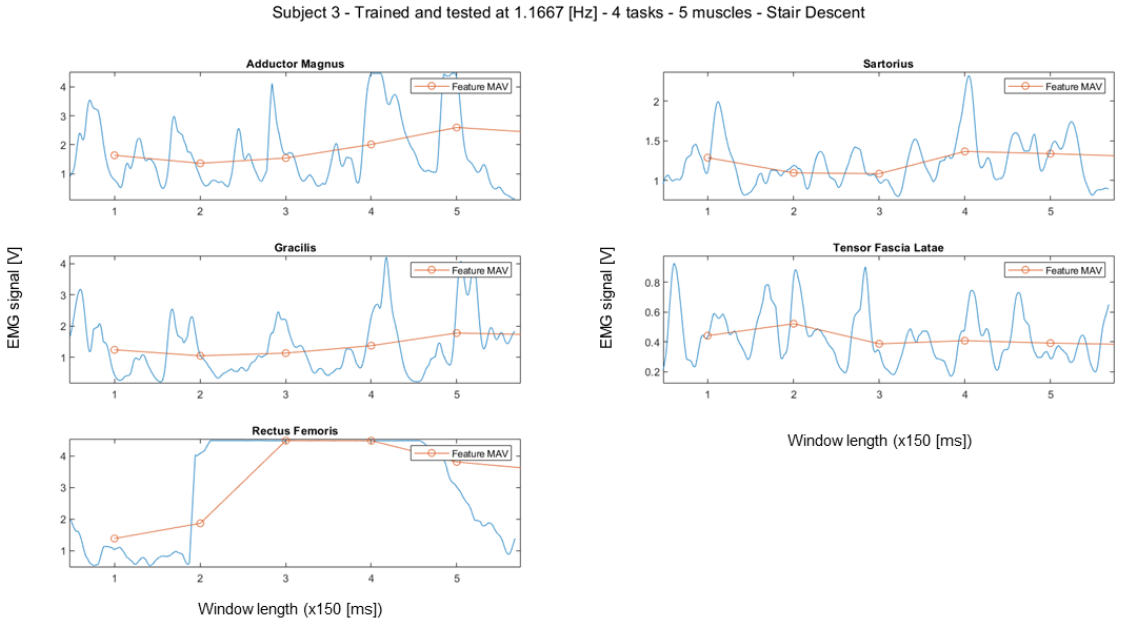


Figure C.26: EMG signals recorded from 5 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair descent task

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 3 muscles

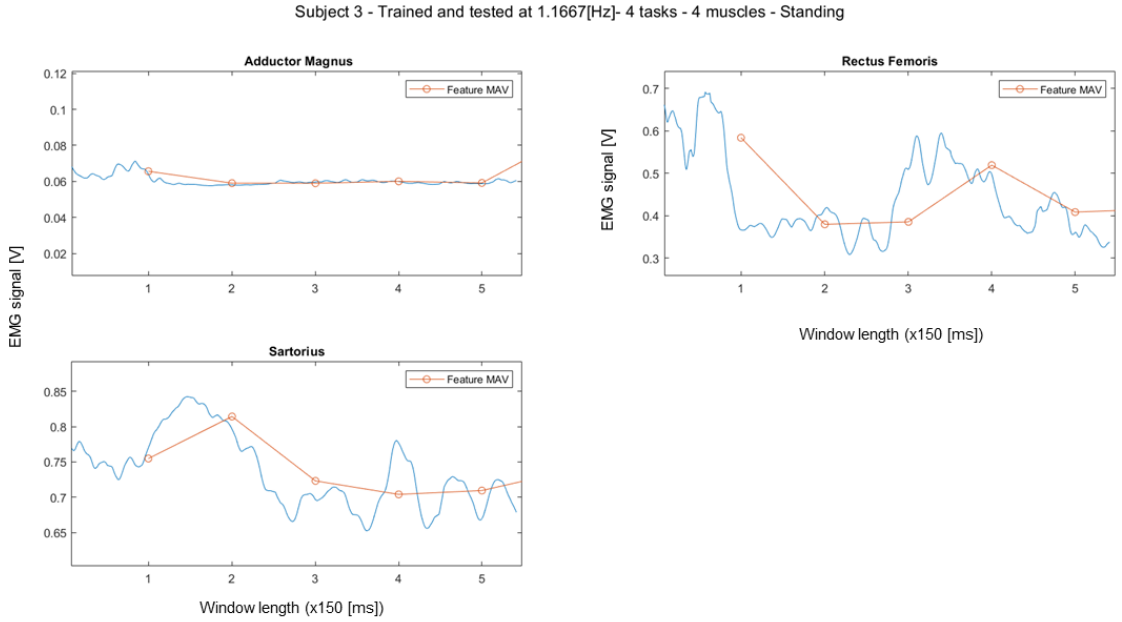


Figure C.27: EMG signals recorded from 3 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the standing task

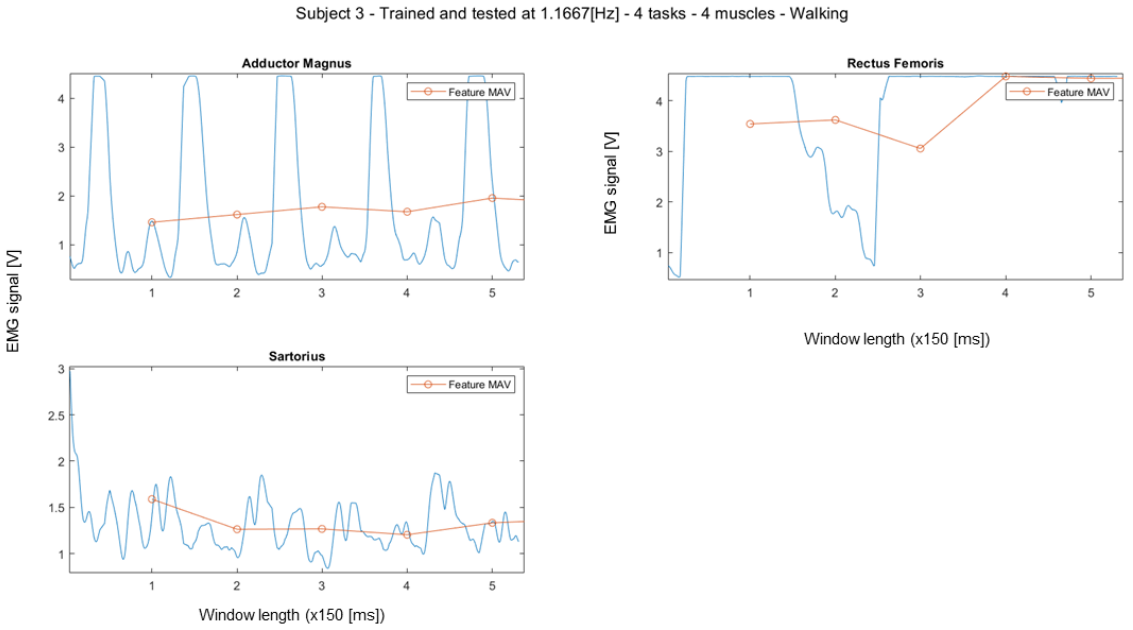


Figure C.28: EMG signals recorded from 3 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the walking task

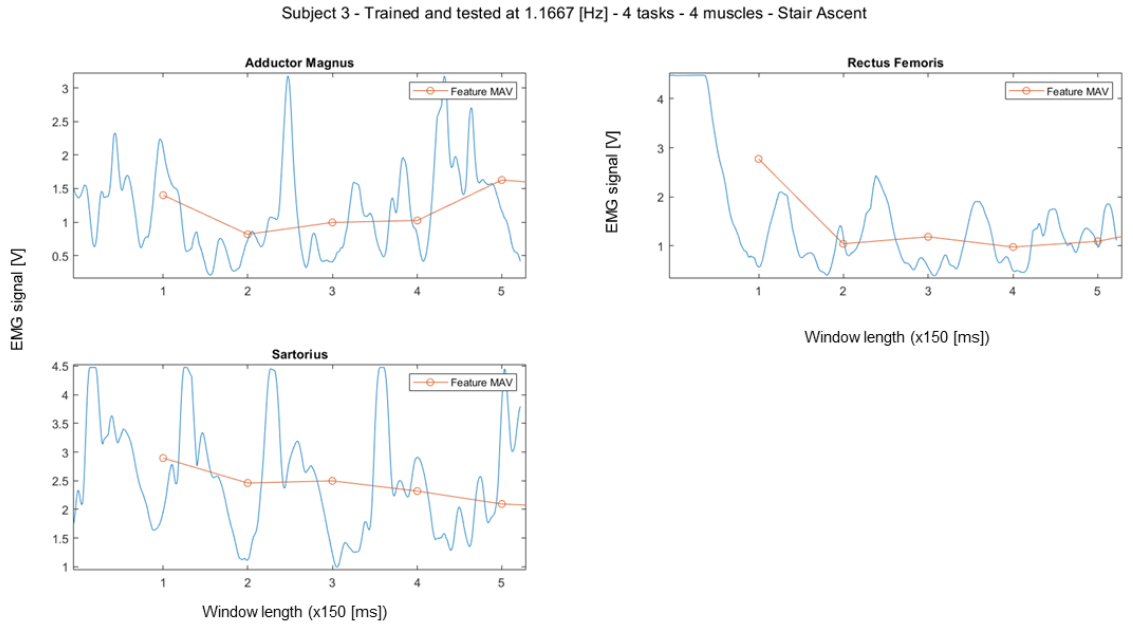


Figure C.29: EMG signals recorded from 3 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair ascent task

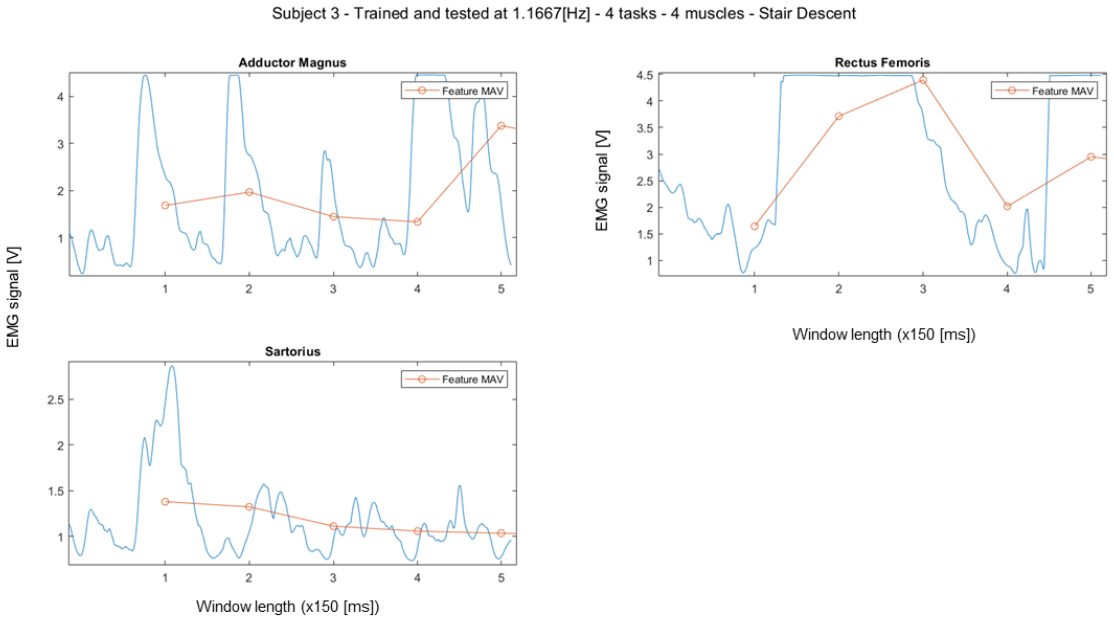


Figure C.30: EMG signals recorded from 3 muscles during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz], during the stair descent task

Appendix D

Results - Features graphs

Subject 1 - 4 tasks - 8 muscles - Varied cadence

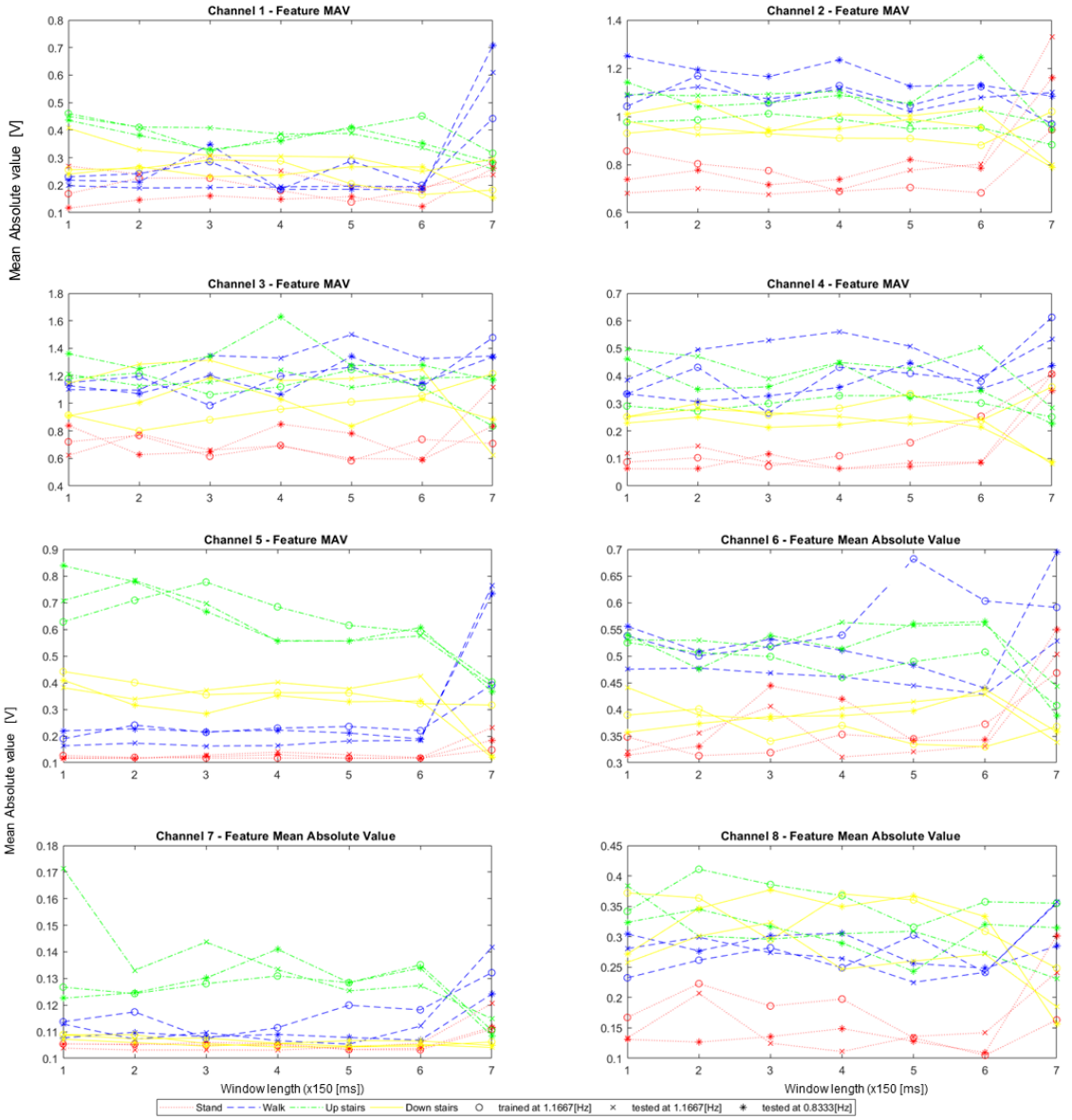


Figure D.1: Mean absolute value of the EMG signals computed during the experiment with subject 1

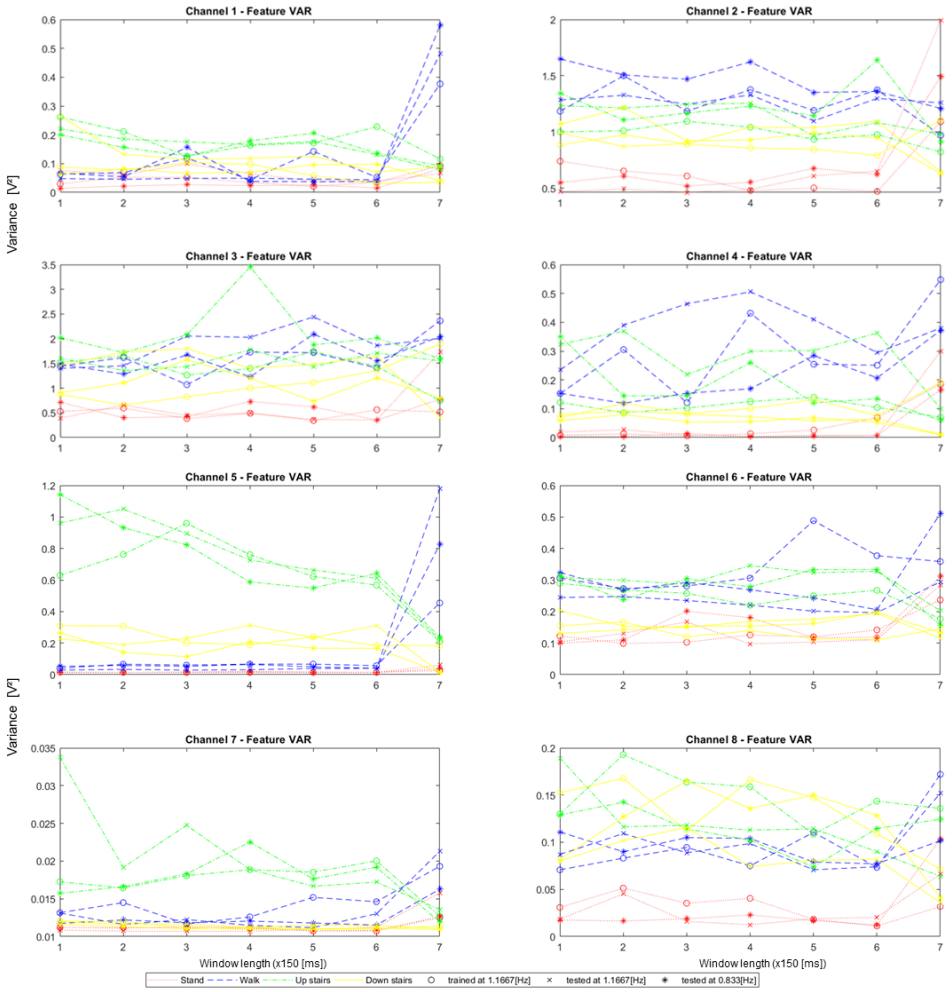


Figure D.2: Variance of the EMG signals computed during the experiment with subject 1

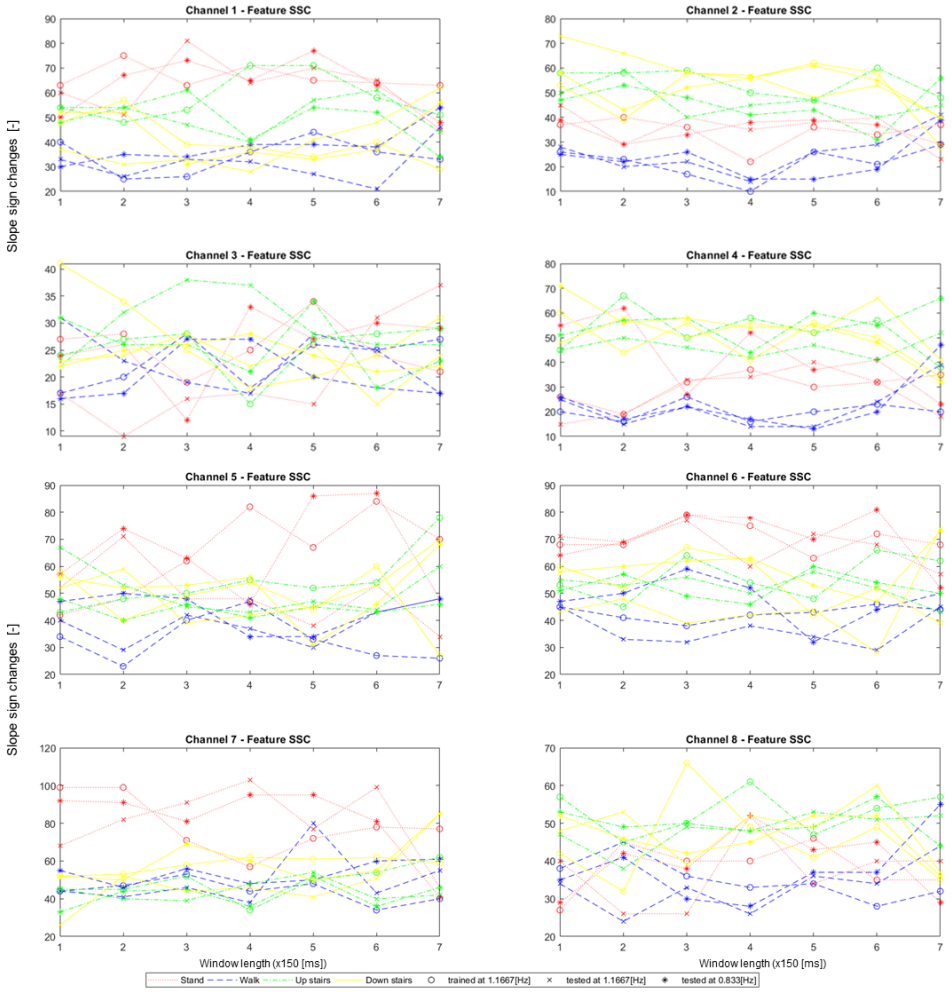


Figure D.3: Slope sign changes of the EMG signals computed during the experiment with subject 1

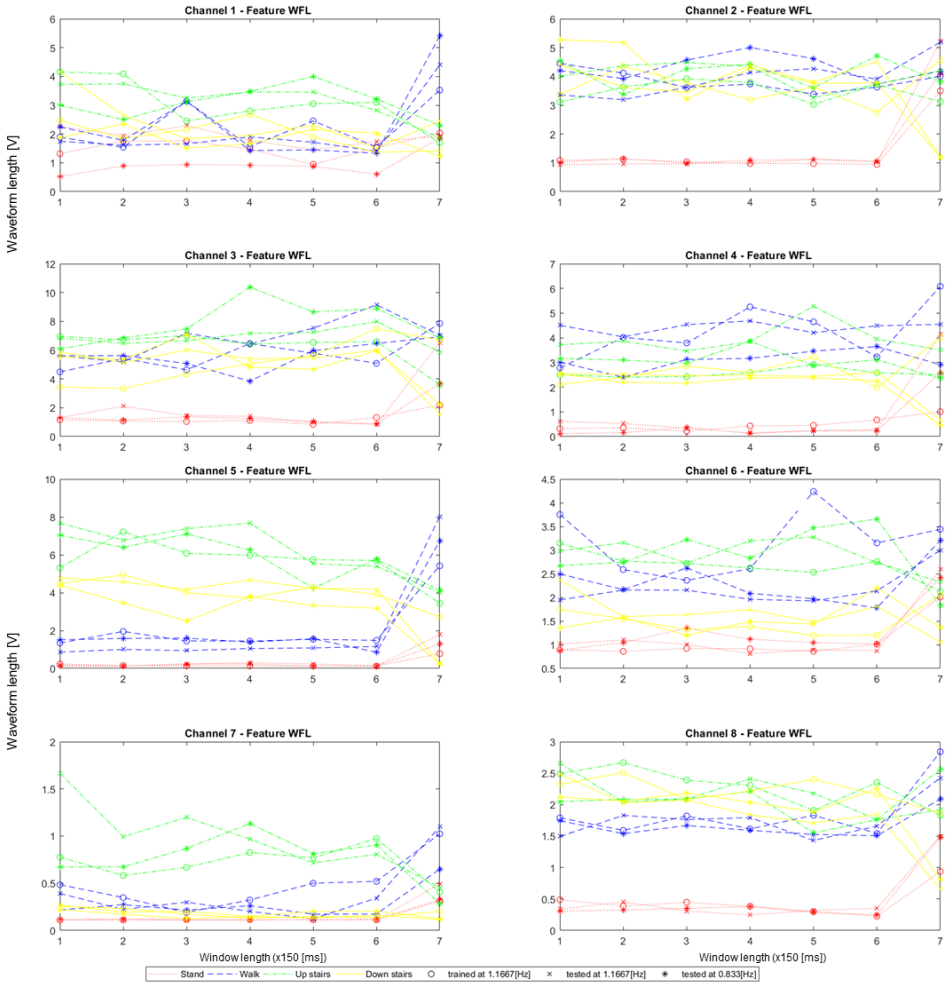


Figure D.4: Waveform length of the EMG signals computed during the experiment with subject 1

Subject 2 - Trained and tested at 1.1667[Hz] - 4 tasks - 7 muscles

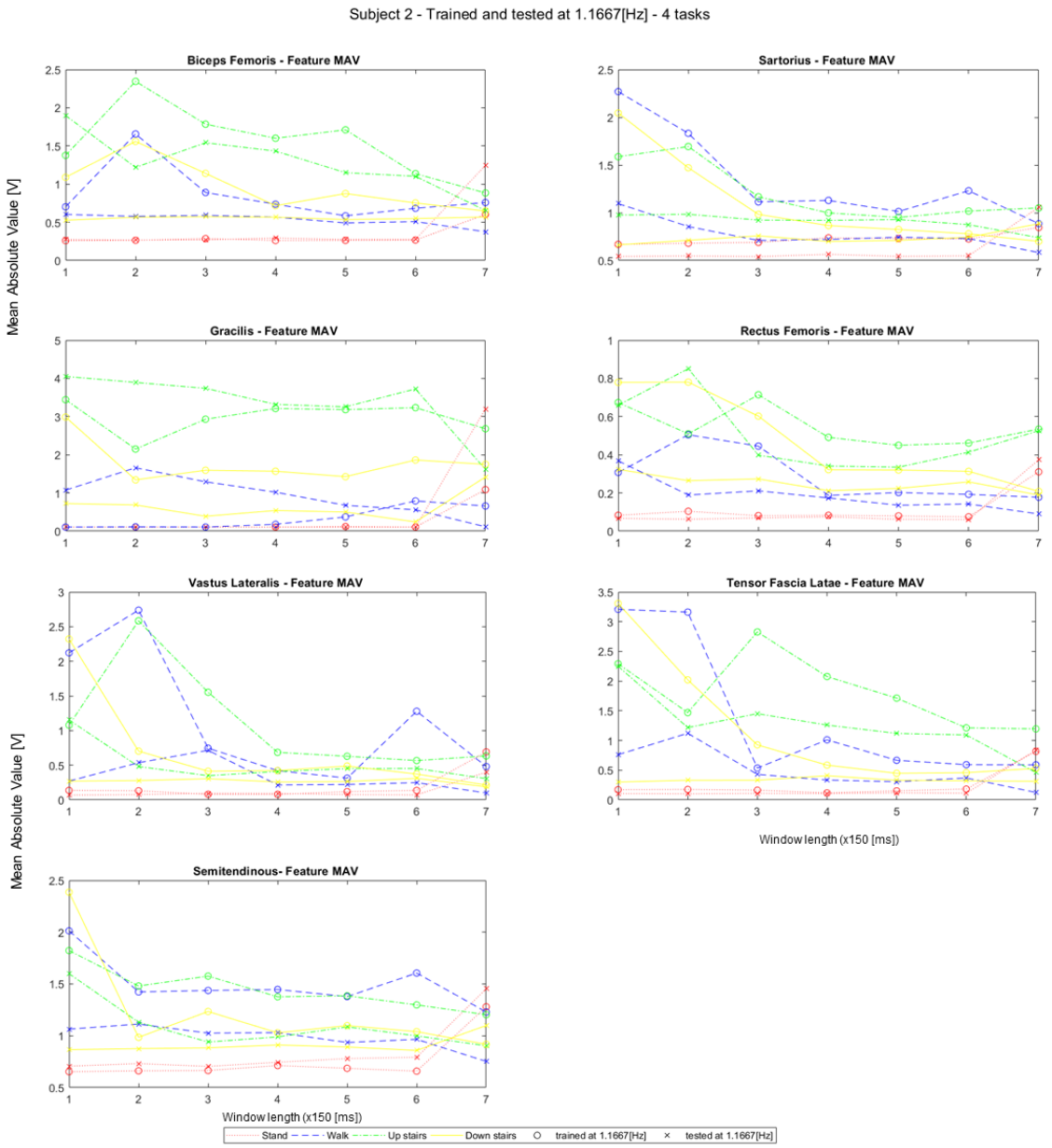


Figure D.5: Mean absolute value of the EMG signals computed during the experiment with subject 2, trained and tested at 1.1667[Hz]

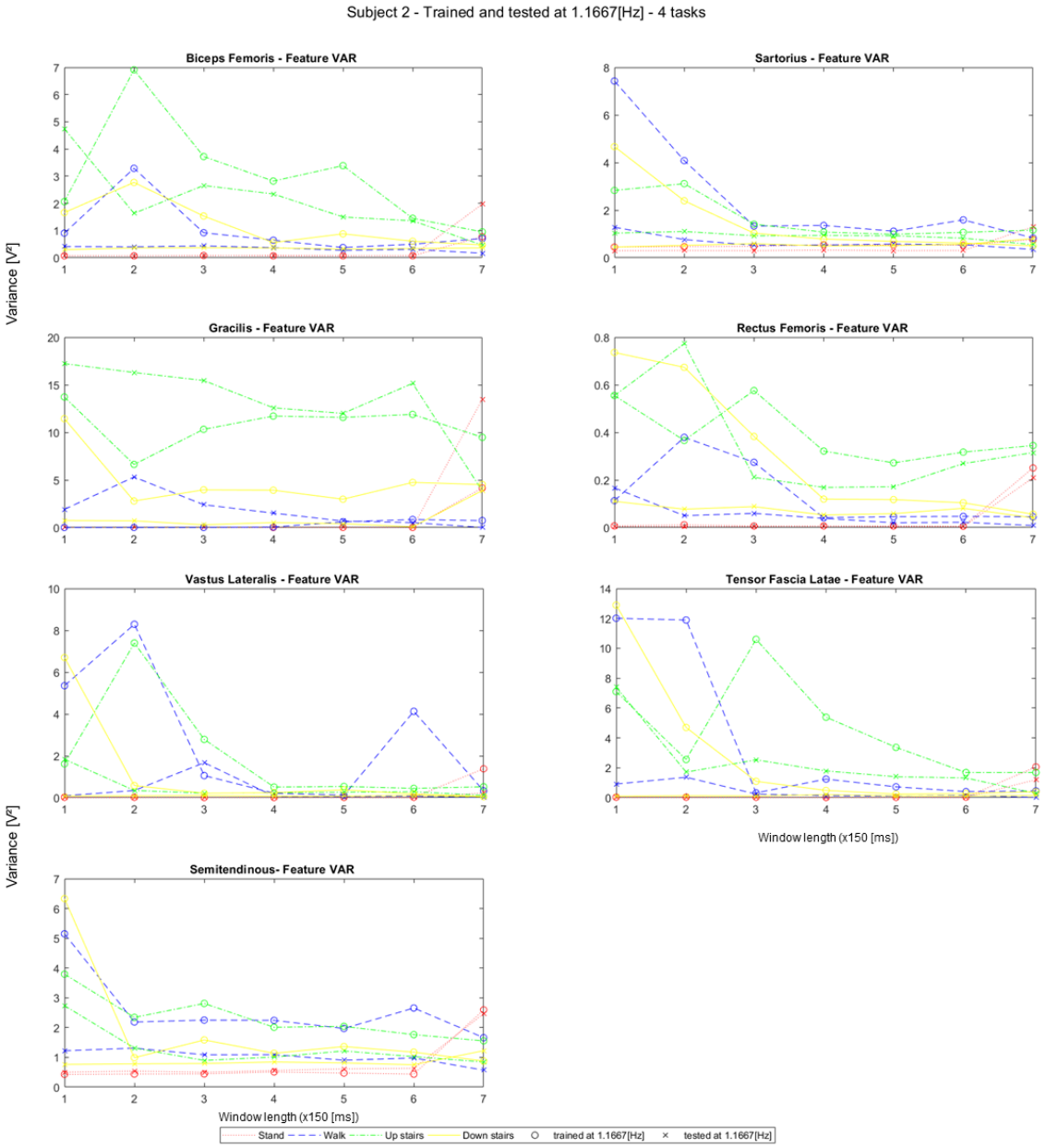


Figure D.6: Variance of the EMG signals computed during the experiment with subject 2, trained and tested at 1.1667[Hz]

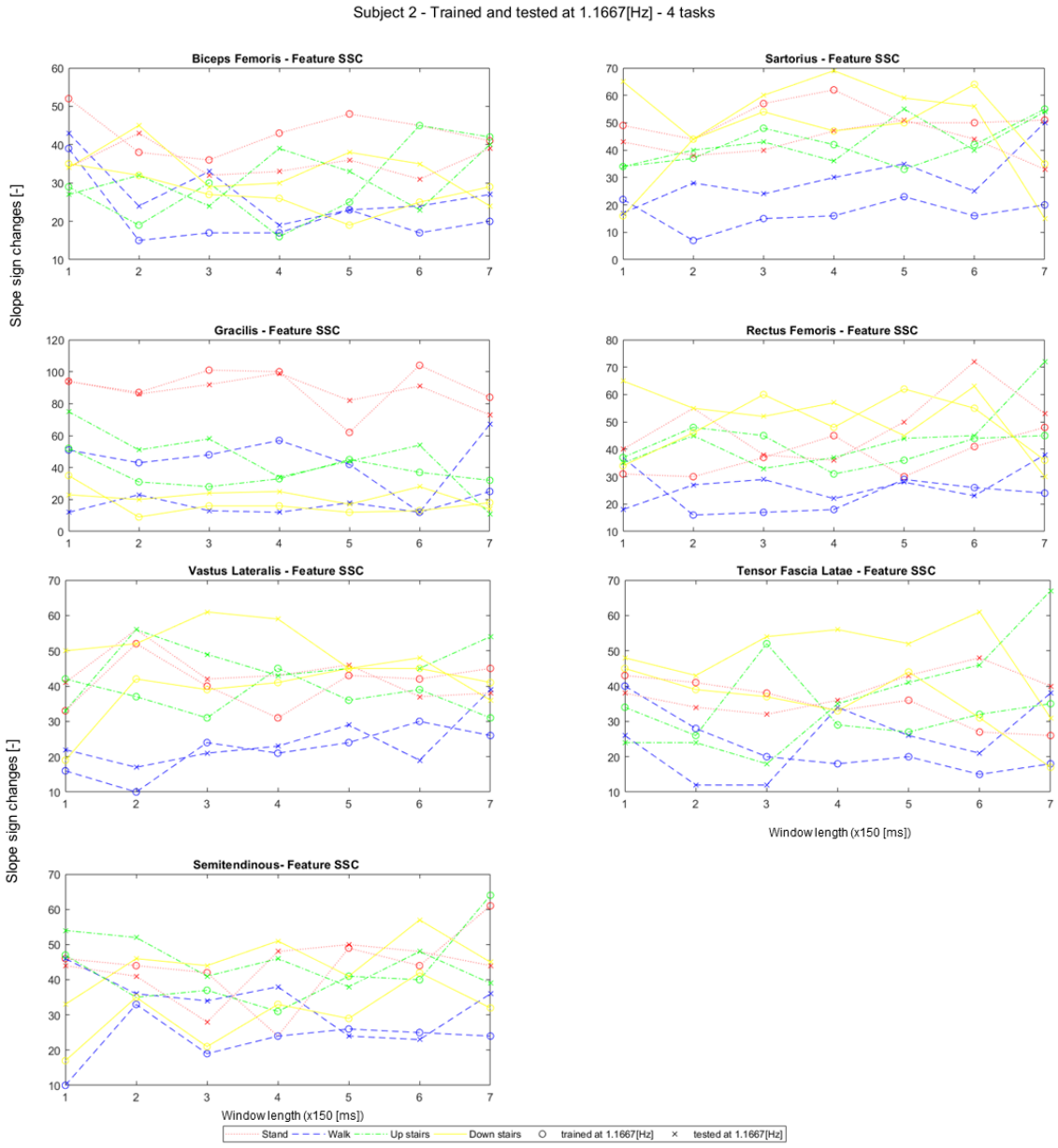


Figure D.7: Slope sign changes of the EMG signals computed during the experiment with subject 2, trained and tested at 1.1667[Hz]

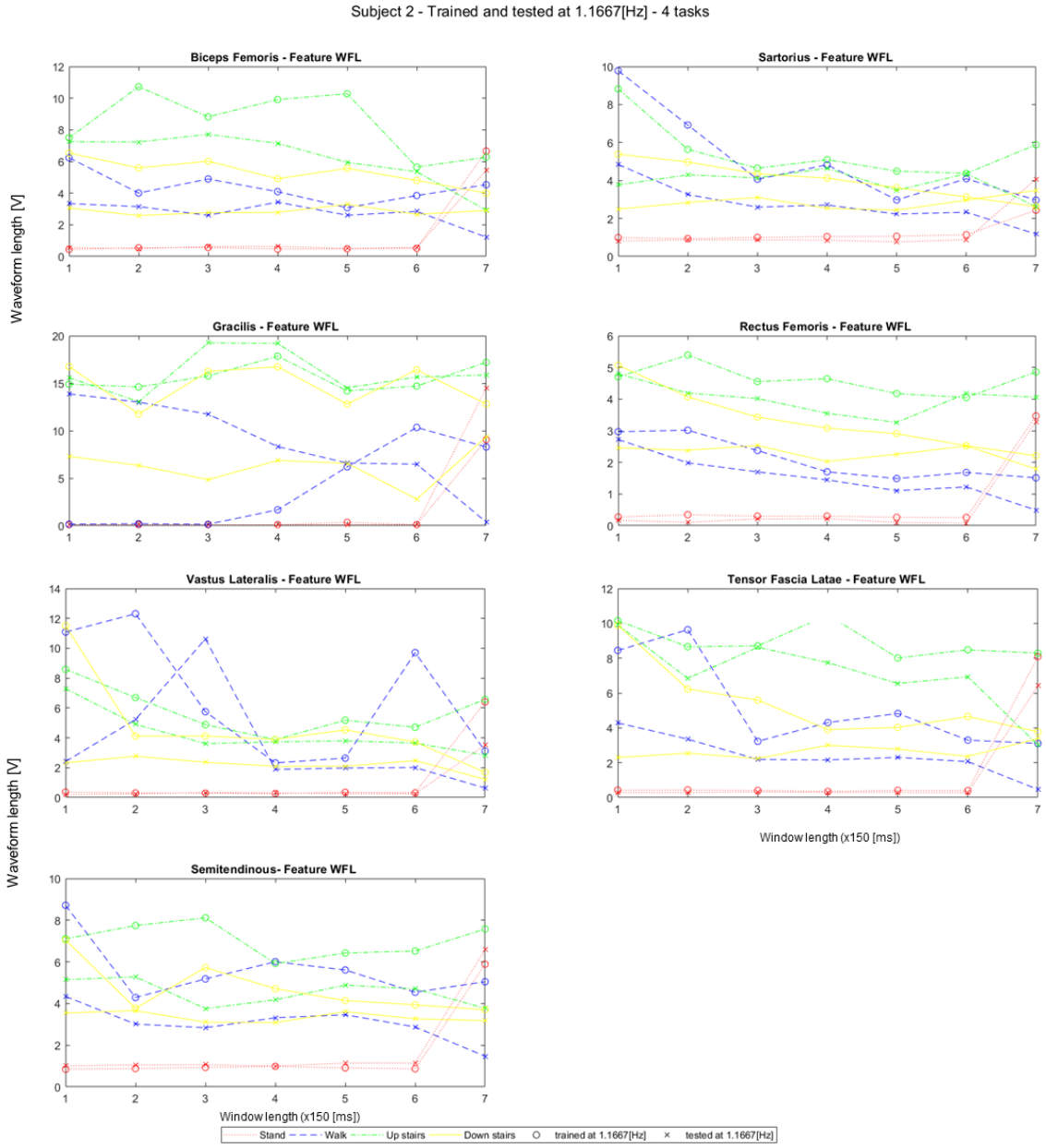


Figure D.8: Waveform length of the EMG signals computed during the experiment with subject 2, trained and tested at 1.1667[Hz]

Subject 2 - Trained and tested at 1.5[Hz] - 4 tasks - 7 muscles

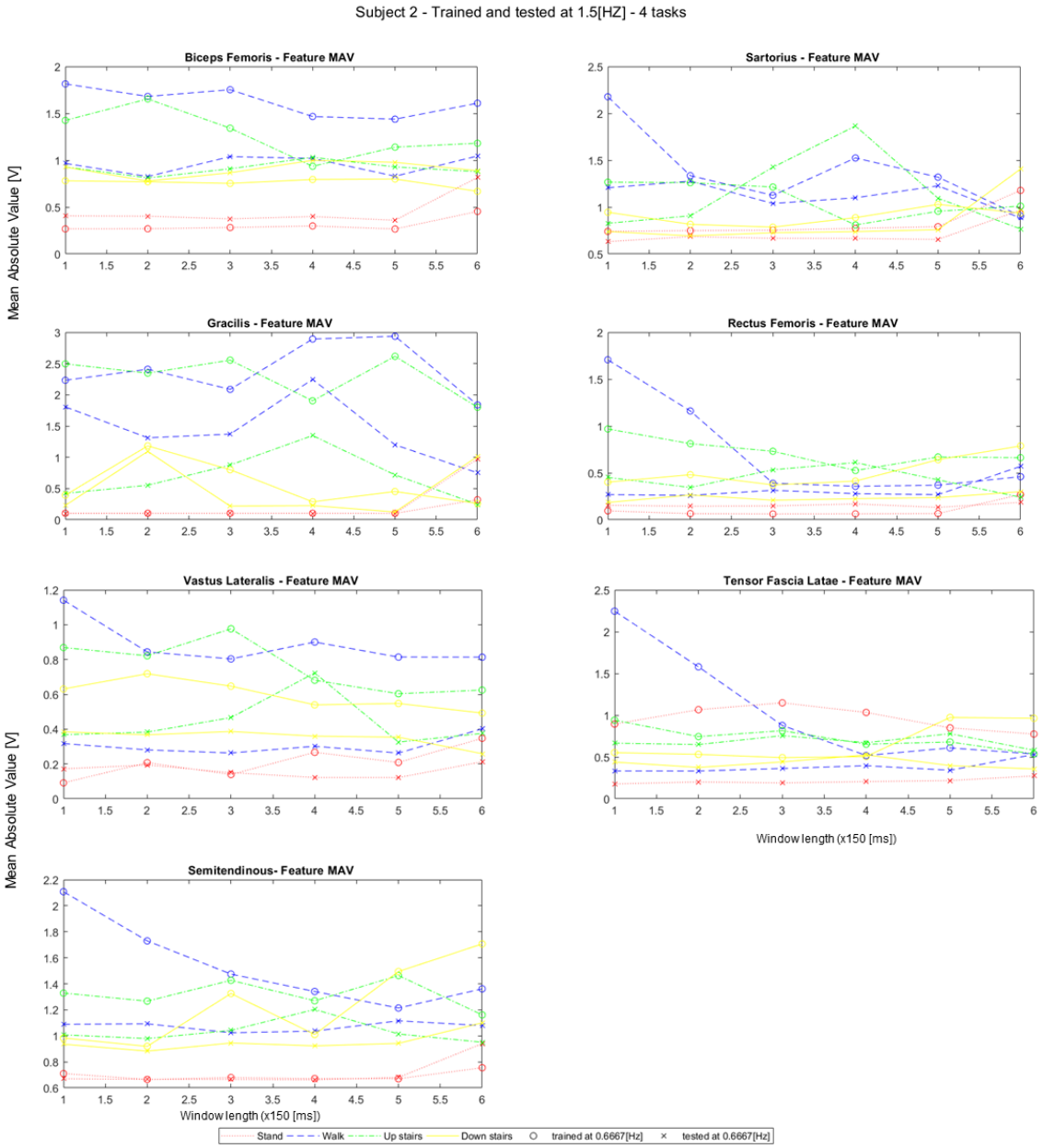


Figure D.9: Mean absolute value of the EMG signals computed during the experiment with subject 2, trained and tested at 1.5[Hz]

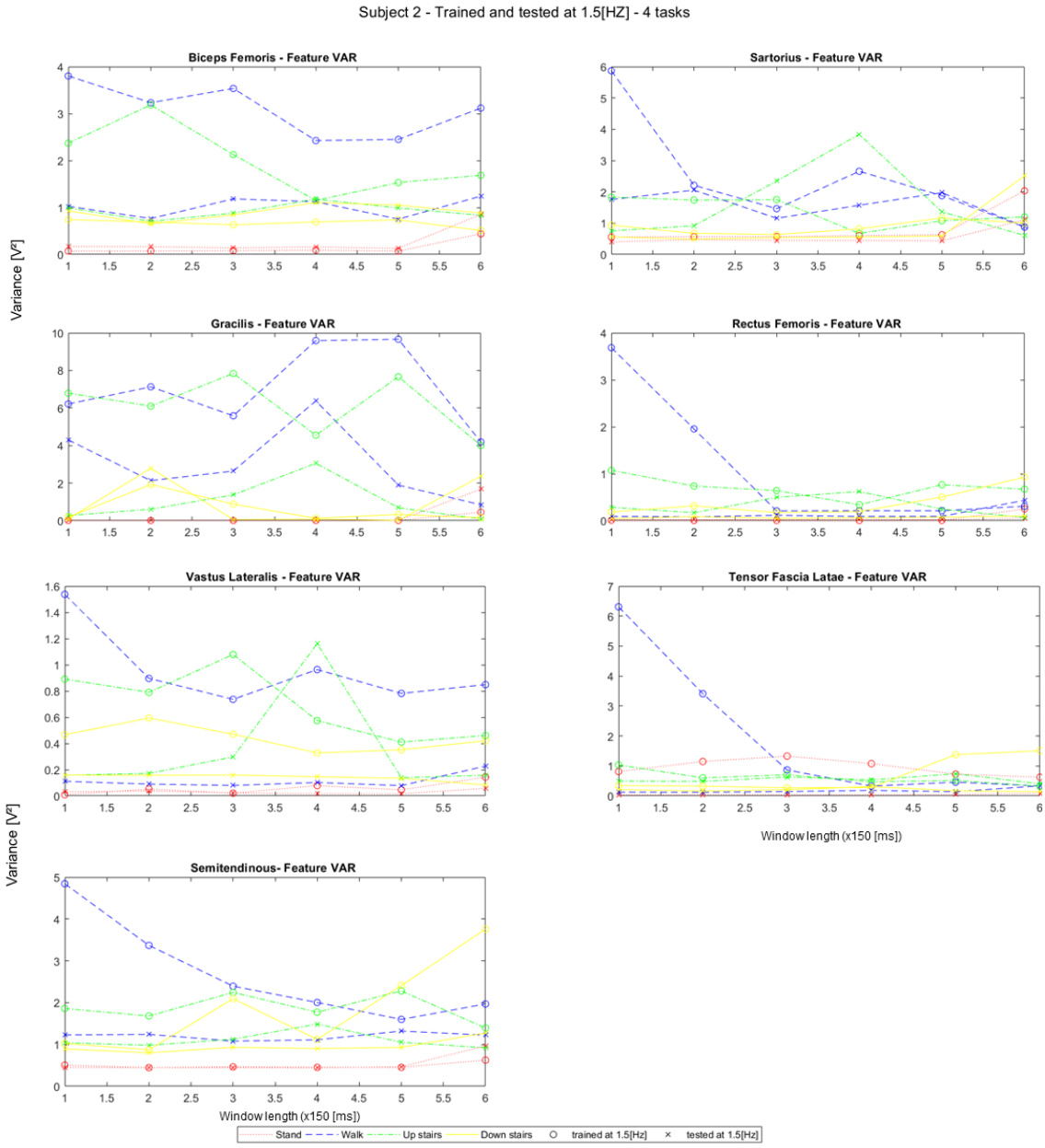


Figure D.10: Variance of the EMG signals computed during the experiment with subject 2, trained and tested at 1.5[Hz]

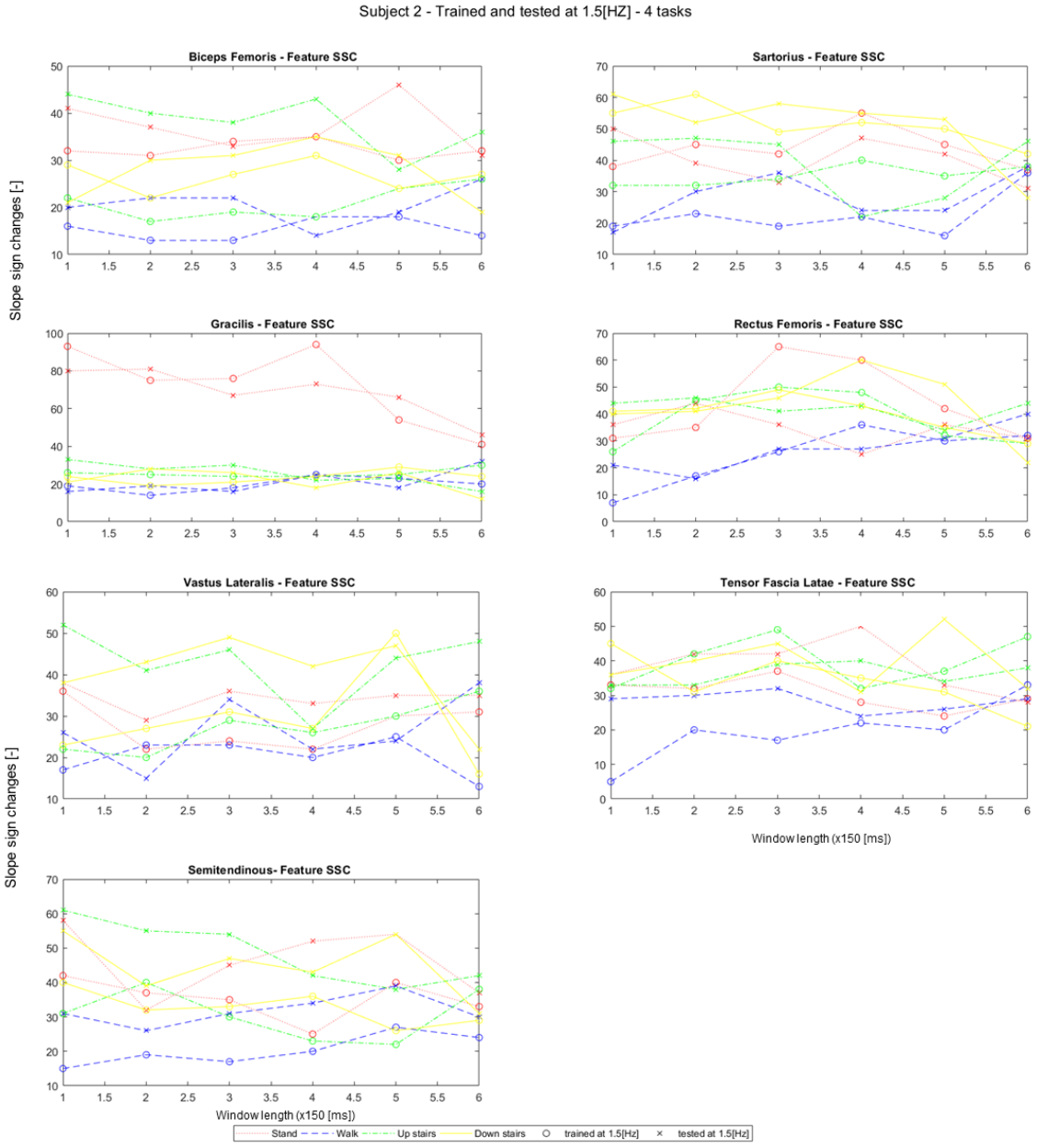


Figure D.11: Slope sign changes of the EMG signals computed during the experiment with subject 2, trained and tested at 1.5[Hz]

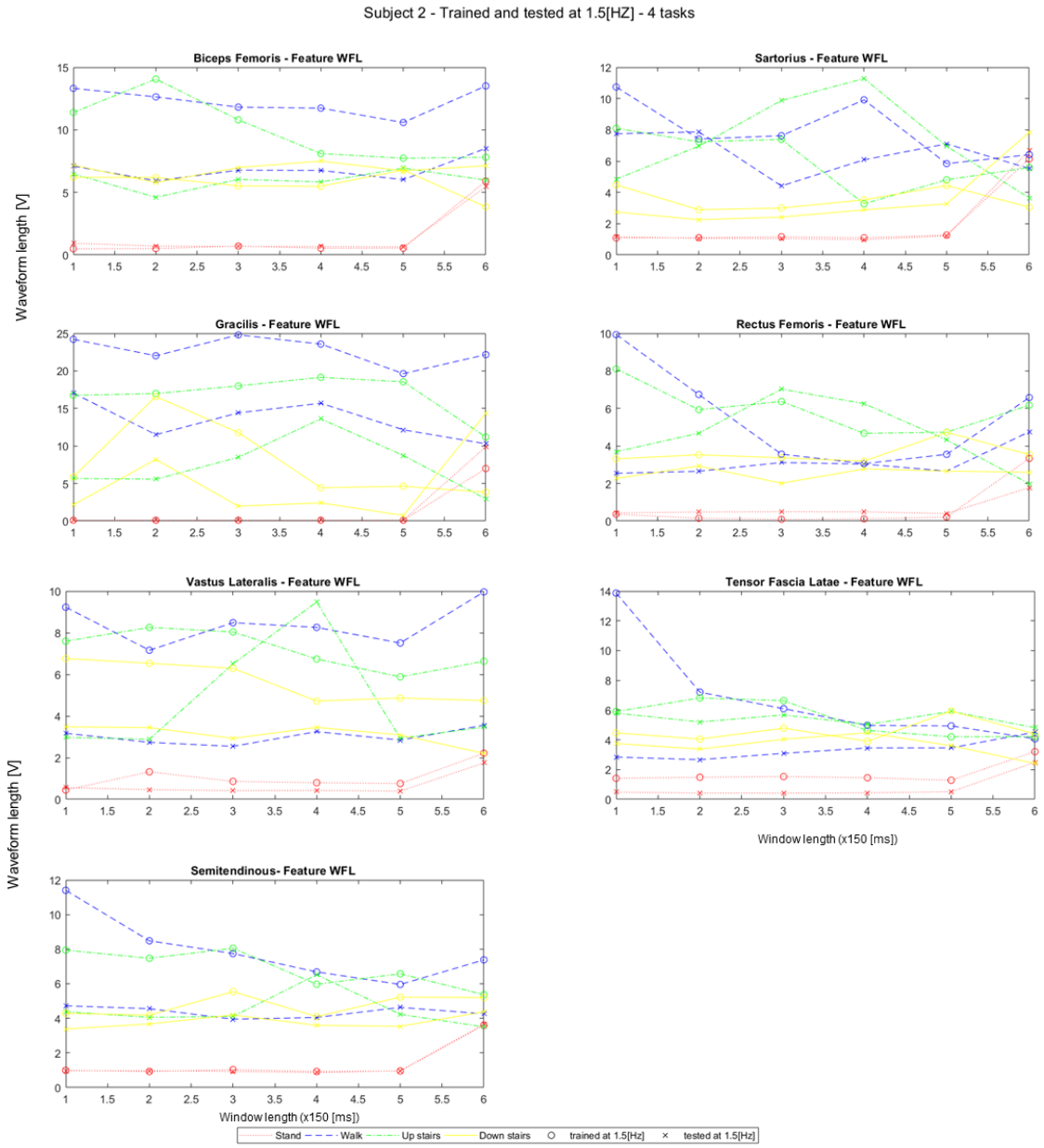


Figure D.12: Waveform length of the EMG signals computed during the experiment with subject 2, trained and tested at 1.5[Hz]

Subject 3 - Trained and tested at 1.1667[Hz] - 6 tasks - 7 muscles

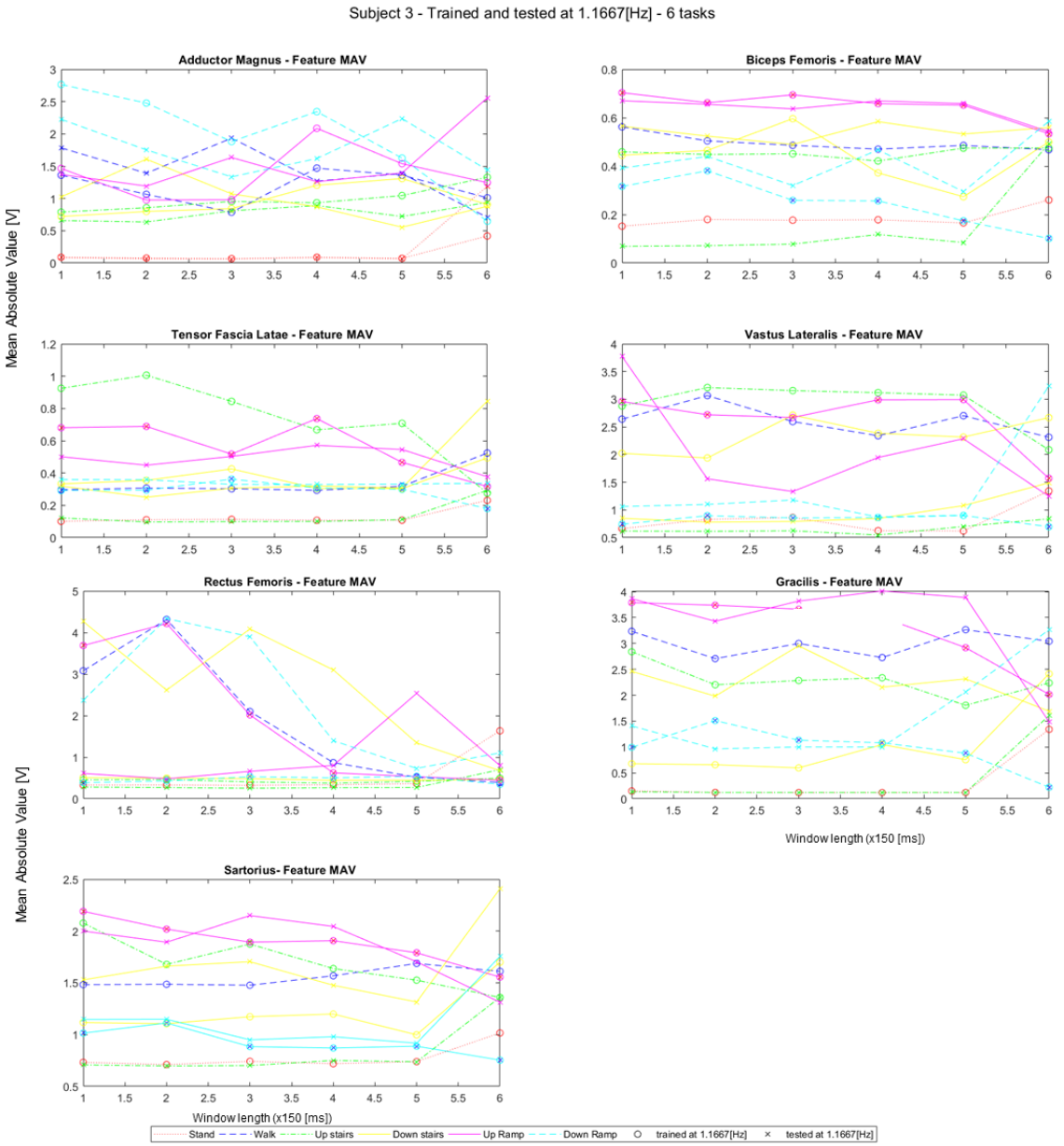


Figure D.13: Mean absolute value of the EMG signals from 7 muscles computed during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz]

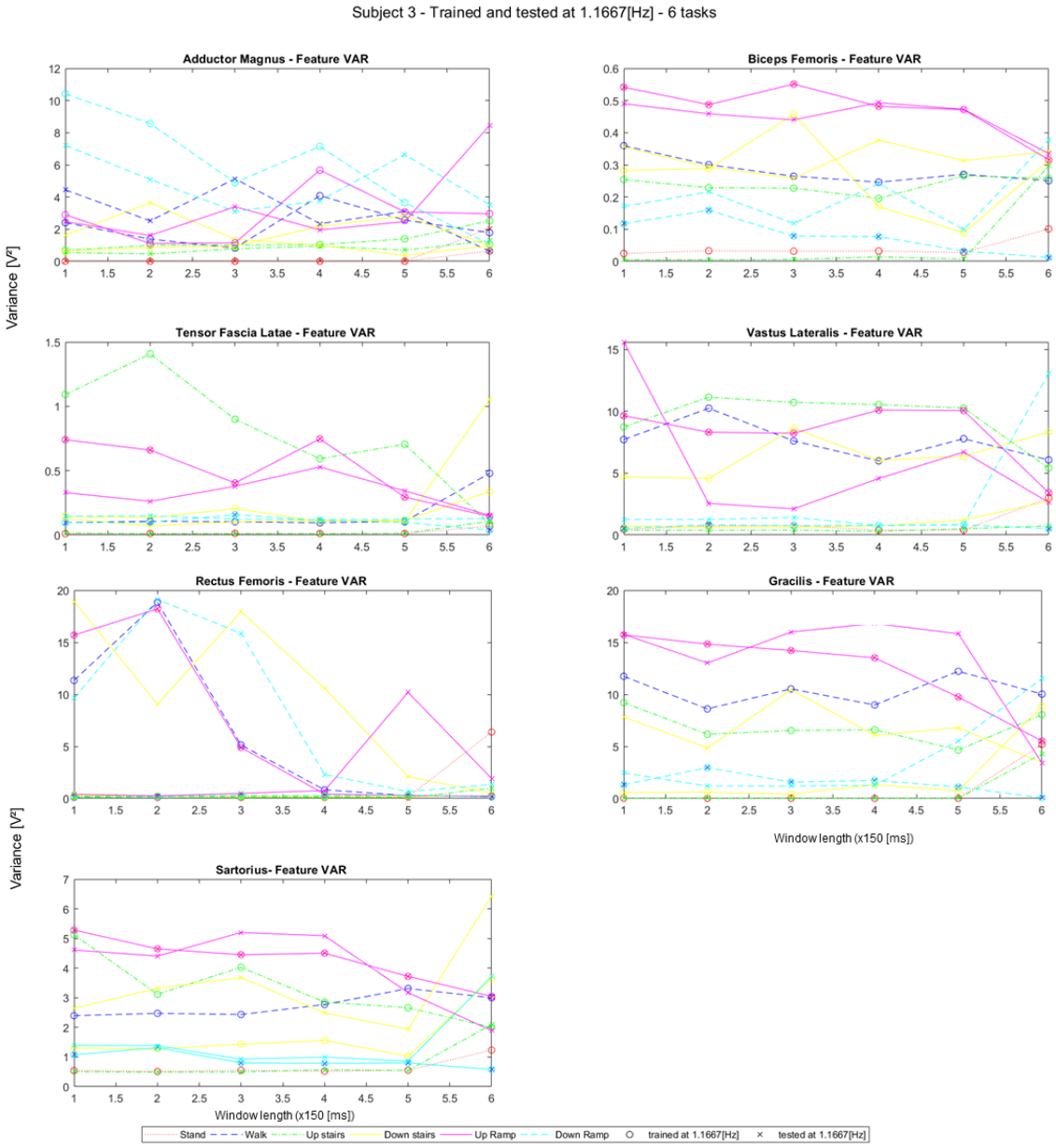


Figure D.14: Variance of the EMG signals from 7 muscles computed during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz]

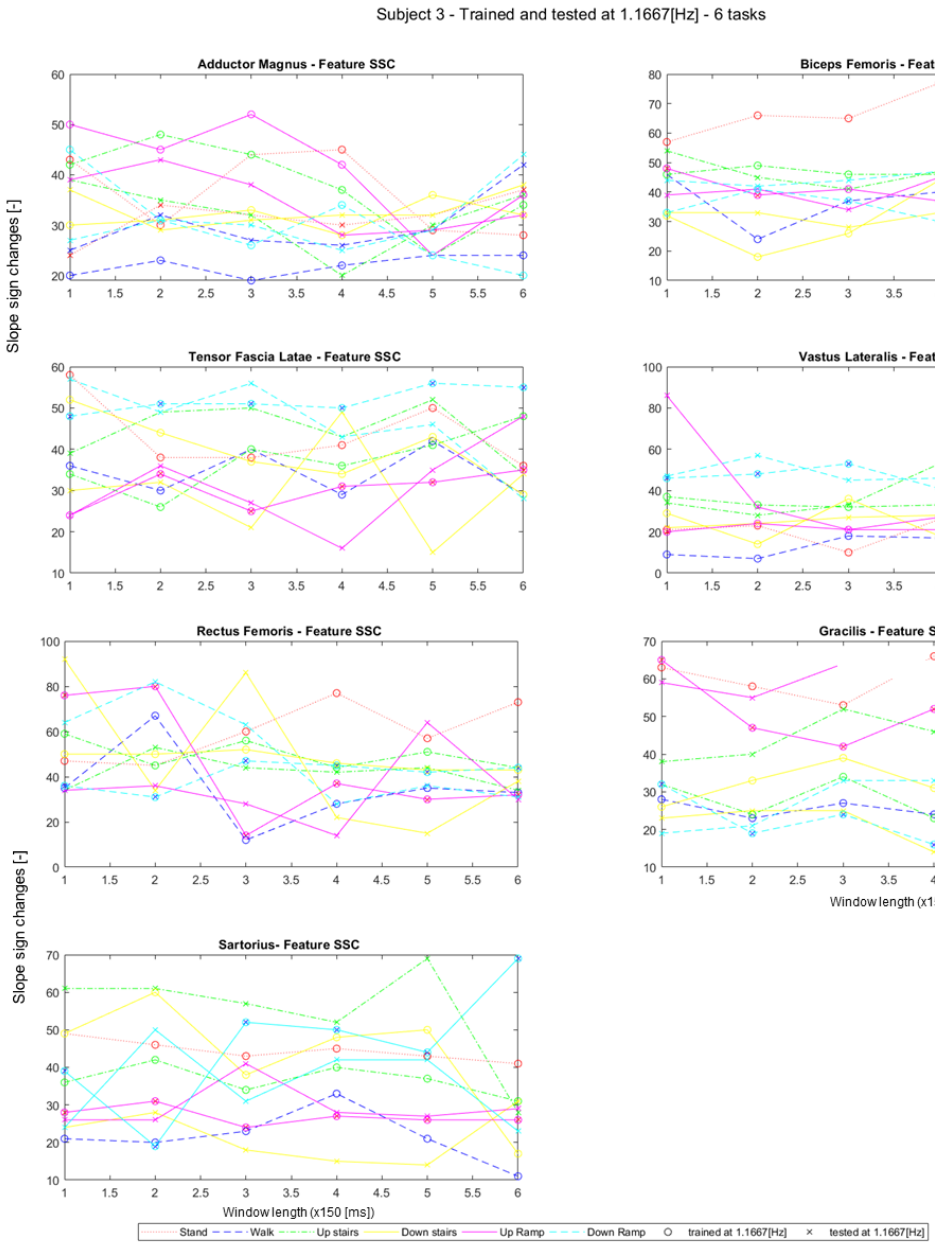


Figure D.15: Slope sign changes of the EMG signals from 7 muscles computed during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz]

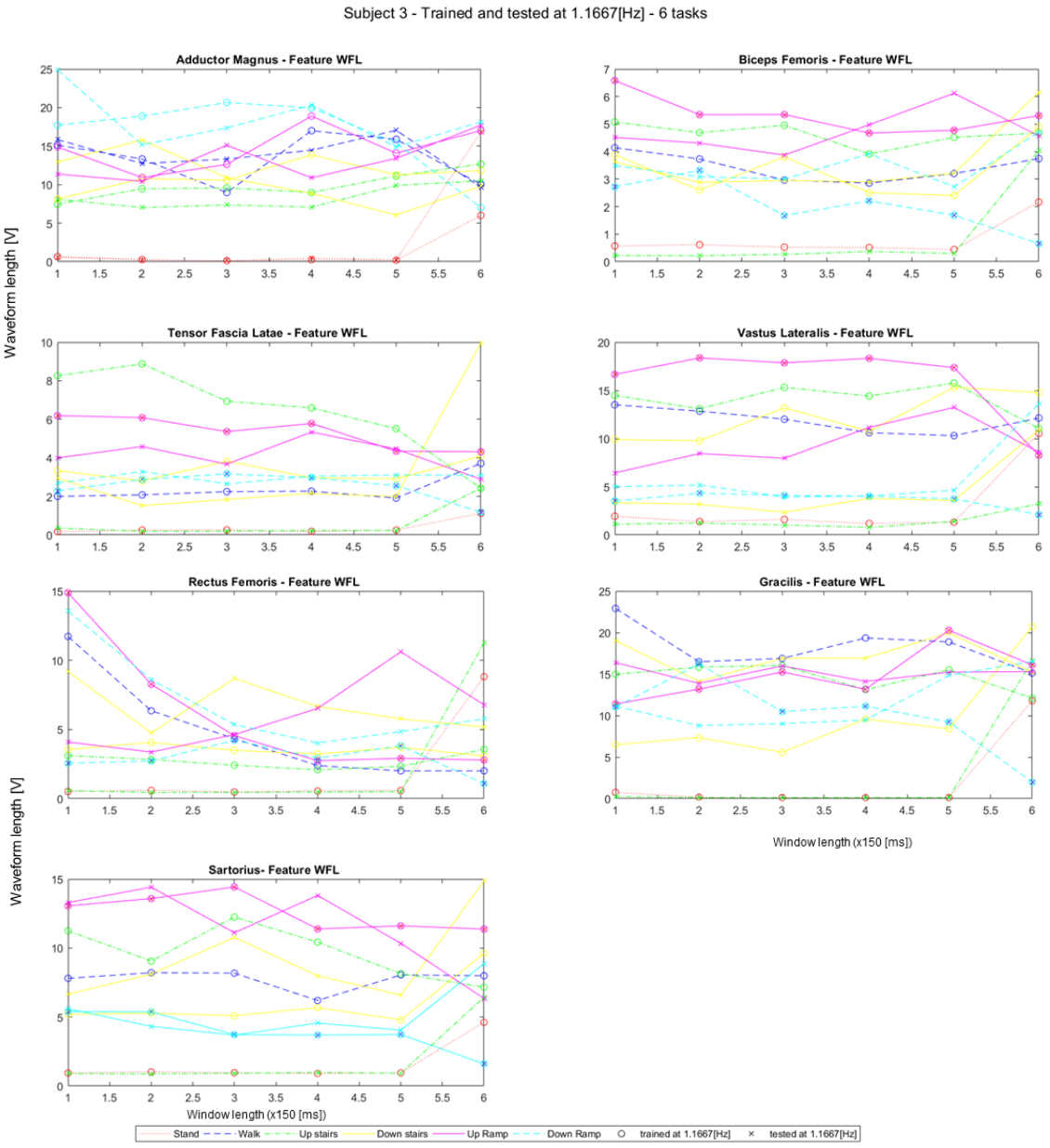


Figure D.16: Waveform length of the EMG signals from 7 muscles computed during the experiment with subject 3 and 6 tasks, trained and tested at 1.1667[Hz]

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 7 muscles

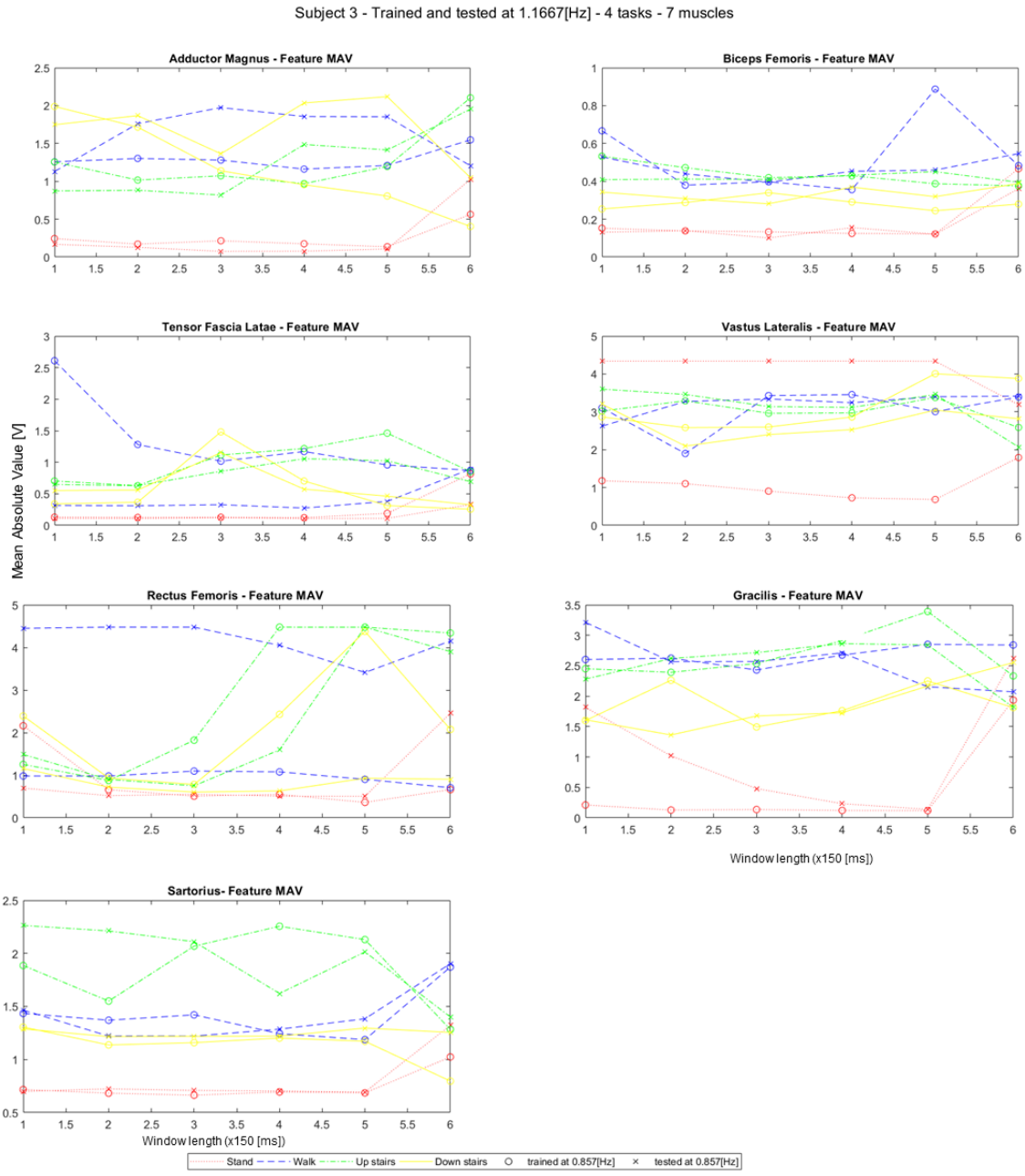


Figure D.17: Mean absolute value of the EMG signals from 7 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

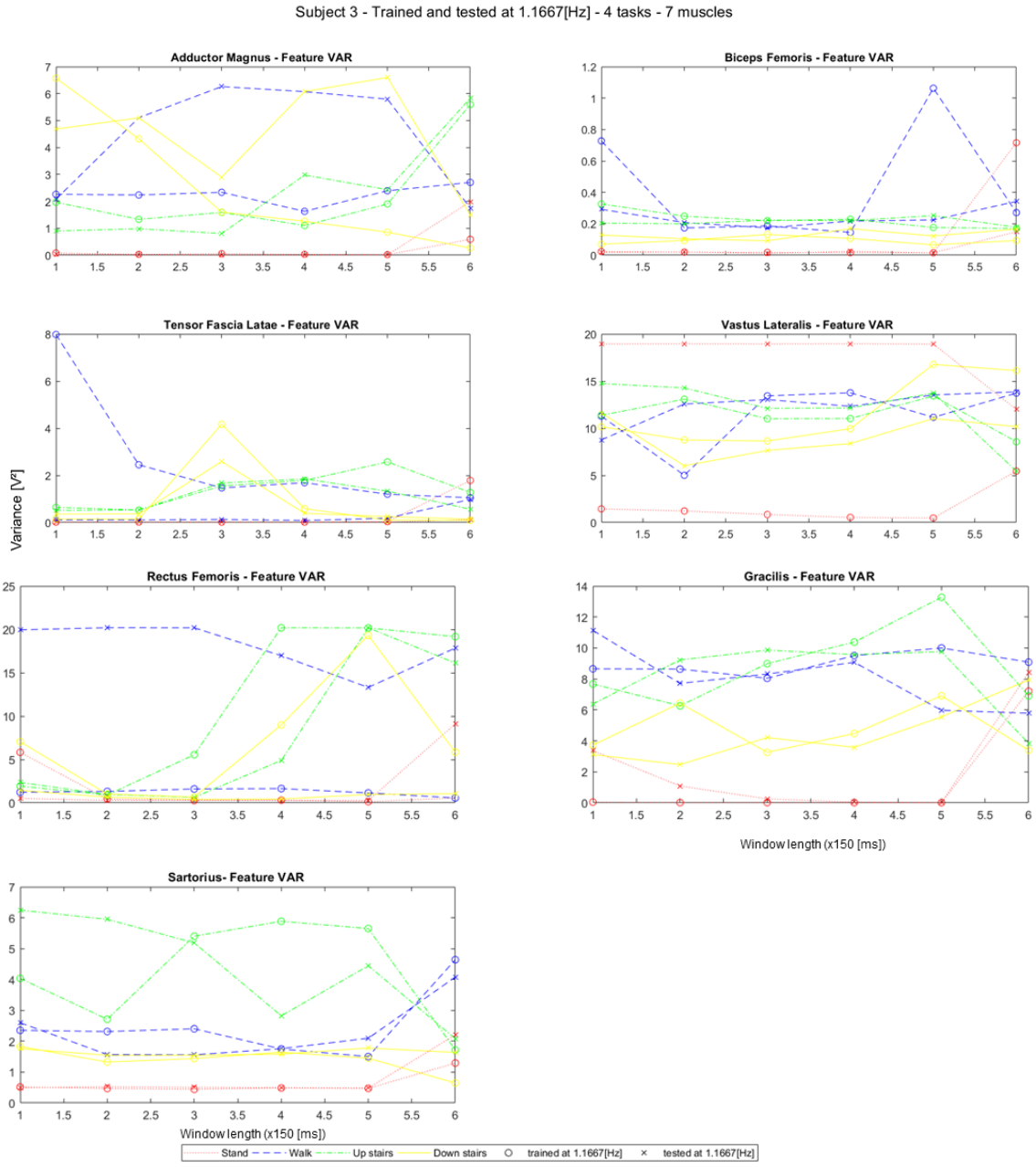


Figure D.18: Variance of the EMG signals from 7 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

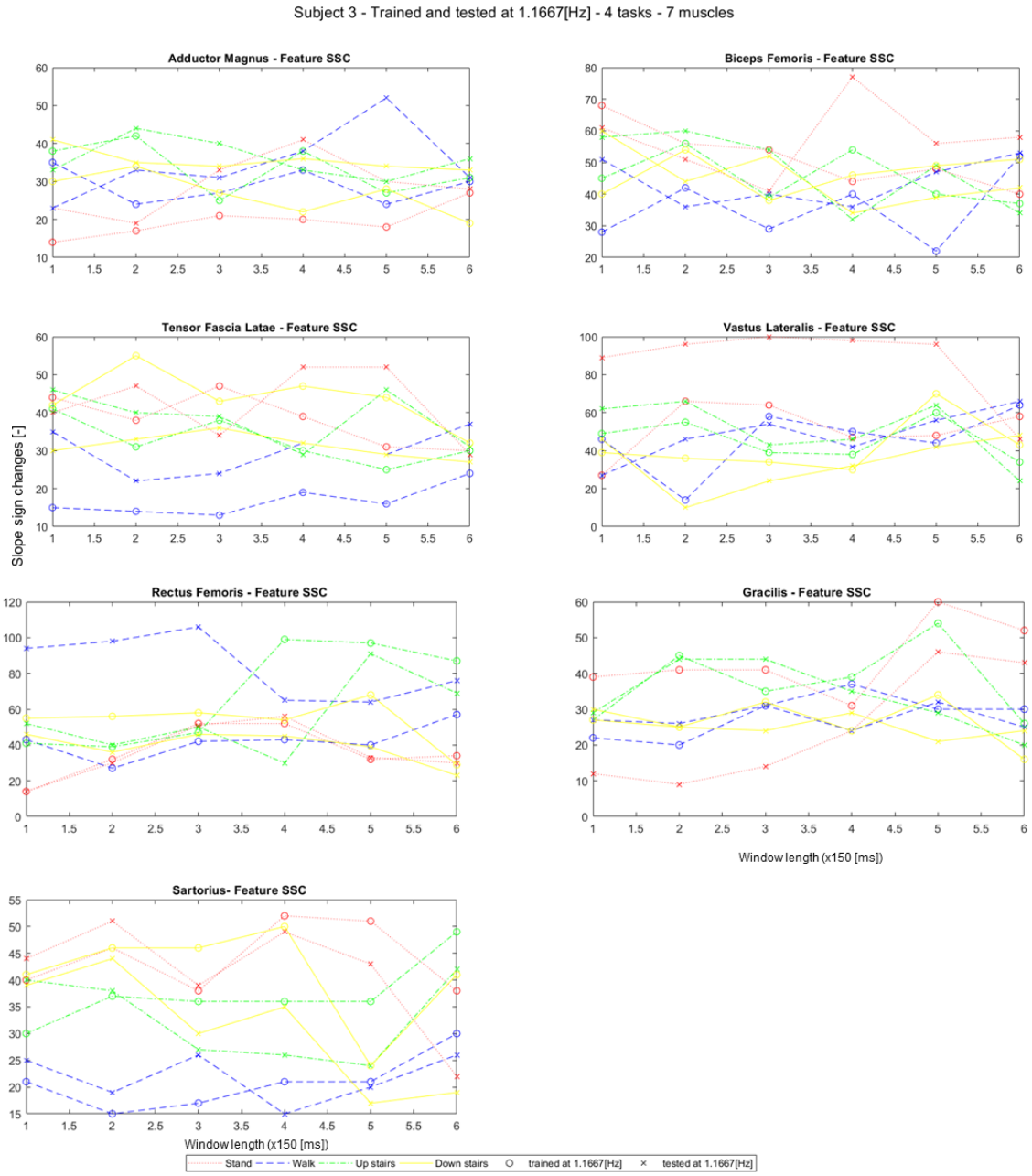


Figure D.19: Slope sign changes of the EMG signals from 7 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

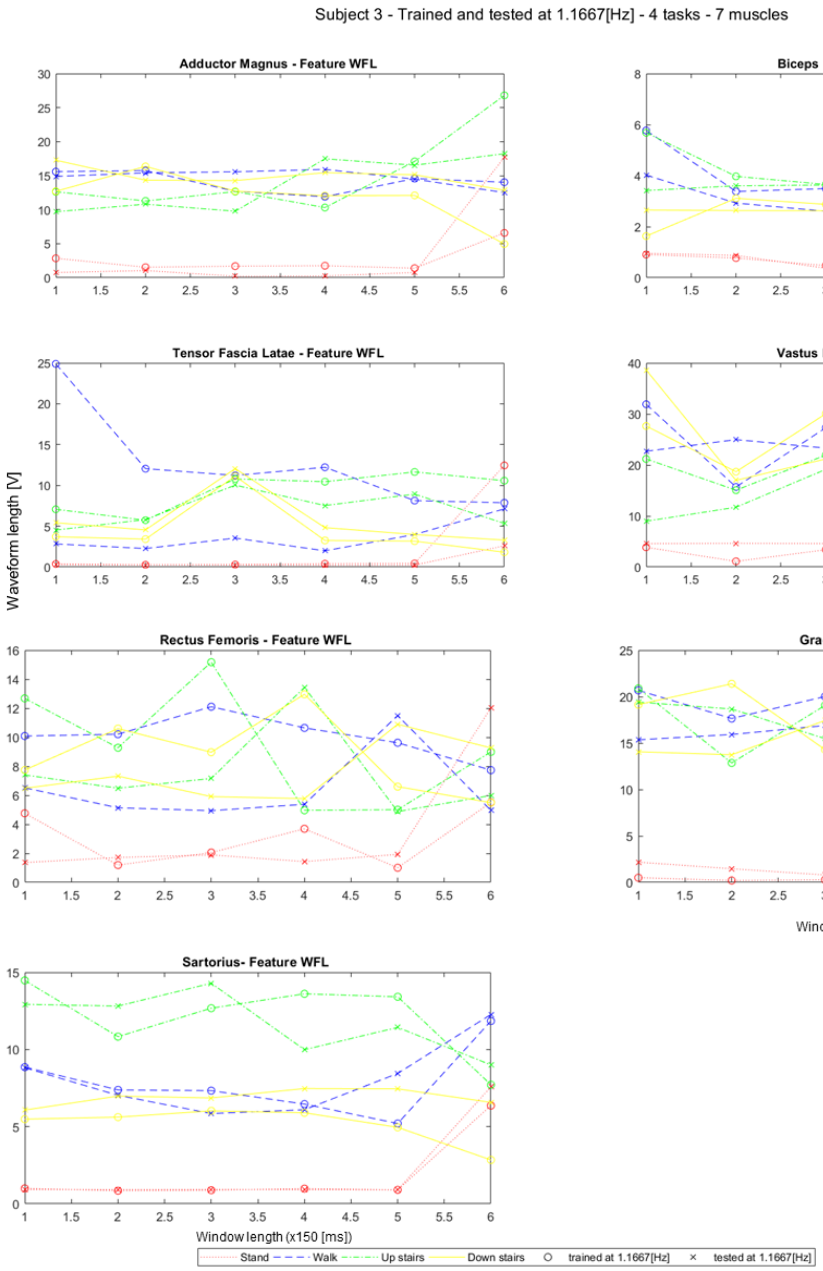


Figure D.20: Waveform length of the EMG signals from 7 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 5 muscles

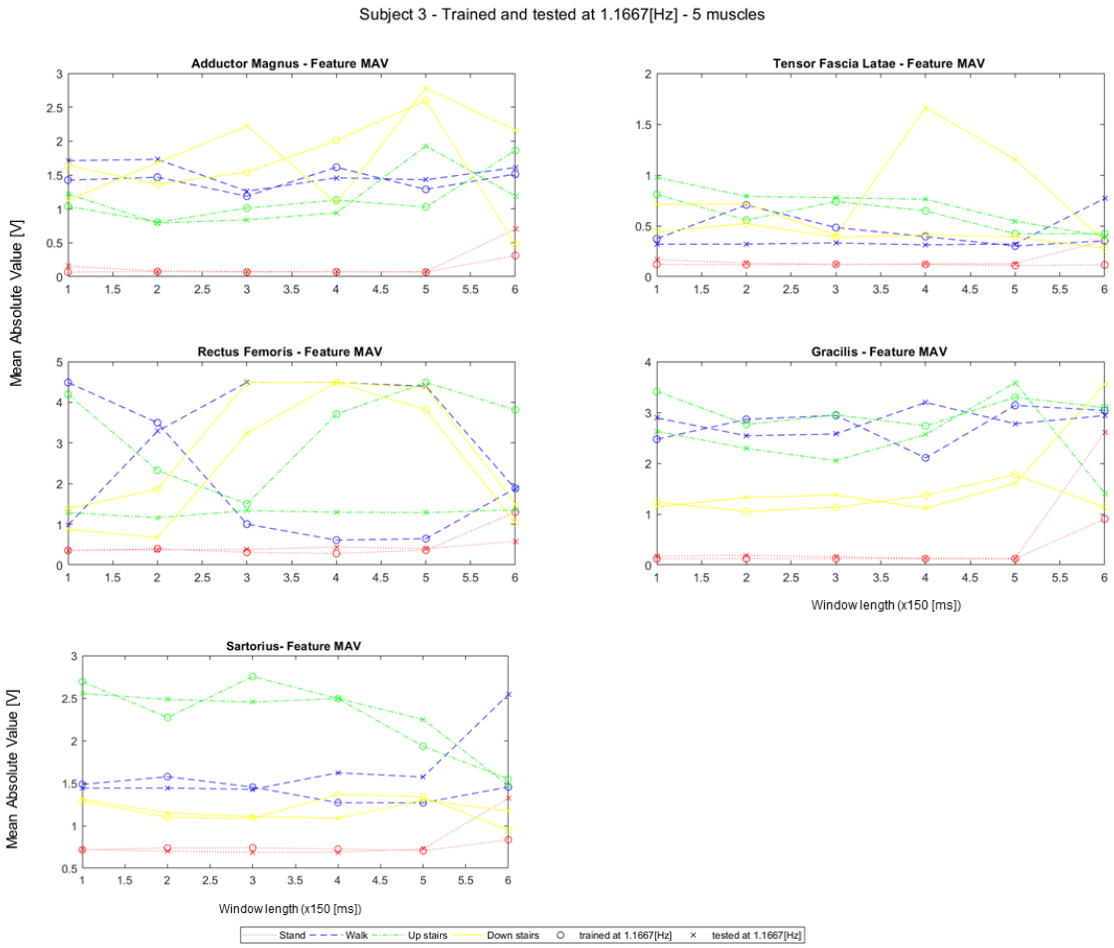


Figure D.21: Mean absolute value of the EMG signals from 5 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

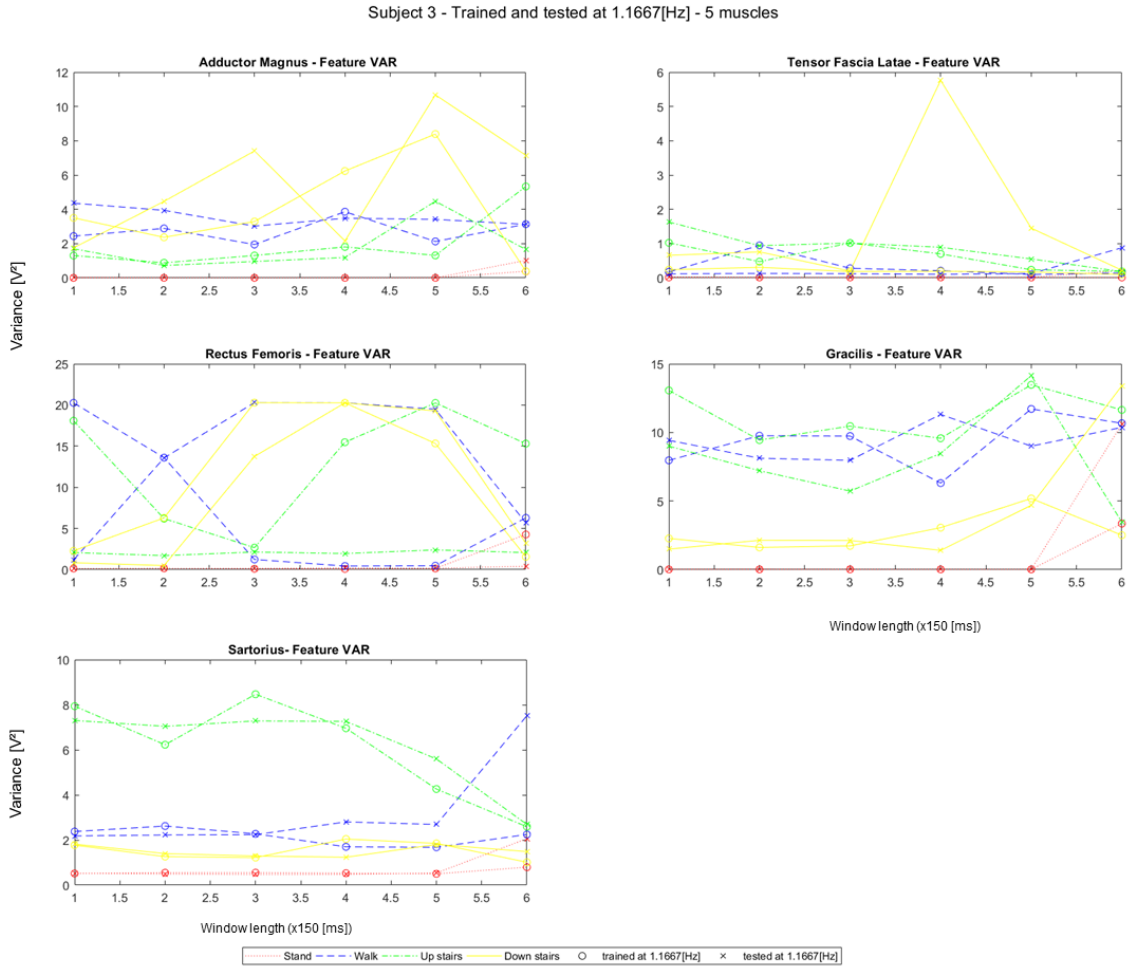


Figure D.22: Variance of the EMG signals from 5 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

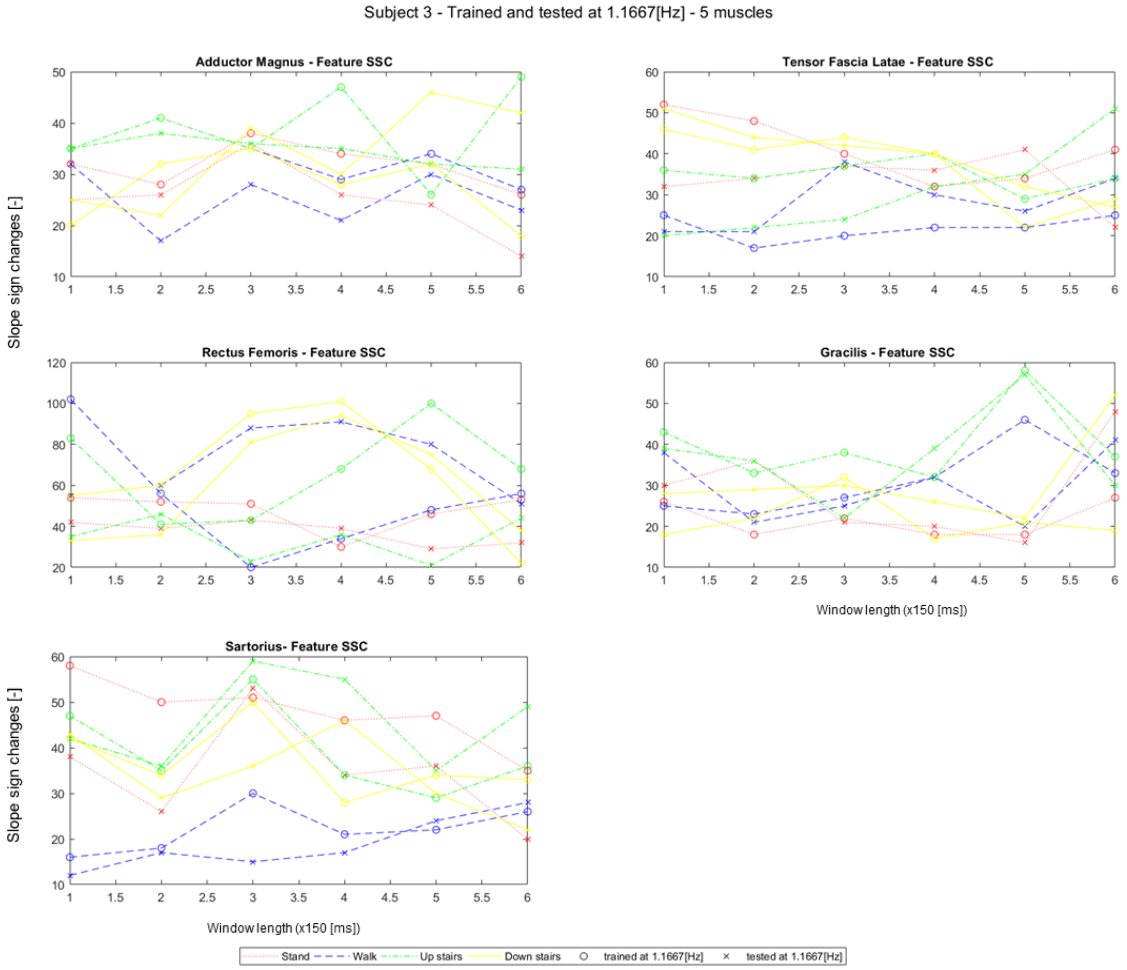


Figure D.23: Slope sign changes of the EMG signals from 5 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

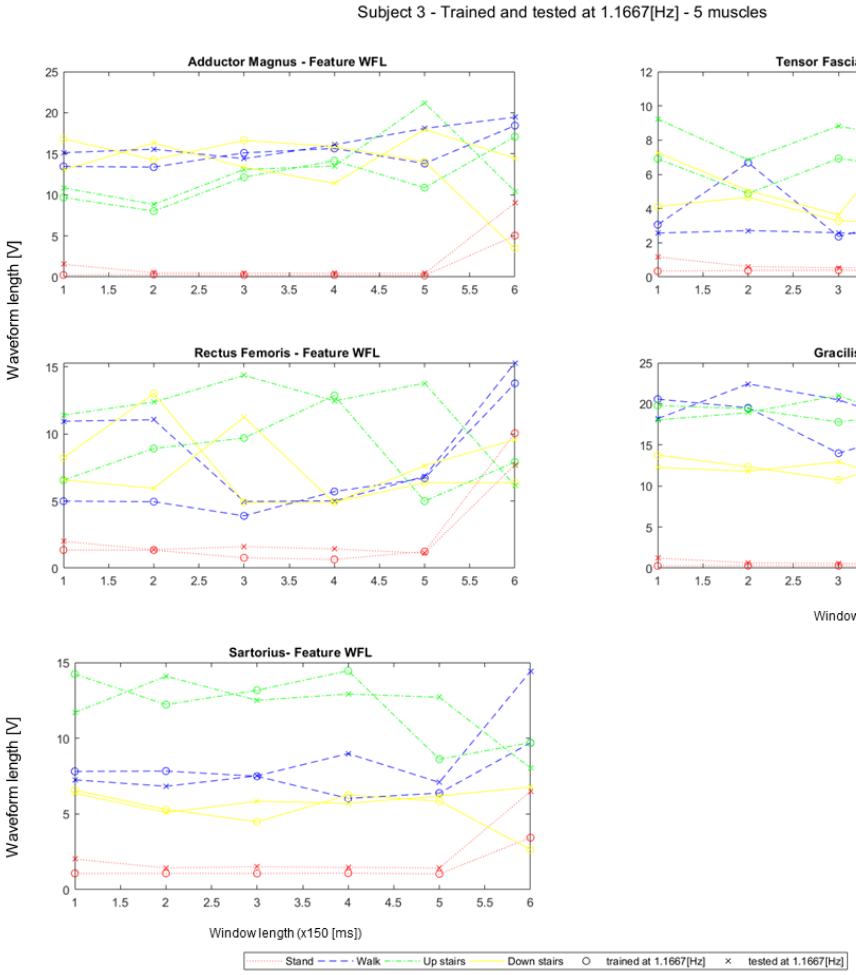


Figure D.24: Waveform length of the EMG signals from 5 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

Subject 3 - Trained and tested at 1.1667[Hz] - 4 tasks - 3 muscles

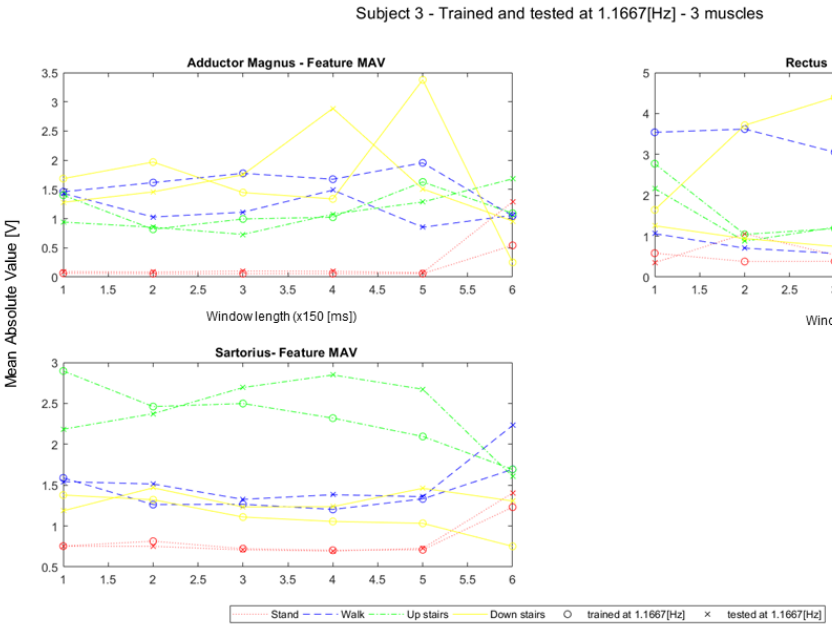


Figure D.25: Mean absolute value of the EMG signals from 3 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

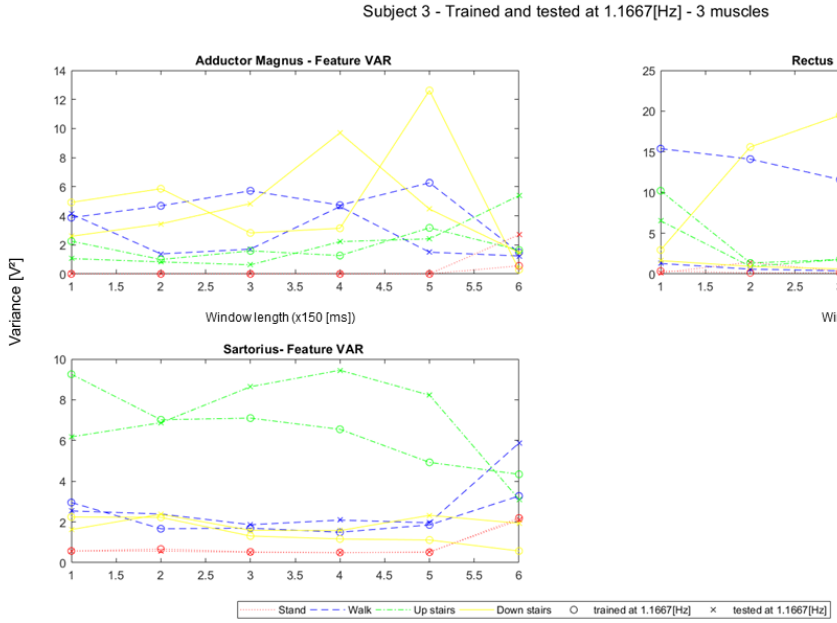


Figure D.26: Variance of the EMG signals from 3 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

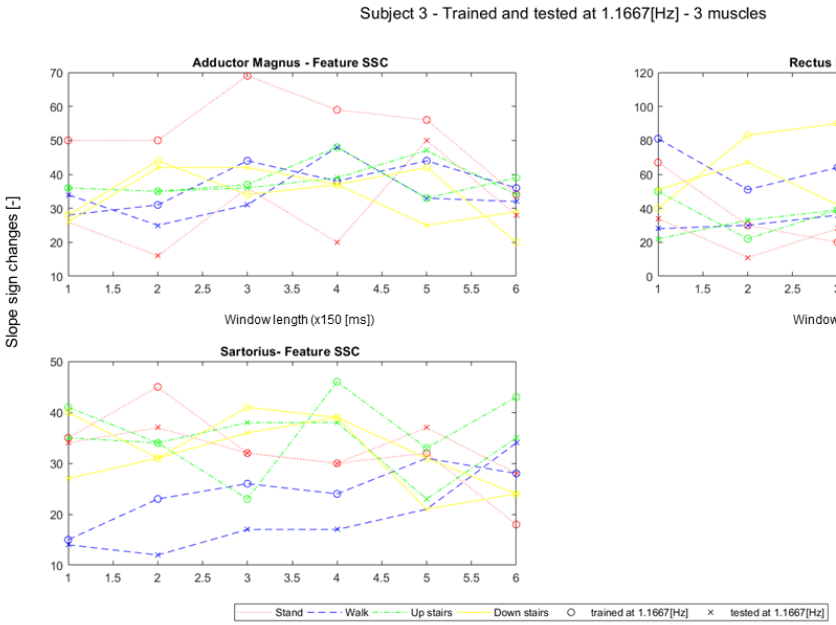


Figure D.27: Slope sign changes of the EMG signals from 3 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

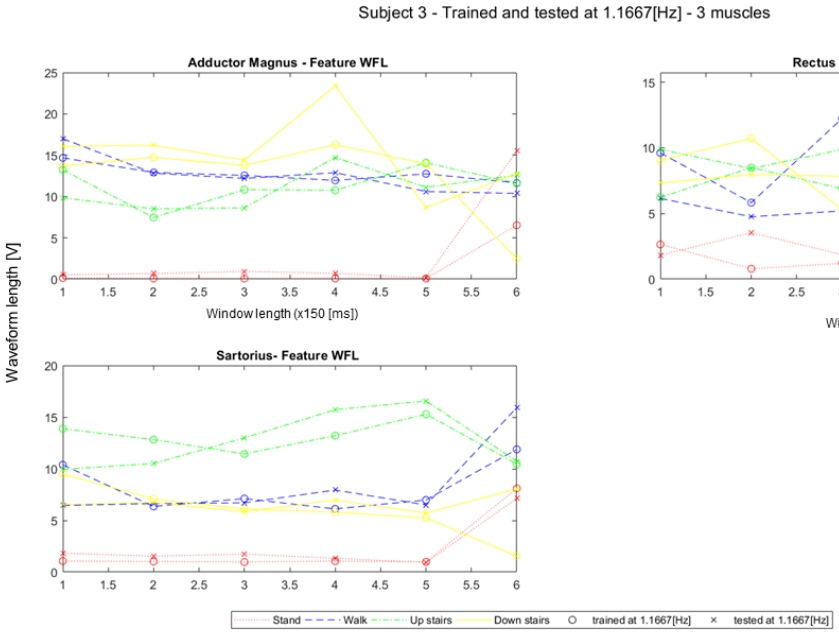


Figure D.28: Waveform length of the EMG signals from 3 muscles computed during the experiment with subject 3 and 4 tasks, trained and tested at 1.1667[Hz]

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**