MASTER THESIS

MYOELECTRIC CONTROL OF BIONIC HANDS VIA MUSCULOSKELETAL MODELLING, ADMITTANCE CONTROL AND FORCE FEEDBACK

Mikel Mateo

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF BIOMECHANICAL ENGINEERING

EXAMINATION COMMITTEE
dr.ir. M. Sartori
dr.ir. A.Q.L. Keemink
G.V. Durandau
dr.ir. U. Yavuz

DOCUMENT NUMBER
BW - 709

UNIVERSITY OF TWENTE.

16/12/2019

Abstract

The loss of an upper limb can have a profound impact in the quality of life, since the individual's ability to perform activities of daily live autonomously, the capability to work and socialize are suddenly limited. Myoelectric prosthesis aim to restore the missing functions by providing a man-machine interface based on electromyography which enables the amputee to control the artificial limb. However, current myoelectric control approaches do not provide truly bio-mimetic control that allows for both intuitive control and reliable grasp dynamics. This is mainly because current machine learning approaches provide a control interface that is non-intuitive and extraordinarily different from human natural control. Furthermore, the lack of afferent feedback pathways forces the amputee to rely mainly on visual feedback to control grasp action, where the implementation of state-of-the-art haptic feedback techniques does not show conclusive results on the help of grasp force control. In this work, a subject-specific EMG-driven musculoskeletal model coupled to an admittance controller is used to drive a bionic hand. The admittance model virtually mimics the grasp dynamics of a real hand interacting with the environment. This way, the prosthetic hand can readjust its control commands during interaction by accounting for external forces. Theoretical stability boundaries of the control system are analysed for stable interaction and experimental tests are carried out comparing the proposed control framework to non-admittance based EMG-driven musculoskeletal modelling during grasping tasks. Experimental outcomes show positive results when compared to non-admittance based control. This control framework sets the bases to enable safer prosthesis-environment interaction without the need of constant visual feedback.

List of Acronyms

ADL	Activities of Daily Living
BMI	Brain-Machine Interface
CC	Co-Contraction
DOF	Degrees of Freedom
ECoG	Electrocorticography
EEG	Electroencephalography
EMG	Electromyography
FRF	Frequency Response Function
HOC	Hand Open-Close
MCP	Metacarpophalangeal
MMI	Man-Machine Interface
MS	Musculoskeletal
MTUs	Musculotendon Units
MVC	Maximal Voluntary Vontraction
MYO	Myoelectric
NMS	Neuromusculoskeletal
PR	Pattern Recognition
sEMG	Surface Electromyography
TMR	Targeted Muscle Reinervation
WPS	Wrist Pronation-Supination

Contents

1 Background **1**
- 1.1 Introduction . . . 1
- 1.2 Demographics and Prevalence . . . 1
- 1.3 Upper limb prostheses . . . 2
- 1.4 Myoelectric Control . . . 4
- 1.5 Challenges in Current Myoelectric Prosthesis . . . 7
- 1.6 Developments in Prosthetic Control . . . 8
- 1.7 Scope of Thesis: Towards bio-inspired grasp force control . . . 10

Bibliography **11**

2 Paper **15**

Appendices **16**

A EMG-driven musculoskeletal model description, scaling and calibration **35**
- A.1 Model description . . . 35
- A.2 Model Scaling . . . 35
- A.3 EMG calibration . . . 36
- A.4 Model calibration . . . 37

B System communication framework **45**
- B.1 Framework overview . . . 45
- B.2 Michelangelo hand interface . . . 46
- B.3 Processing in Matlab . . . 47

C Prosthesis Control In Matlab **51**
- C.1 Matlab GUI for prosthesis control: overview . . . 51
- C.2 Description and steps for prosthesis control . . . 51
- C.3 Friction parameters . . . 54
- C.4 Admittance parameters . . . 54

D Admittance Model Equations **57**
- D.1 Friction model details . . . 57
- D.2 Equations of Motion . . . 59

E Michelangelo hand's force sensor calibration **61**
- E.1 Overview . . . 61
- E.2 Calibration experiment . . . 61

F Prosthesis Dynamics Identification details **65**
- F.1 Identification experiment . . . 65
- F.2 Computing the frequency response function . . . 66
- F.3 Estimating the parameters of the FRF . . . 67

G Stability Analysis **69**
- G.1 A note for coherence . . . 70

Chapter 1

Background

1.1 Introduction

Human hands are an essential tool to interact with the environment, explore, create, communicate and socialise. They conform a highly versatile device capable of performing a large number of tasks using simple to fine and complex movements. They are an extremely complex biomechanical system formed by twenty-seven bones and actuated by more than thirty muscles leading to more than 19 Degrees of Freedom (DOF). Furthermore, thousands of sensors continuously measure physical variables, such as force and joint position, which are used as feedback to redefine our movements[1].

However, traumatic accidents, dysvascular diseases (e.g. diabetes), cancer and congenital deficiencies, can lead to the loss of one or both upper limbs. This can have a life-changing impact in the amputee since the individual's ability to perform activities of daily live (ADLs) autonomously, the capability to work and socialize are suddenly limited [3].The levels of upper limb amputation can be classified as shown in Fig. 1.1.

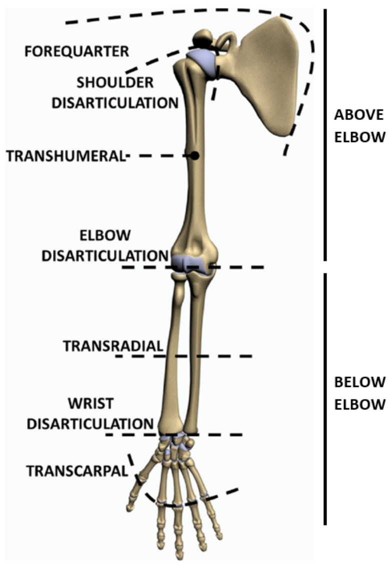


FIGURE 1.1: Upper limb amputation levels. Modified from [2]

Most patients suffering from an amputation require the use of a prosthesis to restore arm function and recover the ability to perform tasks of daily living and increase quality of life. Preferably, prosthesis should replace the function of the lost limb as close as possible, have a natural appearance, be lightweight and affordable. However, there exist an inevitable trade-off among such requirements as they are often hard to combine [4]. Taking those requirements as a base, this thesis is focused on improving the functionality of a prosthetic hand.

1.2 Demographics and Prevalence

Information related to upper limb loss is limited. A study carried out by Ziegler-Grahametal et al. estimated that a number of 541.000 (13.5 per 100.000) people suffer from different upper limb amputation levels only in the United Sate. The main causes of amputation were trauma (92%) and, in less extent, dysvascular disease (7%), and cancer (1%). The number of amputees is expected to double by year 2050, mainly because of dysvascular diseases, especially related to diabetes [5], [6]. Between the few studies carried out in Europe, Østlie et al. found an estimated population prevalence in Norway of 11.6 upper limb amputations per 100.000. Among the upper limb amputees, 95% were unilateral and amputations were caused by trauma (85%), cancer (7%), infection (6%), dysvascular disease (1-2%), and drug overdose (0-1%) [7].

1.3 Upper limb prostheses

The first prostheses date back to ancient Egypt 3000 years ago, when they were both used for cosmetic and functional purposes. As medicine techniques and technology development advanced, doctors were able to prepare the limb stump for emerging prosthetic devices [8]. Today, different types of passive and active upper limb prosthesis exist to serve wide individual needs and preferences.

Passive upper limb prostheses

Passive prostheses are not equipped with any mechanical nor electrical part to provide active grasping capabilities. Passive prosthesis can be subdivided into cosmetic (Fig.1.2a) and functional (Fig.1.2b). Cosmetic prosthesis are mainly designed for aesthetic substitution of the missing limb, while functional ones are designed to assist on specific activities by the use of a tool or hand. Passive functional prosthesis hands have an appearance close to the human hand and provide basic functionalities such as pushing and pulling. Passive functional tools have a mechanical appearance and are designed for activities requiring a special grasp type [1], [9].

(A) Cosmetic prothesis

(B) Prosthetic tool

FIGURE 1.2: Different examples of passive prosthesis

Body-powered prostheses

Body-powered prostheses are active prosthesis that consist of harnesses and cables fastened to the sound limb and to a terminal device, which can either be an artificial hand or a hook. Fig. 1.3 provides an example of their working mechanism. Apart from providing more functionality than passive prostheses, they are more durable and provide feedback to the user through tension cables. However, gross movements are needed to control this kind of prosthesis, which is unintuitive and requires high energy expenditure [1], [2], [11].

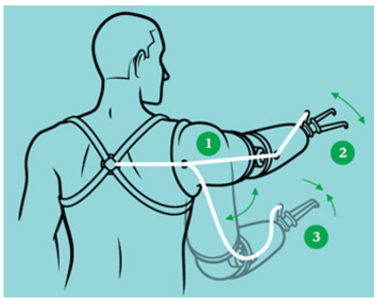


FIGURE 1.3: Body powered prosthesis with integrated cables and harnesses for mechanical control [10]

Externally powered prostheses

Externally powered prostheses are active prosthesis that exploit the power from an external source to drive electrical or hydraulic actuators to move the prosthesis. They generally offer greater control options and grasp force, at the expense of increased maintenance and weight due to built in batteries and electrical components. Nowadays, most commercially available advanced prostheses provide more than 1 DOF and allow for different grips and hand gestures to sequentially select the joints to be controlled. Examples of commercially available prosthesis are Ottobock's Michelangelo (Fig. 1.4a) , Touch Bionics' i-Limb (Fig. 1.4b) and Bebionic hand (Fig. 1.4c). However, with more than 2 DOF the sequential control implies high mental effort and becomes non-intuitive, not natural, slow, difficult to learn and prone to errors [2], [12]. Therefore, to develop an intuitive and robust Man-Machine Interface (MMI) to control active prosthesis is a current challenge researchers are facing.

(A) Michelangelo hand (B) i-Limb hand (C) Bebionic hand

FIGURE 1.4: Commercially available externally powered prosthesis

Man-Machine Interfaces

In prosthetics, there exist several types of MMIs that enable the control of the artificial limbs. More specifically, they depend and are classified based on the type of input extracted from user's intention. This can range from mechanical input (as explained in section 1.3), buttons, to Brain-Machine Interface (BMI) based on Electroencephalography (EEG) and Electrocorticography (ECoG) [13], [14]. However, the most widely used interface in market and research is myoelectric control (MYO), which makes use of surface electrodes (Surface Electromyography (sEMG)) placed in the skin of the residual limb. This electrodes capture the muscle signals driven through the Peripheral or Central Nervous System (PNS and CNS) and derived from a voluntary muscle contraction which enables an amputee to control movements over multiple DOFs [1]. Moreover, researchers have also studied the use of intramusculuar EMG which provides access to deeper muscles and avoiding cross-talk and electrode shift. Although sEMG is preferred for its non-invasive nature and ease of use in daily live, intramuscular EMG seems a promising technique as wireless implantable recording devices are being developed [15], [16].

After the amputation of an arm, the muscles of the amputated region are lost, but the nerves are still functional up to the amputation site. The concept of Targeted Muscle Reinervation (TMR) consists on reusing those residual nerves in healthy muscles to trigger their contraction. Therefore, a muscle that is no longer functional after the amputation is first denervated, divided into several segments surgically and reinnervated with the nerve related to the desired movement (Fig. 1.5). The newly innervated muscle serves as a biological amplifier of the neural commands. This way, when the amputee thinks of closing the hand, the corresponding muscle is activated by the nerve previously used to close the real hand, which triggers the artificial hand to close. This technique allows for more intuitive control over multiple DOFs [13], [14], [17].

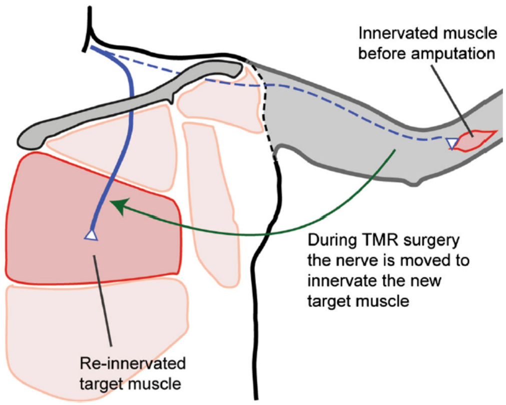


FIGURE 1.5: Targeted muscle reinnervation surgical procedure

1.4 Myoelectric Control

EMG Processing

EMG signal amplitude is usually below 10mV and its frequency content goes from 0 to 500 Hz, which makes it sensitive to noise and artefacts. For this reason, EMG is low and high pass filtered to exclude frequency content outside its common range. Further filtering is made depending on the application of interest and notch filters are applied to remove power line artefacts.

Additionally, EMG signals need to be analysed in real-time. This is made by using time segments called windows, where multiple samples are recorded and their amplitude is averaged. The amplitude of the EMG signal is calculated, for each electrode, by using the root-means-square value ($RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} EMG_i^2}$) or the mean-absolute value ($MAV = \frac{1}{n}\sum_{i=1}^{n} |EMG_i^2|$) for a given n number of samples in each window [18]. There are different windowing techniques, such as adjacent windowing and overlapping techniques, where part of the previous window is used to compute a new average. The use of windowing, windowing technique and length depends on the control framework or algorithm used, as computing the average of such windows adds computation time [19].

Following this process, user's intentions are deciphered and communicated to the motor controller in order to actuate the appropriate DOF. A range of different methods exist to translate information obtained from EMG signals to control commands. These are usually classified either as sequential or simultaneous. Sequential control actuates only 1 DOF at a time, while simultaneous control allows to actuate more than 1 DOF. Although sequential control is most used, research is focusing on simultaneous control of more than one DOF. In the following lines, current MYO control methods are presented.

Control Strategies

On-off control

On-off control is the earliest MYO control method. This control method is suitable only up to 2 DOFs. It actuates the hand at a constant velocity and switches the direction of the DOF when an EMG amplitude threshold is passed. This means that the velocity of actuation is independent of the muscle contraction level. This control strategy can be implemented using a pair of antagonist muscles or only one muscle, where two different contraction levels actuate two different directions [16], [17].

Proportional control

The human neuromotor control system can vary joint torques, velocities and positions at will and therefore, the 'on-off' prosthetic control was non-intuitive for humans. Hence, proportional control was developed. In such control scheme the velocity command given to the motors is proportional to the muscle contraction level. Moreover, contraction could not only be proportionally mapped to velocity, but also to other mechanical quantities such as force or position. Therefore, force control can be used to proportionally vary the grasp force, although this remains non-intuitive. Furthermore, position control can be used to proportionally prescribe position commands to the prosthesis. However, given the stochastic nature of the EMG signal, it would be difficult to reach specific position and even more, the amputee should maintain a constant contraction level to stay in one position which would cause fatigue. Hence, velocity control is preferred over force and position control as it provides smoother control and is less fatiguing for the user, as it is not needed to keep muscle contraction for the prosthesis to stay in a specific configuration. However, proportional control can be useful for gross prosthetic movements but not for finer control [16], [17], [20].

Fig. 1.6 provides an overview of the steps involved in each control method, where it can be seen that other control schemes are used along with proportional control.

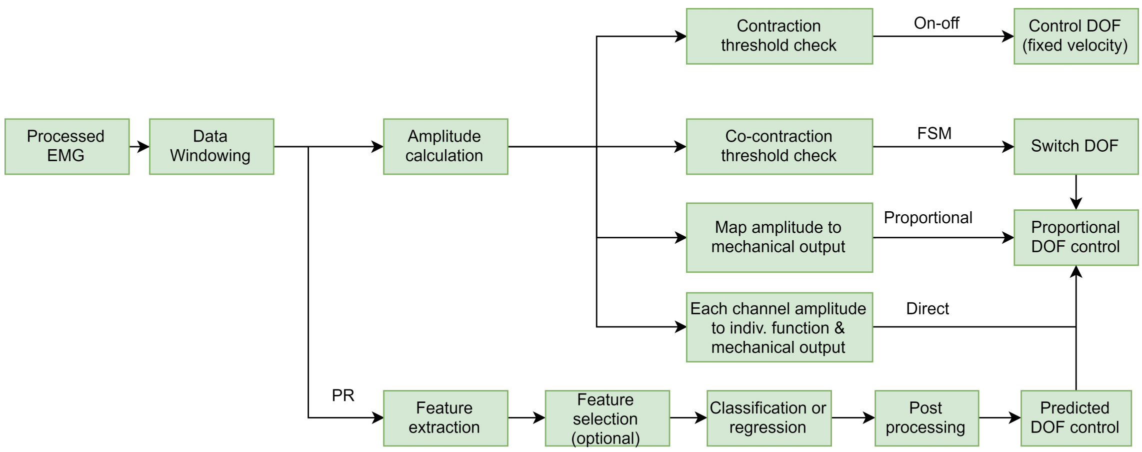


FIGURE 1.6: Scheme representing different MYO control methods. For on-off control a fixed velocity is used. In proportional control a mechanical output (force, velocity, position) can be used to drive the prosthesis. If multiple and cross-talk free control sites are available, direct control can be used to drive the prosthesis in a proportional way assigning one function to each of the control sites. In FSM two control sites can be used to select the desired DOF by co-contraction. In PR, features are extracted and used as input for a classifier or a regression analysis, where the output is the predicted, desired DOF.

Direct control

Proportional control can be refined by mapping individual EMG to individual motor functions on the prosthesis. Such control is called direct control, as there is a one-to-one relationship between EMG of one muscle and prosthetic command. However, the number of independent EMG signals available is limited and sensitive to cross-talk. Therefore implementing this control scheme for more than one DOF can be troublesome [1], [16],

[17]. However, this can be overcome by the use of intramuscular EMG, although individual control of each muscle is challenging for the amputees [21].

Finite state machine

Finite state machine (FSM) appears as an alternative to control prosthesis with more than one DOF when the number of EMG signals is limited, as in pure proportional and direct control. This method consist on switching the DOF to actuate by using different methods. Co-contraction (CC) is regularly used to switch between DOFs. This switching method consists on co-contracting two antagonistic muscles and, when a predefined amplitude threshold is passed, the DOF to actuate is changed (Fig. 1.7). Afterwards, the selected DOF is controlled proportionally [17]. This type of control means a non-intuitive, slow transfer of the user's intentions to the prosthesis and requires long training periods and high-mental effort [12]. However, proportional FSM remains as the default control method in commercially available prosthesis and is often used as the golden standard for comparison with new control methods [2].

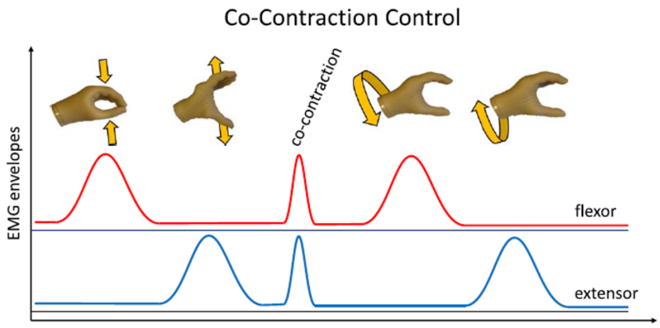


FIGURE 1.7: Co-contraction based FSM

Pattern Recognition

In order to overcome the limitations previously cited, the research community came up with techniques based on Pattern Recognition (PR). The core of PR consists on recognizing patterns in data that will be used to *predict* the output for new data instances. In MYO control, PR algorithms can be trained to recognize the patterns of muscle activation obtained from EMG signals to derive the motor intention (see Fig. 1.6). On this process, data is windowed and multiple features from time, frequency and time-frequency domain are extracted and used as input for the learning algorithm. A higher number of features can result in greater computational complexity and feature selection techniques are often implemented as an additional step to remove irrelevant features.

In this point, features can be classified or used as input for regression methods. In the first case, a classification algorithm is used, such as artifical neural networks (ANN) or linear discriminant analysis (LDA) and a class label (or DOF label in MYO control) is predicted [16]. However, classification can only predict one prosthetic function at a time. To solve this problem, multiple-class algorithms have been developed, although this increases the number of patterns to be trained which can be cumbersome [17], [22]. Furthermore, classification does not support proportional control of the prosthesis and this needs to be computed in parallel, which adds more complexity to the control algorithm.

New regression techniques are used to overcome the downsides of classification based PR. Through this technique, a regression is performed for each DOF where input features are mapped to output continuous value by using a selected function. Therefore, there is no need of computing proportional control in parallel. This allows for the possibility of simultaneous control of more than 1 DOF.

As a last stage, with the aim of removing wrongly predicted instances, post-processing techniques such as majority voting (MV) are used, where certain number of past prediction samples are used to see which one was predicted the most.

1.5 Challenges in Current Myoelectric Prosthesis

The earliest PR-EMG based prostheses date back to 1950s and 1960s and since then many PR based EMG controlled prosthesis studies, offline and online, have been carried out by the showing acceptable level of classification accuracy (>90%) and functionality [18], [23]. However, a study on the prosthesis use carried out in 1995 stated that only the 18% of amputees wears a MYO prosthesis [24], and more interestingly, that the 27% of amputees fitted with an active body-powered or MYO prosthesis use their prosthesis in a passive way. Moreover, a more recent study from 2007 reported that the overall prosthesis rejection rate was found to be as much as 23%, despite the great technological advancements achieved by the research community in the laboratory [11]. The main cause of such rejection for advanced prosthesis is due to the fact that the control interface they provide does not allow for natural control. Human movement implies simultaneous coordination and proportional movement of multiple DOFs across several joints at the same time while applying the right amount of force when handling objects or interacting in any other way with the world.

In this sense, PR-algorithms attempt to solve the challenging problem of modelling human biomechanics by encapsulating all neuromuscular variables (e.g. EMG, joint angles and torques) in a non-linear function. This way, the dynamics of the NMS system are reduced to a black box where the relationship of its variables is not explicitly modelled, which does not allow direct understanding of the transformation mechanics between variables [25]. It is because of this reason that new control strategies for prosthetic hands based on modelling the underlying biomechanics of human movement are being explored. Modelling true biomechanics would allow for more intuitive control, without the learning burden machine learning methods involve.

Furthermore, losing a limb implies the loss of the afferent pathways of sensory organs, such as muscle spindles and Golgi tendon organs. These organs provide position, velocity and force feedback of the controlled limb, which helps the individual to adjust his/her commands to interact with wide kinds of environments, such as grasping objects of different mechanical properties. The loss of such feedback pathways forces the amputee to rely on visual or auditory feedback to control the dynamics of the grasp. In the recent years, researchers have tried to restore such feedback pathways by fitting the amputee with a feedback unit, such as a vibrotactile cuff, to perceive e.g. touch force [26].

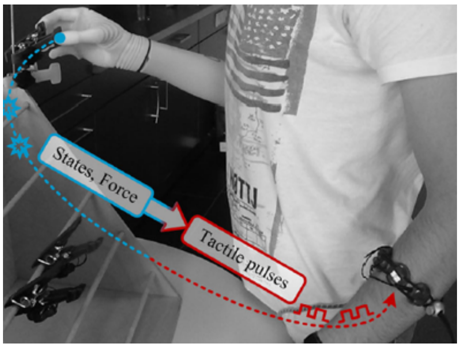


FIGURE 1.8: Example of haptic feedback. A vibrotactile stimulation unit informs the user about prosthesis states and forces [27].

Such type of feedback is also called *haptic* feedback. Nevertheless, the impact these techniques have in grasp force accuracy is still unclear. On the other hand, other researchers attempted to improve grasp force control by feeding back the forces exerted in the prosthetic hand to the prosthetic controller to update control commands. However, the techniques used are based on commanding small force or position increments or decrements to find a suitable grasping force, which neither provide an optimal nor natural way to control interaction forces (refer to Chapter 2, section I-B for an extended review on the mentioned techniques).

In conclusion, research is focused on creating new control strategies that resemble the way humans produce movement and interact with the world. This master thesis aims to provide a proof-of-concept control strategy that allows for intuitive control of the prosthetic hand's DOF's and grasp force control.

1.6 Developments in Prosthetic Control

Model Based Control

Unlike in pattern recognition, building computational models of the NMS system allows to extract parameters that cannot be measured experimentally. This is useful to gain insight on how the internal variables of the NMS change on time, like muscle length and forces. This way, subject-specific NMS can be created and used to plan rehabilitation or design rehabilitation robotics, simulate neuromuscular disorders and analyse motor control [25], [28]. In the scheme of upper limb prosthesis, EMG can provide an experimental interface to extract neural information of an individual's intended movement and estimate the biomechanics of an amputated limb, such as forces and torques, which are transformed and used to control the prosthetic hand. This control scheme is called *EMG-driven musculoskeletal modelling* based control. Recently, Sartori et al. [29] demonstrated the use of EMG-driven musculoskeletal modelling to control a prosthetic hand. Such control scheme showed truly intuitive, human-like prosthetic control over multiple DOFs simultaneously. Because of these reasons, this master thesis is based on the work of Sartori et al.

More in depth, in EMG-driven musculoskeletal modelling, joint-kinematics are extracted by constructing a subject-specific model of the individual's anthropometry, which is used to estimate various parameters of the MTUs, such as length and moment arms (Fig. 1.9 b)). MTU-dynamics (Fig. 1.9 c)) are estimated from the obtained MTU-kinematics and the neural excitations extracted from EMG (Fig. 1.9 a)). The calibration of the model is performed for muscle parameter values that vary non-linearly across subjects [30]. Ultimately, the predicted joint torques (herewith also referred as *moments*) can be transformed and used as input for the controlled device. A more detailed explanation of the NMS model's complete pipeline can be found in chapter 2, section II-B.

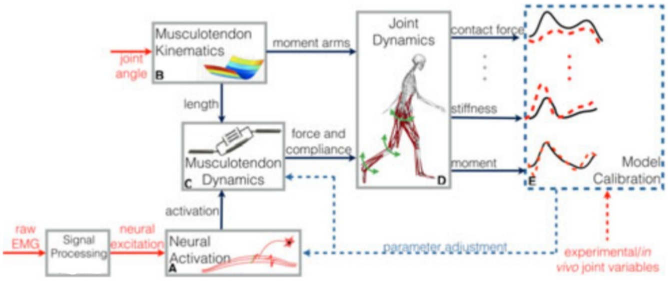


FIGURE 1.9: General NMS modelling pipeline. Modified from [25].

Admittance control: interacting with the environment

In section 1.5, the problem of fine grasp force control in current MYO prosthesis was introduced. Along with this, the efforts of the research community to restore amputees' ability for a better force control were introduced. Here, some researchers tried to control the force *or* the motion of the hand to increase or decrease the interaction force between the prosthetic and an object, for example. Instead of controlling either force or motion, an idea could be to control both variables at once to see what effect causes one on the other: any applied force to an object will cause some degree of deformation on the object and the object will counteract this deformation with a reaction force in opposite direction. Therefore, the key for a safe interaction is to *model the relationship between the forces and the motion* caused by this forces, rather than each of the two variables alone. The *dynamic relation* between force and motion is called *mechanical admittance*. To illustrate this concept an example is provided in the following lines.

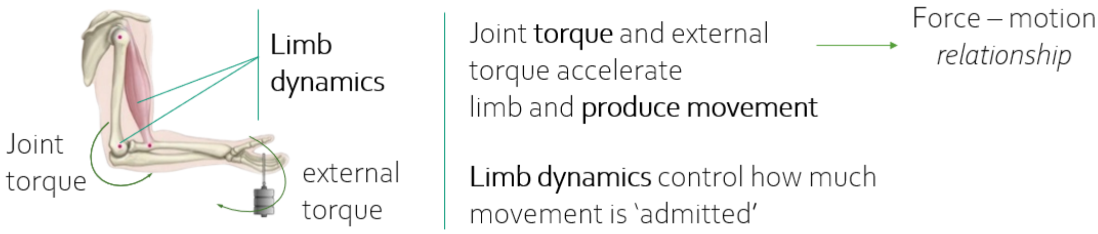


FIGURE 1.10: Illustration of the concept of admittance.

In Fig. 1.10, an arm holding a weight is illustrated. The forces (referred as torques in the figure for context correctness) acting on the arm are: the one produce by the weight and the one produce by the arm (which tries to counteract the weight). The balance between the two forces accelerates and moves the limb, i.e. if the force applied by the human is greater than the weight, the arm will lift and move the weight up. The amount of motion the arm *admits* is ruled by the dynamics of the arm. Then, we can say that by controlling the admittance of the arm we can control motion produced by the forces acting on it. In the fields of robotics and biorobotics, such control paradigm is used to control the *desired* behaviour of the robot (e.g. robotic arm or an exoskeleton) and is often named as *admittance control*.

In admittance control, a virtual model of the dynamics with which we *wish* the controlled robotic device to move is designed. Therefore, such dynamics can also be called *desired* dynamics. This master thesis makes use of the paradigm of admittance control to model the grasp dynamics while accounting for external forces arisen from the interaction with the environment (refer to A, section II-D).

1.7 Scope of Thesis: Towards bio-inspired grasp force control

This thesis focuses on exploiting the opportunities that EMG-driven musculoskeletal modelling and admittance control provide to *improve grasp force control in upper limb MYO-prosthesis*. More specifically, the control strategy proposed in this master thesis focuses on improving the control framework proposed in the work done by Sartori et al. [29].

Bibliography

[1] D. Farina, W. Jensen, and M. Akay, Eds., *Introduction to neural engineering for motor rehabilitation*, Hoboken, N.J: John Wiley & Sons, 2013, 579 pp., ISBN: 978-0-470-91673-5.

[2] F. Cordella, A. L. Ciancio, R. Sacchetti, A. Davalli, A. G. Cutti, E. Guglielmelli, and L. Zollo, "Literature review on needs of upper limb prosthesis users", *Frontiers in Neuroscience*, vol. 10, May 12, 2016, ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00209. [Online]. Available: http://journal.frontiersin.org/Article/10.3389/fnins.2016.00209/abstract (visited on 09/20/2019).

[3] K. Demet, N. Martinet, F. Guillemin, J. Paysant, and J.-M. André, "Health related quality of life and related factors in 539 persons with amputation of upper and lower limb", *Disability and Rehabilitation*, vol. 25, no. 9, pp. 480–486, Jan. 2003, ISSN: 0963-8288, 1464-5165. DOI: 10.1080/0963828031000090434. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/0963828031000090434 (visited on 09/20/2019).

[4] E. Biddiss, "Need-directed design of prostheses and enabling resources", in *Amputation, Prosthesis Use, and Phantom Limb Pain*, C. Murray, Ed., New York, NY: Springer New York, 2009, pp. 7–21, ISBN: 978-0-387-87461-6 978-0-387-87462-3. DOI: 10.1007/978-0-387-87462-3_2. [Online]. Available: http://link.springer.com/10.1007/978-0-387-87462-3_2 (visited on 09/22/2019).

[5] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the united states: 2005 to 2050", *Archives of Physical Medicine and Rehabilitation*, vol. 89, no. 3, pp. 422–429, Mar. 2008, ISSN: 00039993. DOI: 10.1016/j.apmr.2007.11.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0003999307017480 (visited on 09/22/2019).

[6] S. Wild, G. Roglic, A. Green, R. Sicree, and H. King, "Global prevalence of diabetes: Estimates for the year 2000 and projections for 2030", *Diabetes Care*, vol. 27, no. 5, pp. 1047–1053, May 1, 2004, ISSN: 0149-5992, 1935-5548. DOI: 10.2337/diacare.27.5.1047. [Online]. Available: http://care.diabetesjournals.org/cgi/doi/10.2337/diacare.27.5.1047 (visited on 09/22/2019).

[7] K. Østlie, O. H. Skjeldal, B. Garfelt, and P. Magnus, "Adult acquired major upper limb amputation in norway: Prevalence, demographic features and amputation specific features. a population-based survey", *Disability and Rehabilitation*, vol. 33, no. 17, pp. 1636–1649, Jan. 2011, ISSN: 0963-8288, 1464-5165. DOI: 10.3109/09638288.2010.541973. [Online]. Available: http://www.tandfonline.com/doi/full/10.3109/09638288.2010.541973 (visited on 09/22/2019).

[8] A. J. Thurston, "PARÉ AND PROSTHETICS: THE EARLY HISTORY OF ARTIFICIAL LIMBS", *ANZ Journal of Surgery*, vol. 77, no. 12, pp. 1114–1119, Dec. 2007, ISSN: 1445-1433, 1445-2197. DOI: 10.1111/j.1445-2197.2007.04330.x. [Online]. Available: http://doi.wiley.com/10.1111/j.1445-2197.2007.04330.x (visited on 09/25/2019).

[9] B. Maat, G. Smit, D. Plettenburg, and P. Breedveld, “Passive prosthetic hands and tools: A literature review”, *Prosthetics and Orthotics International*, vol. 42, no. 1, pp. 66–74, Feb. 2018, ISSN: 0309-3646, 1746-1553. DOI: 10.1177/0309364617691622. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0309364617691622 (visited on 09/25/2019).

[10] M. Klerstad and M. Svensson, “Creating a body powered prosthetic arm with 3d printing technology”, p. 75,

[11] E. A. Biddiss and T. T. Chau, “Upper limb prosthesis use and abandonment: A survey of the last 25 years”, *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236–257, Sep. 2007, ISSN: 0309-3646, 1746-1553. DOI: 10.1080/03093640600994581. [Online]. Available: http://journals.sagepub.com/doi/10.1080/03093640600994581 (visited on 09/22/2019).

[12] C. Castellini, P. Artemiadis, M. Wininger, A. Ajoudani, M. Alimusaj, A. Bicchi, B. Caputo, W. Craelius, S. Dosen, K. Englehart, D. Farina, A. Gijsberts, S. B. Godfrey, L. Hargrove, M. Ison, T. Kuiken, M. MarkoviÄ‡, P. M. Pilarski, R. Rupp, and E. Scheme, “Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography”, *Frontiers in Neurorobotics*, vol. 8, Aug. 15, 2014, ISSN: 1662-5218. DOI: 10.3389/fnbot.2014.00022. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnbot.2014.00022/abstract (visited on 09/25/2019).

[13] J. Ribeiro, F. Mota, T. Cavalcante, I. Nogueira, V. Gondim, V. Albuquerque, and A. Alexandria, “Analysis of man-machine interfaces in upper-limb prosthesis: A review”, *Robotics*, vol. 8, no. 1, p. 16, Feb. 21, 2019, ISSN: 2218-6581. DOI: 10.3390/robotics8010016. [Online]. Available: http://www.mdpi.com/2218-6581/8/1/16 (visited on 09/26/2019).

[14] A. E. Schultz and T. A. Kuiken, “Neural interfaces for control of upper limb prostheses: The state of the art and future possibilities”, *PM&R*, vol. 3, no. 1, pp. 55–67, Jan. 2011, ISSN: 19341482. DOI: 10.1016/j.pmrj.2010.06.016. [Online]. Available: http://doi.wiley.com/10.1016/j.pmrj.2010.06.016 (visited on 09/26/2019).

[15] L. H. Smith and L. J. Hargrove, “Comparison of surface and intramuscular EMG pattern recognition for simultaneous wrist/hand motion classification”, in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2013, pp. 4223–4226. DOI: 10.1109/EMBC.2013.6610477.

[16] P. Geethanjali, “Myoelectric control of prosthetic hands: State-of-the-art review”, *Medical Devices: Evidence and Research*, vol. Volume 9, pp. 247–255, Jul. 2016, ISSN: 1179-1470. DOI: 10.2147/MDER.S91102. [Online]. Available: https://www.dovepress.com/myoelectric-control-of-prosthetic-hands-state-of-the-art-review-peer-reviewed-article-MDER (visited on 09/26/2019).

[17] A. D. Roche, H. Rehbaum, D. Farina, and O. C. Aszmann, “Prosthetic myoelectric control strategies: A clinical perspective”, *Current Surgery Reports*, vol. 2, no. 3, p. 44, Mar. 2014, ISSN: 2167-4817. DOI: 10.1007/s40137-013-0044-8. [Online]. Available: http://link.springer.com/10.1007/s40137-013-0044-8 (visited on 09/26/2019).

[18] E. Scheme and K. Englehart, “Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use”, *The Journal of Rehabilitation Research and Development*, vol. 48, no. 6, p. 643, 2011, ISSN: 0748-7711. DOI: 10.1682/JRRD.2010.09.0177. [Online]. Available: http:

//www.rehab.research.va.gov/jour/11/486/pdf/scheme486.pdf (visited on 10/02/2019).

[19] M. Hakonen, H. Piitulainen, and A. Visala, "Current state of digital signal processing in myoelectric interfaces and related applications", *Biomedical Signal Processing and Control*, vol. 18, pp. 334–359, Apr. 2015, ISSN: 17468094. DOI: 10.1016/j.bspc.2015.02.009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S174680941500021X (visited on 10/01/2019).

[20] A. Fougner, O. Stavdahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, "Control of upper limb prostheses: Terminology and proportional myoelectric control—a review", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 5, pp. 663–677, Sep. 2012, ISSN: 1534-4320, 1558-0210. DOI: 10.1109/TNSRE.2012.2196711. [Online]. Available: http://ieeexplore.ieee.org/document/6205630/ (visited on 09/30/2019).

[21] L. H. Smith, T. A. Kuiken, and L. J. Hargrove, "Real-time simultaneous and proportional myoelectric control using intramuscular EMG", *Journal of Neural Engineering*, vol. 11, no. 6, p. 066 013, Dec. 1, 2014, ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2560/11/6/066013. [Online]. Available: http://stacks.iop.org/1741-2552/11/i=6/a=066013?key=crossref.edc34b0f5a0b7a836ef83b5d18d6ba47 (visited on 09/30/2019).

[22] A. J. Young, L. H. Smith, E. J. Rouse, and L. J. Hargrove, "Classification of simultaneous movements using surface EMG pattern recognition", *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 5, pp. 1250–1258, May 2013. DOI: 10.1109/TBME.2012.2232293.

[23] N. Jiang, S. Dosen, K. Muller, and D. Farina, "Myoelectric control of artificial limbs #8212;is there a need to change focus? [in the spotlight]", *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 152–150, Sep. 2012. DOI: 10.1109/MSP.2012.2203480.

[24] T. W. Wright, A. D. Hagen, and M. B. Wood, "Prosthetic usage in major upper extremity amputations", *The Journal of Hand Surgery*, vol. 20, no. 4, pp. 619–622, Jul. 1995, ISSN: 03635023. DOI: 10.1016/S0363-5023(05)80278-3. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0363502305802783 (visited on 10/03/2019).

[25] M. Sartori, D. G. Llyod, and D. Farina, "Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies", *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 5, pp. 879–893, May 2016, ISSN: 0018-9294, 1558-2531. DOI: 10.1109/TBME.2016.2538296. [Online]. Available: https://ieeexplore.ieee.org/document/7440844/ (visited on 10/03/2019).

[26] C. Antfolk, M. D'Alonzo, B. Rosén, G. Lundborg, F. Sebelius, and C. Cipriani, "Sensory feedback in upper limb prosthetics", *Expert Review of Medical Devices*, vol. 10, no. 1, pp. 45–54, Jan. 2013, ISSN: 1743-4440, 1745-2422. DOI: 10.1586/erd.12.68. [Online]. Available: http://www.tandfonline.com/doi/full/10.1586/erd.12.68 (visited on 03/01/2019).

[27] M. Markovic, M. A. Schweisfurth, L. F. Engels, T. Bentz, D. Wüstefeld, D. Farina, and S. Dosen, "The clinical relevance of advanced artificial feedback in the control of a multi-functional myoelectric prosthesis", *Journal of NeuroEngineering and Rehabilitation*, vol. 15, no. 1, Dec. 2018, ISSN: 1743-0003. DOI: 10.1186/s12984-018-0371-1. [Online]. Available: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-018-0371-1 (visited on 02/21/2019).

[28] F. Valero-Cuevas, H. Hoffmann, M. Kurse, J. Kutch, and E. Theodorou, "Computational models for neuromuscular function", *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 110–135, 2009, ISSN: 1937-3333, 1941-1189. DOI: 10.1109/RBME.2009.2034981. [Online]. Available: http://ieeexplore.ieee.org/document/5342785/ (visited on 10/15/2019).

[29] M. Sartori, G. Durandau, S. Došen, and D. Farina, "Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling", *Journal of Neural Engineering*, vol. 15, no. 6, p. 066 026, Dec. 1, 2018, ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/aae26b. [Online]. Available: http://stacks.iop.org/1741-2552/15/i=6/a=066026?key=crossref.24203caa63d0aa7e7a24cbd38ab35e99 (visited on 02/21/2019).

[30] M. Sartori, M. Reggiani, D. Farina, and D. G. Lloyd, "EMG-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity", *PLoS ONE*, vol. 7, no. 12, P. L. Gribble, Ed., e52618, Dec. 26, 2012, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0052618. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0052618 (visited on 10/15/2019).

[31] C. Pizzolato, D. G. Lloyd, M. Sartori, E. Ceseracciu, B. J. Fregly, and M. Reggiani, "CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks", p. 23, 2016.

[32] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

[33] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, "Admittance control for physical human–robot interaction", *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, Sep. 2018, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364918768950. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364918768950 (visited on 02/21/2019).

[34] M. Focchi, G. A. Medrano-Cerda, T. Boaventura, M. Frigerio, C. Semini, J. Buchli, and D. G. Caldwell, "Robot impedance control and passivity analysis with inner torque and velocity feedback loops", *Control Theory and Technology*, vol. 14, no. 2, pp. 97–112, May 2016, ISSN: 2095-6983, 2198-0942. DOI: 10.1007/s11768-016-5015-z. [Online]. Available: http://link.springer.com/10.1007/s11768-016-5015-z (visited on 11/13/2019).

Chapter 2

Paper

Myoelectric control of bionic hands via musculoskeletal modelling, admittance control and force feedback

Mikel Mateo, BSc, Massimo Sartori, *Member, IEEE,* Arvid QL Keemink,
Guillaume Durandau, *Student Member, IEEE*

***Abstract*—Current myoelectric upper limb prostheses do not provide truly bio-mimetic control that allows for both intuitive control and reliable grasp dynamics. This is mainly because current machine learning approaches provide a control interface that is non-intuitive and extraordinarily different from human natural control. Furthermore, the lack of afferent feedback pathways forces the amputee to rely mainly on visual feedback to control grasp action, where the implementation of state-of-the-art haptic feedback techniques does not show conclusive results on the help of grasp force control. In this work, a subject-specific EMG-driven musculoskeletal model coupled to an admittance controller is used to drive a bionic hand. The admittance model virtually mimics the grasp dynamics of a real hand interacting with the environment. This way, the prosthetic hand (Michelangelo hand, Ottobock) can readjust its control commands during interaction by accounting for external forces. Theoretical stability boundaries of the control system are analysed for stable interaction and experimental tests are carried out comparing the proposed control framework to non-admittance based EMG-driven musculoskeletal modelling during grasping tasks. Experimental outcomes show positive results when compared to non-admittance based control. This control framework sets the bases to enable safer prosthesis-environment interaction without the need of constant visual feedback.**

***Index Terms*—admittance model, control system stability, EMG-driven musculoskeletal modelling, myoelectric control, system identification.**

I. Introduction

THE loss of an upper limb can have a profound impact in the quality of life, since the individual's ability to perform activities of daily live (ADLs) autonomously, the capability to work and socialize are suddenly limited [1]. Myoelectric prosthesis (MYO) aim to restore the missing functions by providing a man-machine interface based on electromyography (EMG) which enables the amputee to control the artificial limb. Surface electrodes placed in an antagonistic muscle pair in the residual limb capture the muscle signals driven through the neuromotor system, after which they are processed and used as input to control prosthetic joints. Nowadays, most commercially available advanced prostheses provide control of more than one degree of freedom (DOF) and allow for different grips and hand gestures by the co-contraction of two muscles to sequentially select the different joints to be controlled. Examples of commercially available prosthesis are Ottobocks Michelangelo, Touch Bionics i-Limb and Bebionic. However, with more than 2 DOF the sequential control implies high mental effort and becomes non-intuitive, not natural, slow, difficult to learn and prone to errors [2], [3]. Moreover, the lack of proprioceptive feedback forces the user to rely primarily on visual information, which leads to excessive or poor grasp forces [4]. Therefore, despite the great technological advancements, studies stated that the overall prosthesis rejection rate was found to be as much as 23% [5].

The principal problem of MYO-controlled prosthesis is that it is extraordinarily different from human natural control, which requires simultaneous coordination and proportional movement of multiple DOFs across several joints at the same time, while applying the right amount of force when handling fragile objects or interacting with humans and animals [6].

The ultimate goal of upper limb prosthetics is to replicate the human control strategy to provide truly bio-mimetic control that allows for safe grasp dynamics. However, several challenges arise in this path given the state of current MYO-prosthesis.

A. Challenges in Current Myoelectric Prosthesis

Approaches based on machine learning algorithms, like pattern recognition (PR) and regression techniques, have been employed for prosthetic control to resolve the complexity of mimicking human-like control. As such, these techniques do not capture the underlying biomechanics of the human neuromusculoskeletal system (NMS) and they rather reduce the relation between neuro-biomechanical variables to a black box. Moreover, the same movement can be produced by different muscle recruitment strategies due to a redundant number of muscle tendon units (MTUs) controlling a limited number of DOFs. PR-algorithms are sensitive to this fact due to specific training conditions which fail to generalize to new, untrained conditions [6]. Ultimately, they fail to create a bio-mimetic prosthetic interface.

In addition, the loss of a limb carries the loss of the afferent pathways of sensory organs, such as muscle spindles and Golgi tendon organs. These organs provide the position, velocity and force feedback needed close the control loop and adjust neural commands. Currently, amputees depend on mostly visual or auditory feedback from the prosthesis to control the grasping dynamics while their attention is drawn away from the task at hand [7]. Furthermore, the use of prosthetic hands disables the internal (or feed-forward) models of object grasping of the amputees and therefore excessive grasp forces are common in prosthesis wearers [8], [9]. To account for this fact, haptic feedback techniques are being developed to recover lost proprioception and exteroception where the amputee is fitted with a feedback unit, such as a vibrotactile cuff. However, although haptic feedback is able to characterize, for example, grip force intensity, it cannot transmit if applied forces are insufficient or excessive and therefore is not able to guide amputees response[9]. In this scenario, the results regarding the benefits of haptic feedback for grasp force control in the literature are inconclusive and contradictory [10].

In conclusion, the existing gap between the intent of the user, control strategy and reliable interaction with the environment lies in the absence of a bio-inspired model that captures their relationship. This is, there is currently no human-like prosthetic control strategy which allows for an intuitive control and which automatically readjusts its control commands when excessive grasping forces are achieved through the derived intent of the amputee. Therefore, this work is focused on building a control strategy that a) allows for an intuitive control by EMG-driven musculoskeletal (MS) modelling, and b) captures the *dynamic relation* between the predicted forces by such model, phantom limb dynamics and contact forces arising from grasping real objects. As feedback pathways to the user are still an inconclusive research topic, they are left as a future development to complement and complete the proposed human-like control strategy.

B. Towards bio-mimetic control

Unlike machine learning methods, EMG-driven musculoskeletal (MS) modelling provides truly intuitive and robust MYO control over multiple DOFs simultaneously by estimating phantom limb biomechanics out of the EMG signals. In EMG-driven MS modelling, a subject-specific model of the individuals anthropometry is used to estimate net torque over multiple DOFs. Forces and torques produced by individual MTUs are estimated out of their kinematics (length, velocity and moment-arm) and from EMG extracted neural excitations. The MS model is calibrated in a closed loop formulation for anatomical and physiological internal parameters that vary nonlinearly across individuals [11], [12]. In short, MS decodes motor intentions and transforms them into joint torques by taking into account subject's limb kinematics and dynamics.

In the scheme of upper limb prosthesis, Sartori et al. [13] developed a subject-specific, real-time simultaneous multi-joint prosthetic control using an EMG-driven MS model. In that work, the predicted joint torques where directly mapped to joint velocity which was used as input for the control of an upper limb prosthesis. Hence, the prosthetic device was used as a physical integrator into device position, thus eliminating the need of numerical integration of velocity into position. However, the direct mapping from joint torque to joint velocity does not take into account limb dynamics and possible displacements caused by external forces acting on the prosthetic hand. That is, the prosthesis still cannot react to the environment by readjusting its commands when is in contact with an object. Therefore, the dynamic relation between predicted joint torque, external forces and output motion needs to be modelled to achieve truly bio-mimetic control that allows for robust interaction with real objects.

Several studies attempted to integrate external forces in prosthetic control with approaches based on hybrid force-position control with sliding-mode control [14] and object stiffness estimation [15], parallel force-position control [16], artificial neural network (ANN) models to estimate new commands from net external forces [17] and normal to shear force ratio based rules [18], where in many of them the velocity/position/force input signal of the artificial hand is increased until a predefined interaction force threshold is reached. However, none of them established a dynamic relation between input and output (velocity/force) but rather controlled them separately. Moreover, the mentioned approaches highly depend on the kind of prosthesis used (e.g. actuation mechanics, sensors available etc.) which does not help to create a generic solution for this problem.

In this sense, the limb dynamics that convert net torques to motion can be regarded as a *mechanical admittance*. The concept of admittance can be explained as the amount of motion the limb 'admits' for an input torque. Then, admittance *controls* the relation between torque and motion. Admittance control is the inverse of the impedance control paradigm proposed in the works by Hogan [19] and Colgate et al. [20] and which allows to control both motion and force at once. This prevents from controlling each variable separately by modelling their relation, unlike the techniques described above. More explicitly, in admittance control, the relation between force and motion is modelled through a virtual model of the dynamics of the physical system. This virtual model can be considered as the *desired dynamics* the controlled device is aimed to follow. Admittance control-based approaches have previously been used in bio-robotics for exoskeleton control, rehabilitation purposes and movement support to enable the controlled device to respond with desired dynamics [21], [22].

The work in this paper focuses on exploiting the opportunities that EMG-driven MS modelling provides to integrate external forces from the interaction with real environments through an admittance model of the *desired* hand dynamics. Such dynamics are the representation of the forward dynamics of the biomechanics of the hand. This way, the prosthesis is capable of reacting to grasp forces and automatically adjusting velocity control commands for interaction. This scheme provides a proof-of-concept that aims to set the bases to create a generalized upper limb prosthesis control strategy that reproduces the way humans generate movement and interact with the world in a robust manner. We hypothesize that the proposed control method will allow for fine grasp force control by accounting for interaction forces without the amputee needing to readjust control commands based only in visual feedback. The control scheme is evaluated, in terms of performance and robustness, in comparison to the non-admittance-based EMG-driven MS model control scheme proposed by Sartori et al. by performing three experimental tasks with a prosthetic hand. Furthermore, the stability of the system for different environment mechanical properties is assessed.

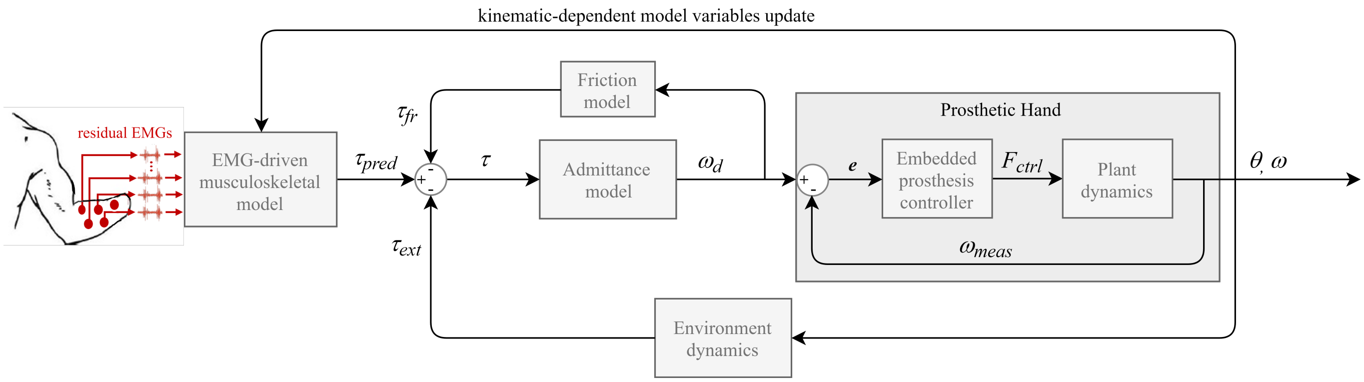


Fig. 1. Proposed control framework composed of the EMG-driven MS model, the admittance model coupled with a joint friction model and the prosthetic hand itself. The EMG-driven MS-model predicts joint torques τ_{pred} based on EMG-derived neural activations generated by subject's intended movement. The admittance model represent the dynamics of the open/close of the hand while interacting with environment, which produces a reaction torque τ_{ext}. The friction model reproduces the biological friction conditions in which joints move and produces a friction torque τ_{fr}. This admittance model coupled to the friction model transforms the net torque τ into desired angular velocity references ω_d. The desired velocity is used as reference for the prosthetic hand controller. The error e is the difference between measured velocity ω_{meas} and ω_d. The hand controller attempts to drive the prosthetic device by applying a control force F_{ctrl}. The output kinematics are ω_{meas} and prosthesis angular position θ.

II. Materials and Methods

The control framework presented in this paper is an extension of the work of Sartori et al. [13]. The focus of this paper is to *improve* the grasping force dynamics by accounting for external forces. This is done through an admittance model, coupled with a joint friction model, which reproduces, to some extent, the desired dynamics of a real hand interacting with the external world. The proposed control strategy is depicted in Fig. 1 and comprises of three major components including: the EMG-driven MS model, the admittance model coupled with a joint friction model and the prosthetic hand. The framework allows for simultaneous control of 2 DOFs: hand opening-closing (HOC) and wrist pronation-supination (WPS). However, because this paper focuses on grasp dynamics, only HOC was used for our experiments to test the implemented admittance model (see section IV for further explanation).

The experimental procedures were performed by three healthy subjects on two consecutive sessions. First, a generic MS model was scaled and calibrated for each subject's anthropometry and force-generating capacity. In the second session, the subject-specific model was used to perform the experimental tasks.

In the following lines, we first describe how anthropometry data were collected for establishing subject-specific MS model (see section II-A). Secondly, we describe the components of our proposed control strategy (sections II-B to II-D) and explain the processes to find the theoretical stability boundaries of the prosthesis-environment interaction (section II-G). Thirdly, the online prosthesis control experimental procedures are described (see section II-H).

A. Data recording and processing

A generic dynamic upper limb model was scaled on OpenSim for each subject. For this, the anatomical position of the shoulder joint, elbow, wrist, metacarpophalangeal (MCP) joint of the middle finger and index finger length were taken for each subject. Scaling of the MS model was therefore done using manual scale factors for each participant instead of using motion capture data for the ease of the subjects. Anatomical measures were taken in natural position, with no elbow flexion, such that the position of the arm matched the pose of the generic OpenSim model. Although the generic MS model provides all DOFs and MTUs in the human hand [23], only a subset of these were employed.

All experimental procedures were performed using a powered multi-functional Michelangelo prosthetic hand (Ottobock HealthCare GmbH, Duderstadt, DE) equipped with WPS and HOC motors and passive flexion-extension (WFE). The hand can produce lateral grasp and palmar grasp, where only palmar grasp (HOC) was used for our experiments.

EMG was recorded using 8 EMG Ottocbock electrodes connected to the central control unit of the prosthesis (AxonMaster 13E500, Ottobock). EMG electrodes consisted of an on-board 90-450 Hz bandpass filter and 50Hz notch filter [24]. Electrodes were placed in eight upper limb muscle groups including: biceps brachii, pronator teres, extensor carpi radialis, extensor carpi ulnaris, extensor digitorum, flexor carpi radialis, flexor carpi ulnaris and flexor digitorum [13]. The subjects were asked to perform a set of movements for EMG normalization including: wrist flexion, extension, pronation, supination, ulnar deviation, radial deviation, hand open and close and rest (no movement). EMG values for each channel were normalized by the highest processed value during the movement phases and removed base line by the mean value during no movement.

The prosthetic hand is sensorized with embedded force sensor positioned at the base of the thumb measuring grasping force (percentage of maximal force) and a position sensor measuring aperture size and wrist rotation angle. The maximum published grip force of the Michelangelo hand is 70 N in palmar grasp [25]. For finer reference, the embedded force sensor was calibrated using an external load cell placed normally to the embedded sensor giving a range up to 74 N, which is in line with similar experiments done by the research community [25]. The commands to the hand and the sensor

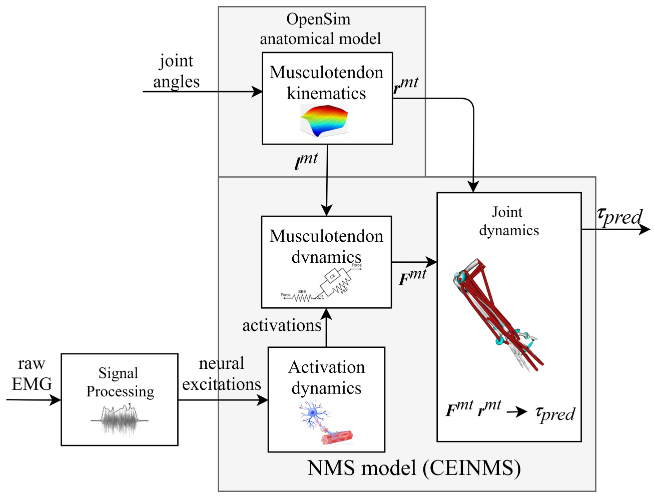


Fig. 2. Schematic structure of EMG-driven MS model. Neural excitations are obtained from raw EMG signals as linear envelopes and transferred into muscle activations. MTU lengths (l^{mt}) and moment arms (r^{mt}) are obtained from the input joint angles using multi-dimensional cubic B-splines. MTU forces (F^{mt}) are obtained from muscle activation and l^{mt} using a Hill-type muscle model. F^{mt} is converted into MTU torque (τ_{pred}) by multiplication with r^{mt}.

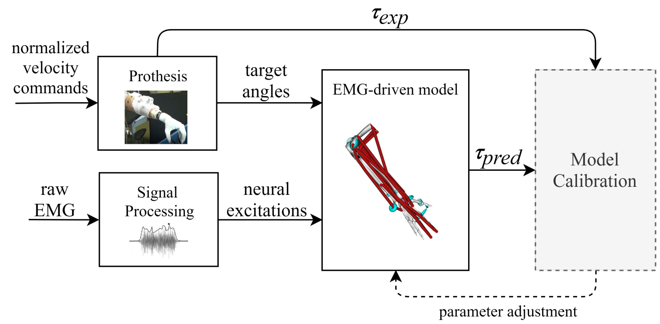


Fig. 3. EMG-driven MS model calibration is performed by comparing predicted torques (τ_{pred}) with experimental torques (τ_{exp}) obtained from prosthetic commands.

data, including processed EMG signals, were transmitted via Bluetooth connection between the AxonMaster unit and a PC at a frequency of 100 Hz.

B. EMG-driven Musculoskeletal Modelling

Limb biomechanics were estimated using Calibrated EMG-Informed NMS Modelling Toolbox (CEINMS) [12], hereafter referred as EMG-driven musculoskeletal (MS) model. The proposed EMG-driven MS model takes as input a) joint angles from prosthetic device and b) raw EMG from amputee's residual limb and outputs joint torques of the phantom limb and intact limb. The EMG-driven MS model is based on four key components: activation dynamics, MTU kinematics, MTU dynamics and joint dynamics. The *activation dynamics* component allocates experimentally measured EMG linear envelopes from 8 muscles to 14 MTUs in the model as detailed in Table I. MTU-allocated activations were further processed by a second order muscle twitch model and adjusted to account for the non-linear EMG-to-force relationship. The *MTU kinematics* component first synthesises subject-specific MTU paths into multidimensional cubic B-splines. Such splines are then used to produce estimates of MTU length (l^{mt}) and three-dimensional moment arms (r^{mt}) as a function of prosthetic joint angles. The *MTU dynamics* component computes individual MTU forces (F^{mt}) obtained from MTU-specific activations and l^{mt} using a Hill-type muscle model [11], [13]. The *joint dynamics* predicts joint torques (τ_{pred}) by the sum of the products of r^{mt} and F^{mt}.

C. Model Calibration

The calibration process consisted on instructing the subjects to mimic predefined motions of the prosthesis. For this exercise, subjects were asked to move through the whole range of motion of the specified DOF while the prosthesis moves at constant speed. Motions included opening and closing of the hand and wrist pronation-supination. The calibration algorithm receives: a) EMG from subject, b) prosthesis DOF angles and c) normalized velocity control commands of the prosthesis, representing the torque associated to reach specific prothesis DOF angles. The calibration of the model is performed for muscle parameter values that vary non-linearly across subjects, including: EMG to-activation non-linearity factor, muscle optimal fiber length, tendon slack length, and muscle maximal isometric force. The initial parameter values are iteratively refined through a simulated annealing algorithm so that the error between the model's predicted torques (τ_{pred}) and normalized prosthetic control commands (velocities representing prothesis torques (τ_{exp})) is minimized [12], [13]. The calibration algorithm receives information from pre-recorded data.

D. Admittance Model

The predicted torques by the EMG-driven MS model are converted into desired angular velocity references (ω_d) via an admittance model of the phantom hand (see Fig. 1). That is, the predicted torque will virtually accelerate a model of the inertial dynamics of the phantom hand causing virtual motion references that will be forwarded to the prosthetic low-level controller. Because external forces are also considered to be acting on the virtual dynamics of the phantom hand, velocity references will change if any external force is present. In short, there is a dynamic relation between force and velocity and there is no longer the need of controlling both separately. Lastly, the admittance model operates in only 1 DOF, i.e. opening and closing of the prosthetic hand.

More explicitly, the virtual model of the phantom hand captures the dynamics of the opening and closing of the missing hand by modelling the index finger as a pendulum as shown in Fig. 4. For simplicity, the interphalangeal joints are considered welded, the thumb is fixed and the pendulum has only 1 DOF at the MCP. Note that this scheme mirrors the actuation mechanics of the Michelangelo hand single-segment fingers [25]. The pendulum consists of a point mass attached to a massless rod, where motion occurs only in two dimensions, i.e. the mass point does not trace an ellipse but an arc. Its

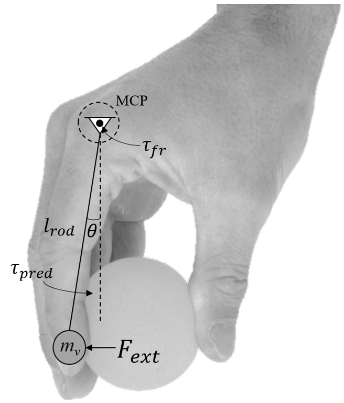


Fig. 4. Body diagram of the virtual dynamics of the index finger modelled as a 1 DOF pendulum. The massless rod, with length l_{rod}, is attached to a point with virtual mass (m_v) and to the MCP joint where friction between synovial surfaces generates a friction torque (τ_{fr}) opposing motion. The external force F_{ext} caused by the interaction with the environment opposes the predicted torque (τ_{pred}) by the MS model.

motion is affected by: an inertial torque, the predicted joint torque by the MS model for the hand open-close DOF (τ_{pred}), friction forces (τ_{fr}) in the attachment point (i.e. the MCP joint) and a external force (F_{ext}) acting on the point mass (i.e. the base of the index finger). Therefore, external forces are considered to act always in the same point and are transformed to torque by the length of the rod (i.e. length of the index finger l_{rod}). The gravitational component was not considered as this is affected by the rotation of the wrist, i.e. because motion only occurs in two dimensions, the gravity would only affect the motion of the finger when its base is facing up or down. This model was chosen for its simplicity and flexibility for future developments (see section IV-H).

The external force is measured from the force sensor embedded in the prosthesis in the base of the thumb and transformed by the length of the index finger, i.e. $\tau_{ext} = F_{ext} l_{rod}$. For accuracy reasons explained in section II-A, the embedded sensor was calibrated with an external load cell giving a range up to 74 N.

The friction force represents the friction present in a human joint: static friction must be overcome by muscle force in order to move; once moving, dynamic friction acts to oppose motion. Without friction, any minimal muscle activation would cause a joint torque and our limbs would constantly shake. Because the MS model does not consider joint friction this is specially an important factor: any registered EMG signal would compute small predicted torques causing the prosthetic hand to constantly move. Moreover, friction acts as a threshold for the external force too, meaning that for the external force to cause any effect on the motion of the pendulum it must exceed friction force first. The friction model used to mimic such behaviour is based on the Stribeck friction model, which is a combination of Coulomb and viscous friction models and where the Stribeck friction accounts for low velocity friction effect happening in lubricated environments, such as human synovial joints (e.g. MCP joint) [26], [27]. The parameters of the friction model were experimentally fine tuned.

Because the friction model is energy passive (for a definition of passivity refer to section II-G), we can consider the viscous friction coefficient as the damping coefficient of the MCP joint. Furthermore, we consider that the external forces do not affect the dynamics of the prosthetic device as it was shown to provide good disturbance rejection. Our admittance model can be obtained by solving the ordinary differential equation defining the motion of the pendulum. This is of the form of (1) in time domain and (2) in Laplace domain, with friction model (3), where: Y_v represents the (virtual) admittance, I_v is the virtual inertia (governed by virtual mass m_v), b_v is virtual damping, Ω is the resulting angular velocity and T input net torque, F_c the Coulomb friction, R is the contact radius between the metacarpal and proximal phalanx on the MCP joint, F_s is the Stribeck force, v_s is the Stribeck velocity and δ_{v_s} controls the decay of the exponential. The dependence of the variables on time (t) and s is omitted for readability:

$$I_v \dot{\omega} - b_v \omega = \tau = \tau_{pred} - \tau_{ext} - \tau_{fr} \tag{1}$$

$$Y_v = \frac{\Omega}{T} = \frac{1}{I_v s + b_v} \tag{2}$$

$$\tau_{fr} = \text{sgn}(\omega_d) R (F_c + (F_s - F_c) e^{(\frac{|\omega_d|}{v_s})^{\delta_{v_s}}}) \tag{3}$$

E. Prosthesis Low-Level controller

The output velocity reference of the admittance model is used to drive the HOC of the Michelangelo hand, where the velocity command for the fingers is mirrored for the actuation of the thumb. Control commands are amplitude-normalized for each subject. The prosthesis HOC angular kinematics are directly modulated as a function of the input command amplitude. The resulting HOC angle is fed into the EMG-driven model to update the kinematic-dependent state in the musculoskeletal model (Fig. 1). Lastly, the admittance model

TABLE I
MAPPING BETWEEN EXPERIMENTAL EMGS AND SIMULATED MTUS.

EMGs	Biceps brachii	Pronator teres	Extensor carpi radialis	Extensor carpi ulnaris	Extensor digitorum	Flexor carpi radialis	Flexor carpi ulnaris	Flexor digitorum
MTUs	BIClong, BICshort	PT, PQ	ECRL, ECRB	ECU	EDCI, EIP	FCR	FCU	FDSI,FDPI FDPM

MTU names: biceps brachii long head (BIClong) and short head (BICshort), extensor carpi radialis longus (ECRL), extensor carpi radialis brevis (ECRB), extensor carpi ulnaris (ECU), extensor digitorum communis indices (EDCI), extensor indicisproprius (EIP), flexor carpi radialis (FCR), flexor carpi ulnaris (FCU), flexor digitorum sublimis indices (FDSI), flexor digitorum profundus indices (FDPI), flexor digitorum profundus (FDPM), pronator quadratus (PQ), and pronator teres (PT).

is constrained to work only in the available range of motion of the prosthetic HOC. This prevents from computing commands that would drive the prosthesis out of its range of motion.

F. System Communication Framework

The real-time modelling framework (i.e. EMG-driven model and admittance model Fig.1) run on a PC (Intel Core i7-8700 at 3.20 GHz, 32 GB RAM). The admittance model and prosthetic hand control part were coded in MATLAB 2017b (The MathWorks, Inc., Natick, Massachusetts, United States). The admittance model was updated at 100 Hz. Two software plug-in modules were developed to enable direct UDP connection between the real-time modelling framework and Michelangelo hand. The first plug-in module allowed to connect the EMG-driven MS model with the admittance model and prosthetic control program. During prosthetic control, the control program updated the MS model with prosthetic DOF angles and EMG signal to receive newly computed torque estimates. The second plug-in module enabled a direct UDP connection to the prosthetic hand.

G. Theoretical stability assessment

Within the admittance control framework, when the prosthetic hand manipulates an object they both exert forces on each other and exchange mechanical power. In such situation, it can be said that they behave as a single coupled system. Coupling an admittance controlled device with external environment dynamics creates a negative force feedback loop (Fig. 5). Hence, the stability of the whole system will depend on the overall prosthesis dynamics and environment dynamics [21]. To ensure safe and robust prosthesis-environment interaction, the stability of the coupled dynamics needs to be assessed.

In our proposed framework, the overall prosthesis dynamics are specified by the admittance controller Y_v, prosthesis low-level control H_c and physical dynamics of the device (i.e. plant dynamics, H_p). The resulting dynamics are called the ***a**pparent dynamics* (Y_a) of the controlled device, where $Y_a = Y_v H_c H_p$. The **e**nvironment can be considered an impedance (Z_e) transforming displacement caused by the interaction with the prosthesis into a reaction force measured by the embedded force sensor in the prosthetic thumb (see Fig. 5). In more complete admittance models, environment dynamics also include the dynamics of the interface between the force sensor and the interaction port (usually called *post-sensor*) dynamics [21]. In this work, such dynamics are not considered for simplicity.

Because there is practically no knowledge of Z_e, one way of ensuring stability in our system is to ensure it is energy passive [21]. Our single-port system will be passive if the extracted power over time does not exceed the initial energy input to our system. For the shake of following conventional notation, we consider our system to be translational where the power is the product between force (f) and velocity (v) as in 4.

$$\int_{-\infty}^{t} f(\tau)v(\tau)d\tau \geq 0, \quad \forall t \geq 0 \tag{4}$$

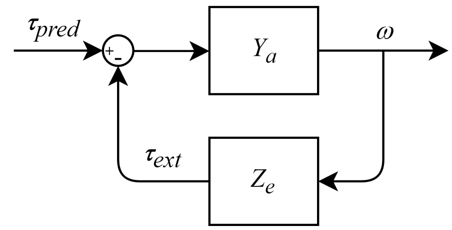


Fig. 5. Admittance and impedance causality of prosthesis-environment interaction (modified from [21]). Y_a represents the apparent dynamics of the prosthetic device and Z_e the unknown environment dynamics. Z_e transforms motion caused by interaction with the prosthesis into a reaction torque, τ_{ext}. Friction torque is removed for clarity.

Therefore, if we design the apparent dynamics Y_a to be passive, the system will be stable for any interaction with passive environment dynamics [28]. Springs, masses and dampers are passive elements, where physical objects can regarded as a combination of the formers. Therefore, theoretically, for passive Y_a dynamics, the prosthesis-environment interaction should always be passive. Passivity of the admittance controlled prosthetic hand can be assessed by observing the behaviour of Y_a. For this, in case the dynamics of the controlled device are not known (i.e. H_c and H_p) they must be identified by system identification techniques. Nevertheless, passive behaviour of a controlled device cannot always be achieved due to weak dynamic performance of the controlled device. Then, the apparent dynamics can be stable but non-passive [21]. In such case, we could assess what range of environment dynamics Z_e could complementarily stabilize Y_a for a stable prosthesis-environment interaction. Such range is called *environment z-width* or *ez-width*.

Environment dynamics can be described in dynamical parameters by its inertia, damping and stiffness. Due to the lack of knowledge of such parameters, we can model the environment as a stiffness (k_e) and a damping (b_e) such that $Z_e = b_e + k_e/s$. The ez-width is calculated based on known Y_v parameters, or better, can help in identifying appropriate I_v and b_v for fine tuning the admittance model for the purpose at hand. For our case, we consider I_v to be governed by a virtual mass m_v (i.e. inertia of a pendulum being $I_v = m_v l_{rod}^2$). Note that the values of m_v and b_v will be a trade-off between stability boundaries and performance, where increasing values of m_v would feel as moving a heavy finger and increasing b_v would slow down the response of the prosthetic hand due to its dissipative nature. The closed-loop transfer function of Fig. 5 is $Y_a/(1+Y_a Z_e)$, with loop-gain $Y_a Z_e$. The ez-width of Y_a can be calculated by determining the phase margin of the loop gain, that is: if the phase margin is negative for given values of k_e and b_e for a chosen values of m_v and b_v, the coupled system is unstable.

H. Experimental Tests

In this section, the experimental tests performed for the identification of prosthetic hand dynamics and stability assessment, and experimental tasks and outcome measures for the validation of the proposed control strategy are explained.

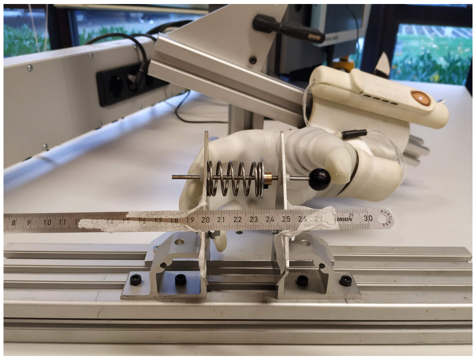


Fig. 6. Experimental set-up for force tracking and blind object tasks. The spring is fixed between two plates and moved by pressing the black ball meeting the thumb.

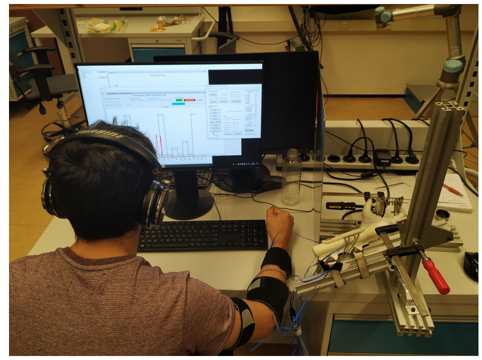


Fig. 7. Set up for blind object experiment. A wall is placed between the subject and the spring so no visual feedback is present during the experiment. Furthermore, the subject is deprived from auditory feedback by wearing noise-cancelling headphones.

Identification of prosthesis dynamics

To assess the ez-width of our admittance controlled prosthetic hand, we first conducted a system identification experiment to identify the unknown dynamics of the prosthesis controller and plant, H_c and H_p respectively. We conducted a short experiment where the prosthetic hand was excited by an input signal that commanded the prosthesis to a specific position. The output position was recorded and compared to the input. The model of the prosthesis dynamics to be identified is defined as a black box containing dynamics of H_c and H_p. A multisine was used as input signal because it is periodic, contains power only at the desired frequencies and allows for the detection of non-linear distortions to certain degree. To prevent leakage, frequencies equal to an integer multiple of the frequency resolution are included. This way each harmonic fits exactly an integer number of times in the multisine signal [29]. The multisine had a period of 120 s (sampled at 100 Hz) and contained 80 frequencies ranging from 0.05 Hz to 4.95 Hz. The variance of the signal was optimized with respect to its amplitude by crest optimization using MATLAB command `iddinput`. Three trials with three different amplitudes were carried out: 20%, 50% and 100% of allowed HOC. These measurements yielded the Frequency Response Function (FRF) for the three amplitudes.

Experimental tasks and outcome measures

Three intact-limb subjects (aged 24.3 ± 1.52) participated in this study and completed three experimental tasks. One subject had previous experience with MS-modelling and MYO control, but not with admittance based control. None of the rest had previous experience with MS-modelling or admittance based control. For ease of reading, the EMG-driven MS model based and admittance controlled prosthesis with environment force feedback is named MS+Adm+Fb, while the EMG-driven MS model with no admittance, i.e. the scheme presented by Sartori et al. in [13], is named MS+NoAdm. For the experimental tasks, only HOC was used as the Michelangelo hand could not grasp an object and pronate/supinate at the same time. This did not affect the goals of the experiments but limited the tasks to 1 DOF. The goals of the experimental tasks were fivefold:

- To evaluate to what extent MS+Adm+Fb control accounts for external forces when compared to MS+NoAdm when neither visual of auditory feedback are present.
- To assess if MS+Adm+Fb control scheme allows for more accurate and precise grasp force control when compared to MS+NoAdm.
- To assess the robustness of the control methods regarding undesired commands or unexpected commands produced by the EMG crosstalk between muscles.
- To determine which control strategy provided the most intuitive control when grasping new objects during un-trained conditions.
- To assess the computational performance of the frame-work for both control strategies.

The experimental procedures were carried out on two session on two consecutive days. In the first sessions, a musculoskeletal model was scaled and calibrated to match subject's anthropometry and force-generating capacity. Furthermore, the subject was trained to use both control methods and performed trials for the first two experimental tasks. The subject was also equipped with the Michelangelo and invited to grasp different objects. A proper training for the specific functional task was not given to the subject. The reason for this was to test the intuitiveness of both control methods when presented to new, untrained conditions and objects. During the second session, the subject-specific model was used for the online prosthesis control experimental tasks. Online control tests were performed with no model re-calibration.

1) Blind object: For this task, the subjects were seated in a chair that could be adjusted vertically in front of a computer screen where the experimental tasks were shown. The prosthetic hand was not worn by the subjects but fixed in a position ready to grasp a fixed object as shown in Fig. 6. A spring was fixed between the point where the index finger and the thumb of the prosthesis meet (i.e. placement of embedded force sensor), and both the prosthetic hand and the object were blinded to the subjects by setting a wall between subjects and prosthetic hand (see Fig. 7). Subjects wore noise-cancelling headphones to prevent auditory feedback when the object was grasped, as this feedback may prevent them from squeezing the object. The subjects were in control of the prosthesis HOC and asked to close it with varying levels of the maximal voluntary contraction (MVC). The measured MVC was plotted against the reference MVC levels in the computer screen as the only feedback to the subject. Subjects performed 3 trials with each control strategy (i.e. MS+Adm+Fb and MS+NoAdm). One trial consisted of 120 seconds partitioned in 24 MVC intervals of 5 seconds each. Four MVC levels were used as reference: 0%, 5%, 10%, 30%. The reference MVC levels were presented in a stair case form, where a 0% MVC was interleaved between each non-zero MVC levels. This fact helped to track the performance of the prosthesis when no command is sent but an external force is present. The displacement due to interaction forces was video-recorded for post-experiment analysis with the help of the ruler placed in the experimental setup (Fig. 6). This experiment allows to test the performance on grasp force control of the two control methods and to what extent they can account for external forces when no visual and auditory feedback of the prosthesis are given to the user. We hypothesized that MS+NoAdm control would cause higher interaction forces than during MS+Adm+Fb. Therefore, the outcome measure of this experiment were the displacement on the spring and the mean interaction force for each MVC level, where higher interaction forces were considered as possible damage or high deformation to the object or prosthesis.

2) Force tracking: For this task the set up was the same as for the previous one. The subjects were asked to grab the object by generating appropriate EMG commands to follow a reference force line shown in the computer screen as accurately as possible with varying interaction force levels. Subjects performed 3 trials with each control strategy (i.e. MS+Adm+Fb and MS+NoAdm). One trial consisted of 120 seconds partitioned in 12 intervals of 10 seconds each. Three interaction force reference levels were chosen: 5, 10 and 20 Newtons, which corresponded to 10 %, 20 % and 40% of the maximal force the subject could achieve, respectively. Force levels were presented in a stair case form, where each level was presented 4 times across the duration of the experiment (see Fig. 12). During the experiment, the interaction force between the object and the Michelangelo hand was measured and plotted in the same screen. The experiment was performed with MS+Adm+Fb and MS+NoAdm. The object used was a spring with specific stiffness k_e as discussed in section III-A. During MS+NoAdm control, the velocity commanded to the prosthetic hand is proportional to the input force, for which subjects would need to carefully produce reference velocities to achieve the reference interaction force. Because MS+Adm+Fb control can account for the interaction forces, we hypothesized that this control method would allow to reach faster balance between the subjects' force and the interaction force for a finer grasp force control. Therefore, the outcome measures for this experiment are the absolute mean error between the reference interaction force and measured interaction force, the variability of the measured interaction force and the goodness of fit between the reference interaction force and measured interaction force. The absolute error is taken as a measure of accuracy, while the variability of the measured forces is expressed as the interquartile range (IQR) and used to evaluate the consistency in the control of force (i.e., precision). The goodness of fit is evaluated in terms of R^2 value and gives information on how good the subjects could follow the reference force levels. The R^2 value, for each MS+Adm+Fb and MS+NoAdm, is computed as the combined mean R^2 value across all trials of each staircase period .

(a) (b)

Fig. 8. Setup for functional task. Subject wore the prosthetic hand and moved cups from the box on the table to a higher positioned box.

3) Functional task: For this task, the subjects were equipped with the Michelangelo hand and asked to grasp and move plastic cups from a box on a table to a higher positioned box (Fig. 8). Plastic cups were chosen as they are non-stiff objects and provide visual feedback when broken. The task was performed with the two control strategies for both untrained and trained conditions. Between conditions, the subjects were given several trials until he was familiarized with both control strategies and therefore trained to accomplish the task. The subjects were asked to transport 6 cups from one box to another 5 times non-stop, i.e. when all 6 cups were transported to the higher box they were transported back to the first box.

This experiment can show the usability of the MS+Adm+Fb in a real situation while assessing the robustness of both control methods when the position of the arm is changed during the task. The primary outcome measure of this experiment is the number of broken or dropped cups during manipulation and the secondary outcome is the number of transported cups per minute (cpm). We hypothesized that MS+Adm+Fb would allow for more intuitive control and faster manipulation of the cups while a safe interaction force is achieved, breaking and dropped less cups than with MS+NoAdm.

The computational performance was assessed in terms of mean and variance of the computational speed for all framework components across all trials.

The subjects performed the experiments in order of increasing complexity: first the blind object test (no direct force control), force tracking (fine force control) and finally the functional task (force control during real situation).

III. Results

In this section, we provide quantitative results for the experiments conducted to identify prosthetic dynamics, ez-width stability ranges and experimental tests.

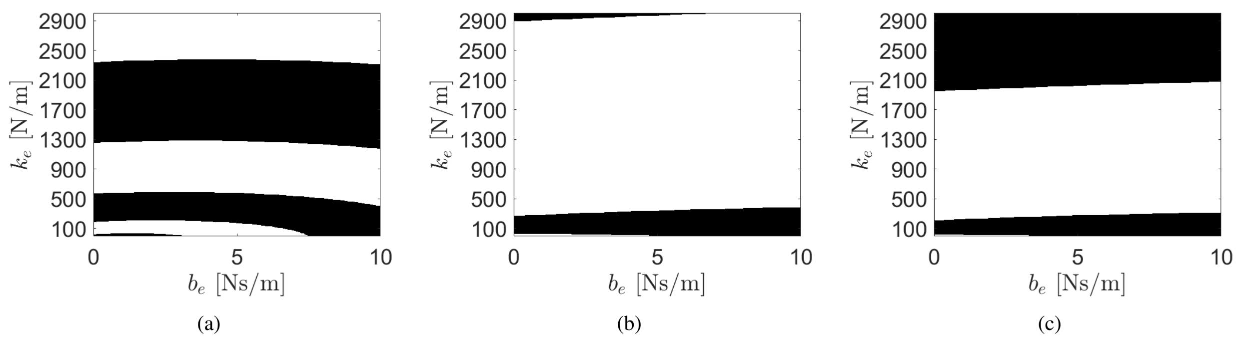


Fig. 9. Results for ez-width stability assessment for (a) theoretical stability range with realistic parameters; (b) experimentally stable parameters for experimental tasks; (c) parameters used during experimental tasks.

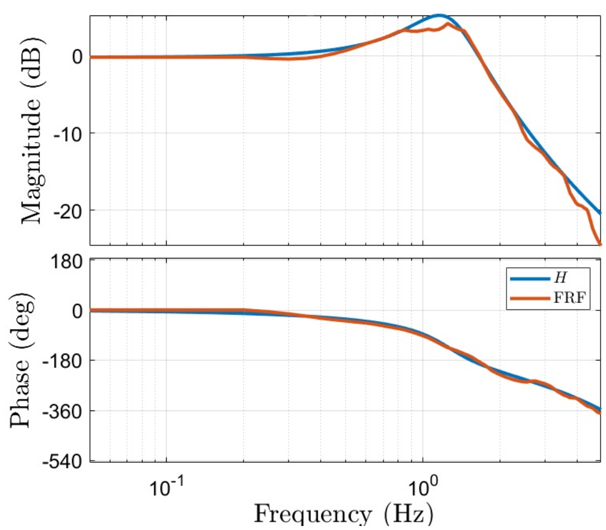


Fig. 10. Identified prosthesis dynamics via least-squares. The transfer function identified H can be found in (4), which corresponds to a second-order transfer function with a delay of 15 ms.

A. Identification of prosthesis dynamics

The resultant FRFs obtained from the system identification experiment showed a non-linear behaviour, as the FRF changed with the amplitude of the excitation signal. For the ease of the stability analysis at hand, the FRF with the highest cut-off frequency was selected (1.86 Hz) so as to preserve largest bandwidth of the prosthetic hand. Furthermore, the response was considered linear, where the non-linear behaviour of the prothesis dynamics is left for further research. The FRF was parametrized by a least-squares routine thus obtaining the transfer function (H, see (5)) for the prosthesis dynamics (Fig. 10). The analysis of the FRF showed a non-passive behaviour due to absolute phase lag not being $\leq 90°$, which marks the passivity constraint [30].

$$H = \frac{3.43s + 60.65}{s^2 + 4.828s + 61.92} \cdot e^{-0.15s} \tag{5}$$

Resulting apparent dynamics, Y_a, where found by multiplying the newly found transfer function H with Y_v. The stability of the system was assessed by coupling Y_a to Z_e for a range of values of k_e and b_e. First, for emulation of realistic values, m_v was set to 0.15 kg and $b_v = 0.5$ Nms/rad. With the resulting ez-width, the theoretical stiffness of the object used for the experimental tasks could be selected to ensure stability of the interaction. However, due to the short stability range provided by such configuration during real experiments and available springs, m_v and b_v values needed to be adjusted to unrealistic finger mass value of $m_v = 4$ kg while b_v was increased up to 7 Nms/rad. With this, the spring stiffness used for our experiments was of 2400 N/m, well within the stable range (Fig. 9(b)). Nevertheless, during experimental tasks, due to subject preferences m_v was decreased to 2 kg and b_v to 5 Nms/rad, which resulted on a faster reaching compared to previous values, although stability range was decreased (see section IV for further explanation).

B. Experimental Tasks

The proposed MS model converted EMG signals into torques produced by 14 MTUs that were used as input to control the hand opening and closing, or HOC, of the prosthetic hand. This allowed to control the HOC of the device with the same human natural hand open-close movement. The calibration of the subject-specific EMG-driven MS-model was performed on a different day prior to experimental session. This provides evidence of the ability of the EMG-driven MS model of maintaining subject-specific parameters across time, which confirms the results obtained by Sartori et al. [13]. Furthermore, this section presents the results obtained during the experimental task presented in section II-H. In the reminder of this section, the intact-limb subjects are referred as SUBJ1, SUBJ2 and SUBJ3.

Blind Object

During the blind object task, subjects followed reference values of MVC. The resulting spring displacement and mean interaction forces were measured. In Fig. 11(a)-(c) the results for the displacement are displayed, where it can already be seen a more linear relationship between MVC (i.e. force production) and displacement for MS+Adm+Fb than for MS+NoAdm, specially for SUBJ2 in Fig. 11(b). This gives the hint that interaction forces increase with increasing MVC, while this relationship is less notable for MS+NoAdm.

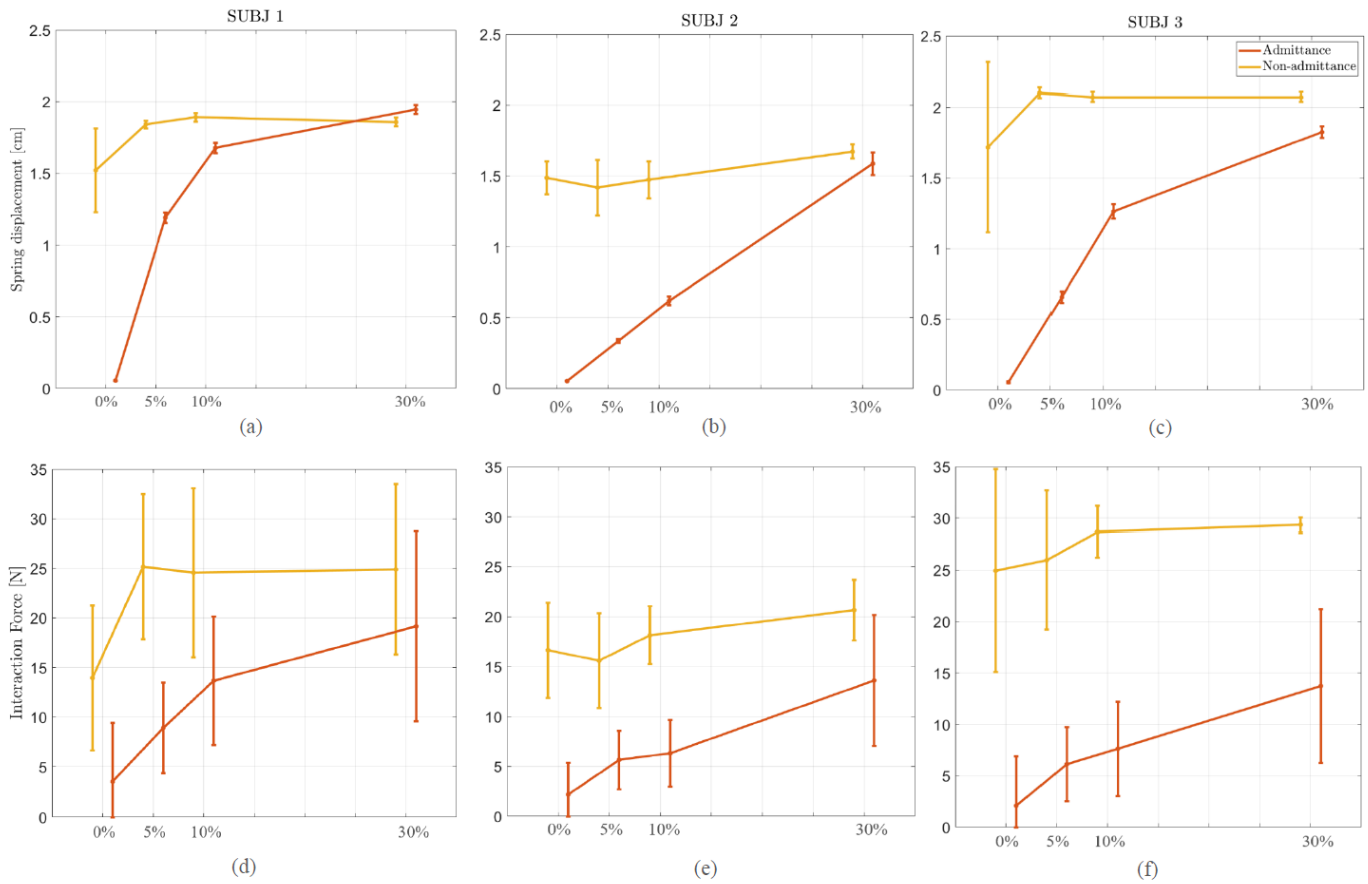


Fig. 11. (a)-(c): mean (ball marker) and standard deviation (vertical lines) of the displacement caused in the spring at different MVC levels for each subject; (d)-(f) mean and standard deviation of interaction forces.

This is confirmed by looking at Fig. 11(d)-(f), where an increasing MVC level produces higher interaction force in MS+Adm+Fb, while this is not the case during MS+NoAdm. During MS+Adm+Fb control, intermittent oscillation were spotted due to minimal torques generated by minimal EMG activations although friction was present. This caused small displacements in the spring as shown in 0% MVC level in Fig. 11. Moreover, the interaction force variance of 0% MVC level in Fig. 11(d)-(f) was mainly caused by subjects' anticipation to the consecutive force level (5% MVC).

Force Tracking

During force tracking experiment, subjects followed a reference force line. A representative result of the measured force profiles for MS+Adm+Fb and MS+NoAdm, obtained from SUBJ1, is shown in Fig. 12. Here is possible to see that during MS+Adm+Fb the subject achieved a smoother profile compared to MS+NoAdm. The absolute error between reference force and measured force is shown in Table II. For all three force reference levels the absolute error of MS+Adm+Fb was smaller compared to MS+NoAdm, meaning that MS+Adm+Fb was more accurate. However, SUBJ2 was slightly less accurate during MS+Adm+Fb compared to MS+NoAdm when tracking mid forces (10 N) and high standard deviation at low forces (5 N). The results for variability in terms of IQR are presented in Fig. 13. MS+Adm+Fb was more precise during all force levels when compared to MS+NoAdm for SUBJ1 and SUBJ3. However, SUBJ2 was less precise during MS+Adm+Fb compared to MS+NoAdm when tracking mid forces, where MS+NoAdm had high accuracy for mid forces. Overall, subjects were more accurate and precise with MS+Adm+Fb than with MS+NoAdm. The amount of outliers in Fig. 13 is due to interaction forces happening outside the reference force level limits (see Fig. 12), i.e the outliers of reference level of 5 N are mainly due the overlap with the consecutive reference force level of 20 N. Regarding the R^2 values presented in Fig. 14, during MS+Adm+Fb all subjects were able to track the reference force better than during MS+NoAdm, which validates the results obtained for accuracy and precision.

Functional Task

Regarding the functional task, the subjects were asked to grasp and move 6 plastic cups from a box to a higher position 5 times (total number of transported cups = 30). The results are presented in Tables III and IV, and Fig. 15. Overall, during both untrained and trained conditions, subjects crushed and dropped less cups with MS+Adm+Fb than with MS+NoAdm. Furthermore, during trained condition

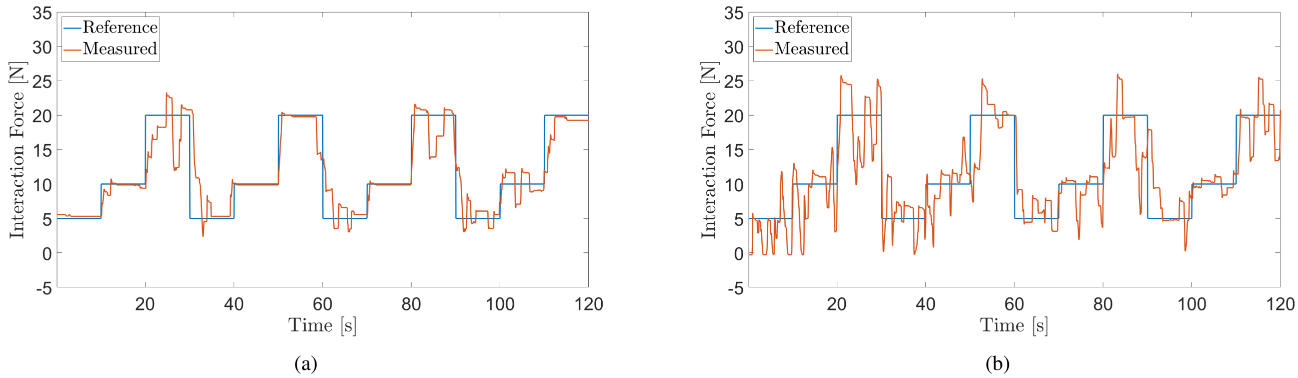


Fig. 12. A representative result from a subject performing force tracking experiment with (a) admittance based control (b) non-admittance based control

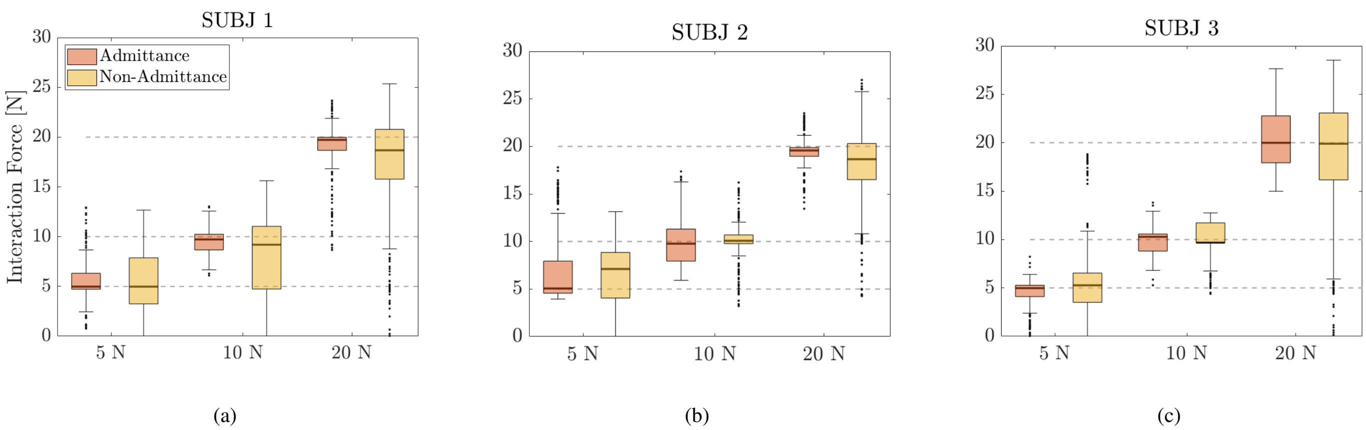


Fig. 13. Interquartile ranges for MS+Adm+Fb and MS+NoAdm for reference force levels 5 N, 10 N and 20 N. Dashed gray lines correspond to target reference force levels.

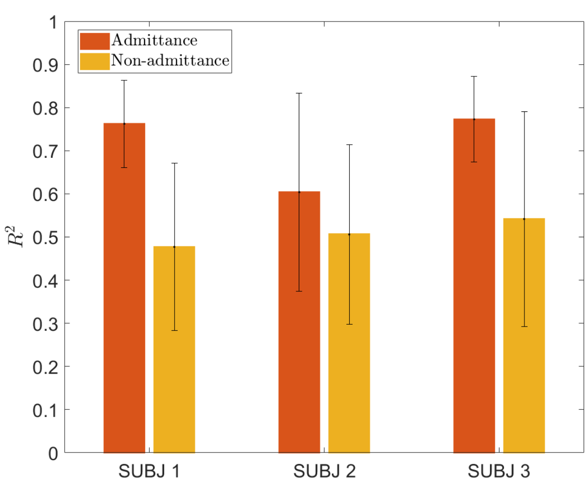


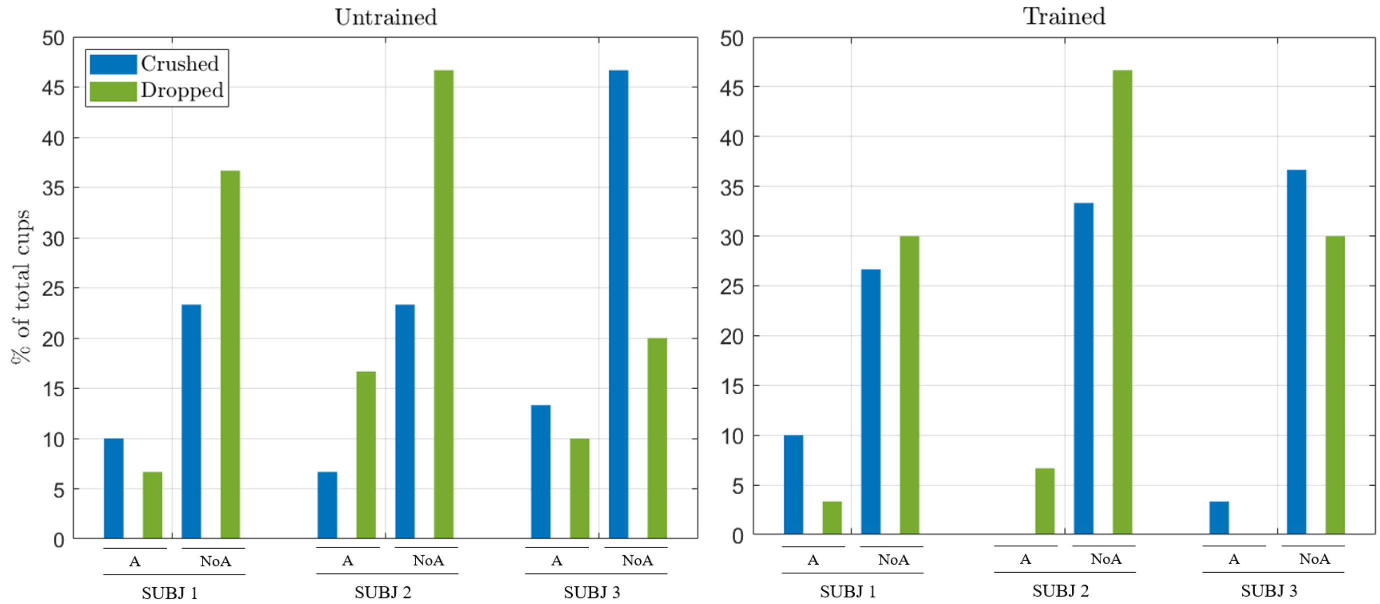
Fig. 14. R^2 values during admittance and non-admittance based control for each subject.

TABLE II
ABSOLUTE ERROR (ε) BETWEEN MEASURED FORCE AND REFERENCE FORCE LEVELS OF 5, 10 AND 20 N.

		5 N	10 N	20 N
SUBJ1	ε_A	1.63±2.19	1.02±1.01	2.25±2.91
	ε_{NoA}	2.95±2.71	3.44±3.17	3.84±3.90
SUBJ2	ε_A	3.17±4.09	1.90±1.47	1.42±2
	ε_{NoA}	3.62±2.87	1.18±1.41	3.36±2.97
SUBJ3	ε_A	0.98±1.86	1.25±1.15	2.85±2.28
	ε_{NoA}	3.52±3.75	1.70±2.07	4.47±4.11

A = admittance based control, NoA = non-admittance based control

the number of crushed and dropped cups was less than during untrained conditions for both MS+Adm+Fb and MS+NoAdm. Regarding the transported cups per minute (cpm) in Fig. 15, generally subjects were faster during trained condition than untrained. However, each subject exhibited a different behaviour and approach to the task during trained condition which affected their own cpm scores, as discuss in section IV-E. Furthermore, all subjects reported MS+Adm+Fb to be more intuitive during untrained conditions and stated they were more confident using the prosthetic hand. Further results are given as video recordings. Lastly, an additional video was recorded while SUBJ1 handled one plastic cup with his/her hand and another one with the prosthetic hand. SUBJ1 was able to squeeze both cups at the same time without breaking any of them, although the subject reported to make higher



(a) (b)

Fig. 15. Crushed and dropped plastic cups during (a) untrained and (b) trained conditions. A = admittance based control, NoA = non-admittance based control

TABLE III
FUNCTIONAL TASK SCORES DURING ADMITTANCE BASED CONTROL.

	Untrained			Trained		
	C	D	T	C	D	T
SUBJ1	3	2	5	3	1	4
SUBJ2	2	5	7	0	2	2
SUBJ3	4	3	7	1	0	1
Total	9	10	19	4	3	7

C = crushed cups, D = dropped cups, T = total faults.

TABLE IV
FUNCTIONAL TASK SCORES DURING NON-ADMITTANCE CONTROL.

	Untrained			Trained		
	C	D	T	C	D	T
SUBJ1	11	11	22	8	9	17
SUBJ2	7	14	21	10	11	22
SUBJ3	14	6	20	11	9	20
Total	32	31	63	39	28	57

C = crushed cups, D = dropped cups, T = total faults.

forces to accomplish the same movement in the prosthetic device.

Computational Performance

The framework generated prosthesis commands at average speeds of 49 ± 38 ms, where 50% of the commands were generated within 16 and 56 ms. These numbers are higher than the ones reported by Sartori et al. [13], although the framework used in that work did not make use of neither UDP connection nor admittance control implemented in MATLAB.

IV. Discussion

A. Main Results

In this study, a subject-specific EMG-driven musculoskeletal model coupled to an admittance model of the hand opening and closing dynamics was used to control the grasping forces of a prosthetic hand. EMG-driven MS-model predicted joint torques were translated into prosthetic commands with no need for explicit conversion into position commands. The

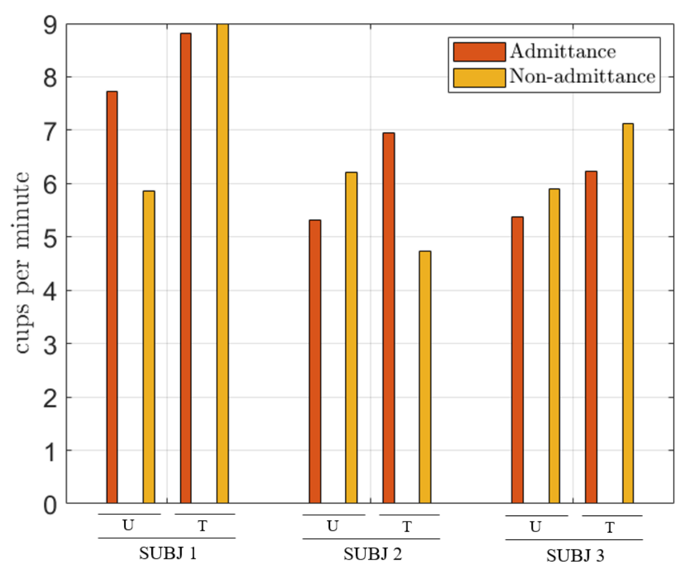


Fig. 16. Cups per minute transported during untrained (U) and trained (T) conditions.

proposed control strategy was evaluated in terms of stability and compared to non-admittance based EMG-driven musculoskeletal modelling control in terms of performance, robustness and intuitiveness. Therefore, this study provides a-proof-of-concept control framework that is subject-specific and that can account, to some extent, for external forces thus readjusting control commands sent to the prosthesis.

Results showed the ability of natural control for hand open-close movement in both control strategies tested. Moreover, during admittance based control, the external forces arising from interaction with the environment were taken into account and the control strategy readjusted prosthetic commands to enable finer grasp force control when compared to non-admittance based control. This shows the possibility for further refinement of the admittance and friction models employed in this work for better future results.

B. *Stability and system identification results*

The results obtained for the stability range provide a theoretical result of the ez-width for different configurations of the admittance parameters. In general, the ez-widths obtained are not uniform which does not provide clear view of the stability ranges. This means that if the ranges for k_e and b_e are further increased to what is shown in Fig. 9 similar encirclements can be seen. Such shape of the ez-widths obtained is due to the delay in the dynamics of the prosthesis, where higher delay would make the system less stable, apart from making it less intuitive for the subject. The unclear stability ranges can also be explained by the non-linear behaviour of the prosthesis. Since the analyses presented in this paper assume the linearity of the identified prosthesis dynamics, the stability results should be taken as an approximation of the stability boundaries of the real non-linear dynamics. Moreover, post-force sensor dynamics were not analysed in this work. The knowledge of such dynamics could also help to gain insight on the true stable regions of the system. Nevertheless, the ez-widths obtained for the different parameters complied with the stability rules. Higher damping and mass contributed to increased stability ranges when compared in the same range of k_e and b_e. This is specially noticeable looking at Fig. 9(b) and 9(c).

The imprecise theoretical stability regions lead to different admittance configurations during experimental tasks. The realistic case where a small mass of the finger is considered resulted in very unstable behaviour during experimental tasks, regardless of the stiffness of the objects grasped. This complies with the known difficulties of admittance control in rendering low inertias, i.e. for a small inertia, a minimal force would cause a high sudden acceleration. [21]. The spring used in the experiments was experimentally chosen for a specific, non-realistic admittance model parameter configuration (m_v = 4 kg, b_v = 7 Nms/rad). However, during experimental tasks, several parameters were tried by discussion with each subject where each of them reported different preferences. SUBJ1 and SUBJ3 reported a preference of m_v = 2 kg, b_v = 4 Nms/rad, while SUBJ2 reported a preference for m_v = 4 kg, b_v = 10 Nms/rad. SUBJ1 and SUBJ3 preferred faster performance over additional stability, while SUBJ2 opted for additional stability over faster performance. Subject performance preferences influenced the results obtained during the experimental tasks, as discussed later. This highlights the fact that the admittance model needs to be adjusted to subject preferences and/or physical abilities, such as force generation. Nevertheless, as stated above, the stability boundaries provided are theoretical and the real stability ranges may be different from the ones reported. This means that if an object stiffness is theoretically passive, this case could not be the same during real object grasping, or the vice-versa. However, an analytical stability assessment could yield finer results rather than the numerical approach used in this work.

Additionally, the sampling rate of the whole system plays an important role during experimental conditions. The control framework was implemented in different platforms, i.e., the EMG-driven MS-model, admittance controller and Michelangelo hand control interface operated in parallel in different platforms. Regarding the variance of the computational times registered, the update rate of the admittance model is highly variable on time. Reduced or varying update rates can directly affect the performance and stability of the admittance model. Therefore, if a more complex admittance model needs to be implement in the future, the complexity of the overall system communication framework should be simplified to work in one platform.

C. *Blind object task results*

Regarding Fig. 11, it can be seen that the results obtained by different subjects were comparable, disregarding the force generating capabilities and admittance parameter configuration of each individual. For MS+Adm+Fb control it can be seen a proportional relationship between activation and interaction force, and interaction force and displacement. The difference between subjects' curves for MS+Adm+Fb can be explained by individual force generating capabilities and admittance parameter configuration. Interestingly, SUBJ2 achieved a linear relationship between activation and spring displacement. This could suggest that a proper admittance tuning can lead to better muscle activation-interaction force relationship. Although hardly noticeable in Fig. 11, there was an intermittent oscillatory behaviour during MS+Adm+Fb control when no activation was present (0% MVC). This was due to a possible unstable interaction, where the balance between predicted torque, friction and external torque did not reach a steady state. However, due to the constant amplitude of the oscillations it could be said that the interaction was *marginally* stable. To counteract this effect, the Coulomb and Stribeck force could be increased, although the subject would have to overcome this higher force to move the prosthetic hand. Moreover, the mass could be increased, although a higher virtual mass would need of higher predicted torques generated by the user to be accelerated. In this sense, SUBJ1 and SUBJ3 reported the preference of a lower virtual mass value, while SUBJ2 opted for an increased virtual mass.

Furthermore, from Fig. 11 it can be interpreted that during MS+Adm+Fb, when no activation was present, the prosthetic hand opened due to the interaction force with the spring, while for MS+NoAdm this is not the case. The reason for this is that the prosthesis itself is non-backdrivable. During non-admittance control, this fact allows the user to relax while the prosthetic hand holds the force reached. This may seem as a benefit during object manipulation, as the user is freed from maintaining the muscle contraction to hold the prescribed force. However, when the force needs to be increased, the user needs to activate the muscles from a resting state up to the point where the control signal is higher than the level corresponding of the current grasping force. This is the reason why in Fig. 11(d)-(f) MS+NoAdm does not show and increasing interaction force. In admittance based control, this behaviour is avoided by the opposing force of the environment accounted by the admittance model, where the prosthetic hand opens automatically (according to the dynamics of the admittance model and the object) when forces are exerted on the force sensor embedded in the thumb. Therefore, MS+Adm+Fb control frees the user from constant visual and auditory monitoring of the grasp forces exerted on the manipulated object. This can be beneficial for amputee, where the absence of afferent feedback pathways invalidated their internal model of the grasp forces for known objects. Because of this fact, the benefit of admittance-based control could be remarkable during adaptation phases to the prosthetic device.

D. *Force tracking tasks results*

The force tracking task provided insight on the accuracy and precision of the control strategies employed. By looking at the force profiles in Fig. 12 it can be seen that MS+Adm+Fb provided more accurate and precise results. This is further confirmed by the absolute error where the standard deviation of mid forces is specially low. Moreover, looking at Fig. 13 it can be seen that during MS+Adm+Fb the medians of all subjects are almost exactly at reference force levels, which is not the general case for MS+NoAdm. However, it is interesting to note that SUBJ2 achieved better accuracy results for mid forces (10 N) during MS+NoAdm compared to MS+NoAdm and SUBJ3 achieved good results with MS+Adm+Fb also for mid forces. The reason for this could be that, during MS+NoAdm, they used Michelangelo hand's non-backdrivable feature to slowly increment the interaction force until 10 N were reached. An example of such behaviour can be spotted in SUBJ1 too in Fig. 12(b) at 10 N in the forth period. However, regarding the whole trial of MS+NoAdm, SUBJ1 opted for opening/closing the hand on a trial-and-error fashion which lead to a bigger IQR at 10 N compared to the other subjects. Regarding this effect, the higher force step from 10 N to 20 N and the sudden fall from 20 N to 5 N prevented subjects from increasing or decreasing force in small steps and achieving such effect at low and high forces. For high forces (20 N), the variance is maximal for both control strategies, which is in line with the results of the research community [31]. However, admittance-based control registered a smaller variance than during non-admittance based control, meaning MS+Adm+Fb was still more precise at high interaction force reference values. The high variance in the IQR (Fig. 13) of the high force was due to neighbouring force reference values. From Fig. 12 it can be seen that the subject anticipated to the next force reference level, which affected the results in both control strategies. More interestingly, the erratic behaviour during MS+NoAdm is in line with other force tracking experiments results performed by the research community [32]. As explained in the work of Dosen et al. [32], when no kind of feedback is present, the user is not aware of the control queues sent to the prosthetic hand, which makes its control rather unpredictable. This way, a sudden increase in the interaction force comes by surprise. In this point, the user needs to open the hand to generate less grip force which results on a sudden drop in the interaction force. For finer force control, the user tends to slowly open the hand to accomplish small force decrements which often lead to excessing hand opening and losing contact with the object. Such overall behaviour leads to a difficult control on force increments and decrements. This behaviour lead to poorer R^2 values during MS+NoAdm when compared to MS+Adm+Fb, which shows that all subjects were better at following the reference force line with MS+Adm+Fb. Overall, admittance-based control outperformed non-admittance-based control in terms of accuracy, precision and goodness of fit.

E. *Functional results*

During functional tasks, the validity of both control schemes regarding ADLs was assessed. The limitations of MS+NoAdm control explained above were more present during this experimental tasks.

With respect to the results obtained for cup crushes, drops and transported cups per minute (cpm), there are four general conclusions: 1) subjects crushed and dropped more cups during MS+NoAdm than MS+Adm+Fb, 2) subjects improved their crush and drop results from untrained to trained condition, for both MS+Adm+Fb and MS+NoAdm, 3) in general, subjects transported cups faster when trained, that is, cpm results incremented for MS+Adm+Fb and MS+NoAdm and 4) subjects transported cups faster during MS+NoAdm, both during untrained and trained conditions.

Regarding the first point more in detail, the subjects crushed more cups with MS+NoAdm trying to reach a point for force balance between prosthetic force and external force by opening and closing their hand. The reason for this is that, during MS+NoAdm, when the cup was already grasped, subjects tended to relax the muscles hoping the prosthesis would keep the force due to its non-backdrivability. This produced minimal activations that the EMG-driven MS-model transformed into minimal torques commanded to the prosthesis, which made the hand vary its position rapidly thus losing contact with the cup. The result was the same during force tracking task: due to sudden prothesis position changes subjects did not manage to increment and decrease the opening of the hand to reach a safe balance. In MS+Adm+Fb, this effect is counteracted during admittance control by virtual mass and damping and specially friction model: minimal torques need to exceed stiction forces (i.e. Coulomb friction forces) before they produce any motion. From this, it can be said that the inclusion of a virtual friction model in the control framework was helpful to control prosthetic motions produced by undesired torques from minimal EMG. Overall, this suggests that MS+Adm+Fb control is more robust than MS+NoAdm during functional activities. Nevertheless, as commented above, the performance of the MS+Adm+Fb can vary with increased object stiffness which results in a less stable interaction.

Points two to four can be analysed together. To start with, during MS+Adm+Fb, SUBJ2 and SUBJ3 improved their crush and drop results from untrained to trained while achieving higher cpm in trained condition. However, SUBJ1 achieved the highest cpm scores, but did not improve its results as significantly as SUBJ2 and SUBJ3 from untrained to trained. Therefore, this suggests a trade-off between performance speed, interaction accuracy (crushes and drops) and learning patterns: interaction accuracy is limited by interaction speed, and interaction speed improves through learning. Regarding such trade-off, MS+Adm+Fb showed better results than MS+NoAdm during untrained conditions and improved its results in trained conditions (i.e. more than a half less crushes and drops compared to untrained), which was not the case for MS+NoAdm. In conclusion, this meant that MS+Adm+Fb provided a more intuitive grasp force control and allowed for quicker learning. However, due to the reduced sample size, these results cannot be generalized and a more extensive study should be done in the future.

It is also worth noting the difference in the configuration of the admittance parameters across subjects. In theory, higher virtual mass and damping would mean more stability, then less crushes but lower cpm. Nevertheless, the cpm scores achieved by SUBJ2 were comparable to the ones of SUBJ3, who opted for admittance parameters that provided faster performance over stability. Even more, during trained condition, SUBJ2 had better cpm than SUBJ3 while having comparable total faults (crushed+drops). This can suggest that a slower prosthetic hand did not stop SUBJ2 from transporting cups faster than SUBJ3. Besides providing a final conclusion, how different admittance parameters affect overall performance must be researched with different groups and configurations in future work.

In conclusion, the results obtained during the functional task are in line with our hypothesis: regarding the results of untrained and trained conditions, subjects crushed and droped less cups during MS+Adm+Fb than MS+NoAdm, which, along with the overall opinion of the subjects, shows that admittance-based control provided a more intuitive and safer interface for grasp force control.

F. Computational performance results

The computational speed of the non-admittance-based control and the admittance-based control framework was the same as the one reported in the results section, where the added computational complexity of the admittance model does not seem to have a big impact in the overall performance. However, the whole system is very sensitive to the overall running frequency. This means that if commands are received from the Michelangelo hand at 100 Hz, there is a delay in processing this information both from the MATLAB control script and the joint torque computation on the MS model. This could explain the high variance of the time taken to generate prosthetic commands. The performance of the whole framework could be increased if this was running in one lower level control platform (e.g. C++).

G. Limitations

During experimental task, several issues were encountered. The maximal force for closing movement of the index finger predicted by the model was found to be lower than measured values registered by the research community, which are in the range of 50-60 N [33]–[35]. The maximal external force the prosthetic hand can register is 74 N, therefore, for stable interaction, the maximal external force can never be higher than the maximal predicted torque during maximal muscle activation. To solve this problem, the predicted torque was artificially increased by a gain to match normal index finger force values. Furthermore, the range of the external force was decreased until a comfortable balance was reached, i.e. if a maximal activation (100% of MVC) meant a predicted force of 60 N, the maximal external force was lowered to 30 N to comply with more achievable values of MVC levels employed during experimental tasks. Note that this is equivalent to artificially reducing the stiffness of the spring to almost half of it, because for the same displacement, the registered force is a percentage of the maximum force the spring can deliver. Considering this fact, the spring used in the experiments would be within the stable regions of Fig. 9(c).

Furthermore, the Michelangelo hand device available for experimentation had two problematic behaviours that limited the performance of the control strategy. First, the prosthetic hand would return to a programmed 'natural position' if the open/close position of the hand was more than 50%. This was programmatically solved although such behaviour kept sporadically happening. Secondly, when the prosthetic hand open/closed while pronating or supinating, for forces higher than 11% of the maximal force the rotation of the hand was blocked. This prevented from using simultaneous control of both DOFs.

Lastly, for the calibration of the embedded force sensor on the Michelangelo hand, a 1 DOF load cell was used. Due to the specific location and orientation of the embedded force sensor, it was challenging to position the load cell such that the force was measured in only one component. If more accurate calibration is intended, the use of a force/torque load cell is recommended to calculate force components in more than 1 DOF.

H. Future work

Apart from the already mentioned recommendations, several points should be considered regarding future work.

1) Improvement of admittance model: The admittance model could be improved in many ways. The simple model used in this paper works as a proof-of-concept. For future work, it could be interesting to use the real equations of motion of the index finger. Note that the Michelangelo hand does not allow for finger phalanx flexion, which is worth to take into account when driving the real prosthesis. Furthermore, human tendons of finger flexors and extensor cross several joints thus reducing the mass and inertia of the finger. This allows for fine and rapid movements. This way, the dynamics of the fingers are mainly controlled by damping and stiffness values [36]. Another point to take into account is the effect of gravity. Depending on the dynamics of the prosthetic hand and its weight, it could be interesting to compensate the torque generated by the gravity on the prosthetic fingers for finer movements.

Furthermore, the parameters of the admittance model should be chosen regarding stability but also the preferences of the subject. Regarding this fact, it could be interesting to find ways of fine-tuning the admittance for each subject given specific physiological variables and stability boundaries. Moreover, this could lead to a *variable* admittance controller, where admittance parameters are adjusted in time given the task at hand and the intention of the user, the same way as humans regulate limb dynamics for safe interaction [36]. For example, the virtual damping could vary in time: with high damping, large muscle activation and long time are required by the user, but high accuracy movements are performed instead; with low damping, smaller muscle activations and short time are required by the user, although less accurate prosthetic motion is achieved. This could help to find a better trade-off between contact stability and faster performance during free movement in comparison with fixed admittance parameter values.

2) Admittance for 2 DOFs: In this work admittance was implemented only in 1 DOF, i.e. hand open/close. However, the pronation and supination of the hand affect the way we hold objects and the muscles involved in it. To account for how external forces accelerate the pronation and supination of the wrist, an adequate admittance model that takes into account this forces should be implemented. For this specific example, the dynamics of the Michelangelo for pronating and supinating should be identified too. However, in robotic devices with multiple DOFs, energetic coupling between non-linear DOFs could result in instability effects which are not present in single-DOF stability analyses [21].

3) Investigate prosthesis dynamics: To further complete the identification of prosthesis dynamics, it would be interesting to investigate the non-linear behaviour of the Michelangelo hand by a more profound analysis. This way, finer stability regions could be found. Moreover, post-sensor dynamics could also help to elucidate stable performance of the prosthetic device.

4) Inclusion of afferent feedback: Regarding the way humans control their movements, the inclusion of afferent feedback could complete and complement the current admittance-based control framework. The inclusion of afferent feedback pathways from muscles spindles and Golgi tendon organs in the proposed control strategy could be done in two ports: 1) in the admittance model and/or 2) in the NMS model. In the first case, the output virtual velocity and position could be used as input to a feedback block containing the dynamics of muscle spindles. In the second case, the MS model could be enhanced to include the dynamics of muscle spindles and Golgi tendon organs. In such scheme, either virtual or measured velocity and position could be feed-backed to muscle spindle dynamics while the output force of the MS could be feed-backed through the dynamics of the Golgi tendon organs. This would place most of the control in the side of the prosthetic controller. Therefore, this scheme could be coupled to a haptic feedback framework that informs the user of interaction parameters, i.e. interaction force, limb position etc.

I. *Considerations for framework generalization*

The identified dynamics, stability assessment and admittance model were found and tuned for the Michelangelo hand by Ottobock. Nevertheless, the same process can be followed for particular hand dynamics in the likes of [37], where the specific dynamics of the prosthetic hand are explicitly modelled and parameters are found by system identification techniques. This way, this work provides a possible generalised control framework, that can be extended to other prosthetic models which could even include finger phalanx motion. In this, the EMG-driven MS-modelling can predict torques based on subject-specific parameters that can be used in conjunction with an admittance model suitable for specific prosthetic hand dynamics that accounts for external forces. Note that and embedded or couple force sensor is needed for this framework, or force estimation through motor current such as in [38].

V. Conclusion

In this work, a subject-specific EMG-driven musculoskeletal model coupled to an admittance and friction model that captures, to some extent, the natural grasp dynamics was used to control a prosthetic hand. The proposed control strategy accounted for external forces arising from the interaction with the environment. This provided an intuitive control framework that closer matches human natural control by giving the user the ability to control the force exerted on the environment. The framework was shown to be more accurate and precise than non-admittance-based EMG-driven MS modelling controlled prosthetic hand. It also shows the opportunities for exploiting the benefits of both model-based control and admittance control to work towards a better bio-mimetic control.

Acknowledgement

The author is very thankful to dr. Massimo Sartori, dr. Arvid QL Keemink and Guillaume Durandau for supervising this master assignment. Thank you dr. Utku Yavuz for providing external supervision for this work. Furthermore, many thanks to Tom Baumeister, Eileen Frerk, Carlos Fambuena, Miguel Bhagubai, Wouter Abbas and Nikolai A. Melikian.

References

[1] K. Demet, N. Martinet, F. Guillemin, J. Paysant, and J.-M. Andr, "Health related quality of life and related factors in 539 persons with amputation of upper and lower limb," *Disability and Rehabilitation*, vol. 25, no. 9, pp. 480–486, Jan. 2003, ISSN: 0963-8288, 1464-5165. DOI: 10.1080/0963828031000090434. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/0963828031000090434 (visited on 09/20/2019).

[2] C. Castellini, P. Artemiadis, M. Wininger, A. Ajoudani, M. Alimusaj, A. Bicchi, B. Caputo, W. Craelius, S. Dosen, K. Englehart, D. Farina, A. Gijsberts, S. B. Godfrey, L. Hargrove, M. Ison, T. Kuiken, M. Markovi, P. M. Pilarski, R. Rupp, and E. Scheme, "Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography," *Frontiers in Neurorobotics*, vol. 8, Aug. 15, 2014, ISSN: 1662-5218. DOI: 10.3389/fnbot.2014.00022. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnbot.2014.00022/abstract (visited on 09/25/2019).

[3] F. Cordella, A. L. Ciancio, R. Sacchetti, A. Davalli, A. G. Cutti, E. Guglielmelli, and L. Zollo, "Literature review on needs of upper limb prosthesis users," *Frontiers in Neuroscience*, vol. 10, May 12, 2016, ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00209. [Online]. Available: http://journal.frontiersin.org/Article/10.3389/fnins.2016.00209/abstract (visited on 09/20/2019).

[4] B. Peerdeman, D. Boere, H. Witteveen, R. Huis in 'tVeld, H. Hermens, S. Stramigioli, H. Rietman, P. Veltink, and S. Misra, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *The Journal of Rehabilitation Research and Development*, vol. 48, no. 6, p. 719, 2011, ISSN: 0748-7711. DOI: 10.1682/JRRD.2010.08.0161. [Online]. Available: http://www.rehab.research.va.gov/jour/11/486/pdf/peerdeman486.pdf (visited on 11/06/2019).

[5] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: A survey of the last 25 years," *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236–257, Sep. 2007, ISSN: 0309-3646, 1746-1553. DOI: 10.1080/03093640600994581. [Online]. Available: http://journals.sagepub.com/doi/10.1080/03093640600994581 (visited on 09/22/2019).

[6] M. Sartori, D. G. Llyod, and D. Farina, "Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 5, pp. 879–893, May 2016, ISSN: 0018-9294, 1558-2531. DOI: 10.1109/TBME.2016.2538296. [Online]. Available: https://ieeexplore.ieee.org/document/7440844/ (visited on 10/03/2019).

[7] C. Antfolk, M. DAlonzo, B. Rosn, G. Lundborg, F. Sebelius, and C. Cipriani, "Sensory feedback in upper limb prosthetics," *Expert Review of Medical Devices*, vol. 10, no. 1, pp. 45–54, Jan. 2013, ISSN: 1743-4440, 1745-2422. DOI: 10.1586/erd.12.68. [Online]. Available: http://www.tandfonline.com/doi/full/10.1586/erd.12.68 (visited on 03/01/2019).

[8] I. Saunders and S. Vijayakumar, "The role of feed-forward and feedback processes for closed-loop prosthesis control," *Journal of NeuroEngineering and Rehabilitation*, vol. 8, no. 1, p. 60, 2011, ISSN: 1743-0003. DOI: 10.1186/1743-0003-8-60. [Online]. Available: http://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-8-60 (visited on 03/11/2019).

[9] D. D. Damian, M. Fischer, A. Hernandez Arieta, and R. Pfeifer, "The role of quantitative information about slip and grip force in prosthetic grasp stability," *Advanced Robotics*, vol. 32, no. 1, pp. 12–24, Jan. 2, 2018, ISSN: 0169-1864, 1568-5535. DOI: 10.1080/01691864.2017.1396250. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/01691864.2017.1396250 (visited on 03/06/2019).

[10] M. Markovic, M. A. Schweisfurth, L. F. Engels, T. Bentz, D. Wstefeld, D. Farina, and S. Dosen, "The clinical relevance of advanced artificial feedback in the control of a multi-functional myoelectric prosthesis," *Journal of NeuroEngineering and Rehabilitation*, vol. 15, no. 1, Dec. 2018, ISSN: 1743-0003. DOI: 10.1186/s12984-018-0371-1. [Online]. Available: https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-018-0371-1 (visited on 02/21/2019).

[11] M. Sartori, M. Reggiani, D. Farina, and D. G. Lloyd, "EMG-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity," *PLoS ONE*, vol. 7, no. 12, P. L. Gribble, Ed., e52618, Dec. 26, 2012, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0052618. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0052618 (visited on 10/15/2019).

[12] C. Pizzolato, D. G. Lloyd, M. Sartori, E. Ceseracciu, B. J. Fregly, and M. Reggiani, "CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks," p. 23, 2016.

[13] M. Sartori, G. Durandau, S. Doen, and D. Farina, "Robust simultaneous myoelectric control of multiple degrees of freedom in wrist-hand prostheses by real-time neuromusculoskeletal modeling," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066 026, Dec. 1, 2018, ISSN: 1741-2560, 1741-2552. DOI: 10.1088/1741-2552/aae26b. [Online]. Available: http://stacks.iop.org/1741-2552/15/i=6/a=066026?key=crossref.24203caa63d0aa7e7a24cbd38ab35e99 (visited on 02/21/2019).

[14] E. D. Engeberg, S. G. Meek, and M. A. Minor, "Hybrid forcevelocity sliding mode control of a prosthetic hand," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1572–1581, May 2008, ISSN: 0018-9294. DOI: 10.1109/TBME.2007.914672.

[15] R. Andrecioli and E. D. Engeberg, "Adaptive sliding manifold slope via grasped object stiffness detection with a prosthetic hand," *Mechatronics*, vol. 23, no. 8, pp. 1171–1179, Dec. 2013, ISSN: 09574158. DOI: 10.1016/j.mechatronics.2013.10.006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957415813001980 (visited on 03/18/2019).

[16] R. Barone, A. L. Ciancio, R. A. Romeo, A. Davalli, R. Sacchetti, E. Guglielmelli, and L. Zollo, "Multilevel control of an anthropomorphic prosthetic hand for grasp and slip prevention," *Advances in Mechanical Engineering*, vol. 8, no. 9, p. 168 781 401 666 508, Sep. 2016, ISSN: 1687-8140, 1687-8140. DOI: 10.1177/1687814016665082. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1687814016665082 (visited on 03/07/2019).

[17] C. F. Pasluosta and A. W. L. Chiu, "Modulation of grasping force in prosthetic hands using neural network-based predictive control," in *Artificial Neural Networks*, H. Cartwright, Ed., vol. 1260, New York, NY: Springer New York, 2015, pp. 179–194, ISBN: 978-1-4939-2238-3 978-1-4939-2239-0. DOI: 10.1007/978-1-4939-2239-0_11. [Online]. Available: http://link.springer.com/10.1007/978-1-4939-2239-0_11 (visited on 03/15/2019).

[18] N. Wettels, A. Parnandi, Ji-Hyun Moon, G. Loeb, and G. Sukhatme, "Grip control using biomimetic tactile sensing systems," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 6, pp. 718–723, Dec. 2009, ISSN: 1083-4435, 1941-014X. DOI: 10.1109/TMECH.2009.2032686. [Online]. Available: http://ieeexplore.ieee.org/document/5290153/ (visited on 03/07/2019).

[19] N. Hogan, "Impedance control: An approach to manipulation: Part itheory," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–7, Mar. 1, 1985, ISSN: 0022-0434, 1528-9028. DOI: 10.1115/1.3140702. [Online]. Available: https://asmedigitalcollection.asme.org/dynamicsystems/article/107/1/1/400604/Impedance-Control-An-Approach-to-Manipulation-Part (visited on 10/19/2019).

[20] J. E. Colgate and N. Hogan, "Robust control of dynamically interacting systems," *International journal of Control*, vol. 48, no. 1, pp. 65–88, 1988.

[21] A. Q. Keemink, H. van der Kooij, and A. H. Stienen, "Admittance control for physical humanrobot interaction," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1421–1444, Sep. 2018, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364918768950. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364918768950 (visited on 02/21/2019).

[22] J. Lobo-Prat, K. Nizamis, M. M. Janssen, A. Q. Keemink, P. H. Veltink, B. F. Koopman, and A. H. Stienen, "Comparison between sEMG and force as control interfaces to support planar arm movements in adults with duchenne: A feasibility study," *Journal of NeuroEngineering and Rehabilitation*, vol. 14, no. 1, Dec. 2017, ISSN: 1743-0003. DOI: 10.1186/s12984-017-0282-6. [Online]. Available: http://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-017-0282-6 (visited on 03/07/2019).

[23] S. Schaffelhofer, M. Sartori, H. Scherberger, and D. Farina, "Musculoskeletal representation of a large repertoire of hand grasping actions in primates," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, pp. 210–220, Mar. 2015, ISSN: 1534-4320, 1558-0210. DOI: 10.1109/TNSRE.2014.2364776. [Online]. Available: http://ieeexplore.ieee.org/document/6936371/ (visited on 10/27/2019).

[24] Ottobock, *Ottobock electrodes*, https://shop.ottobock.us/c/Electrode/p/13E200~550, 2019.

[25] J. T. Belter, J. L. Segil, A. M. Dollar, and R. F. Weir, "Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review," *The Journal of Rehabilitation Research and Development*, vol. 50, no. 5, p. 599, 2013, ISSN: 0748-7711. DOI: 10.1682/JRRD.2011.10.0188. [Online]. Available: http://www.rehab.research.va.gov/jour/2013/505/pdf/belter505.pdf (visited on 07/14/2019).

[26] H Olsson, K. Astrom, C Canudas de Wit, M Gafvert, and P Lischinsky, "Friction model and frictio compensation," *European Journal of Control*, vol. 4, no. 3, pp. 176–195, 1998. DOI: https://doi.org/10.1016/S0947-3580(98)70113-X.

[27] B. Sagbas, "Biotribology of artificial hip joints," in *Advances in Tribology*, P. H. Darji, Ed., InTech, Oct. 26, 2016, ISBN: 978-953-51-2742-0 978-953-51-2743-7. DOI: 10.5772/64488. [Online]. Available: http://www.intechopen.com/books/advances-in-tribology/biotribology-of-artificial-hip-joints (visited on 07/16/2019).

[28] D Weir and J Colgate, "Stability of haptic displays," in *Haptic Rendering*, M. Otaduy, Ed., A K Peters/CRC Press, Jul. 25, 2008, pp. 123–156, ISBN: 978-1-56881-332-5 978-1-4398-6514-9. DOI: 10.1201/b10636-9. [Online]. Available: http://www.crcnetbase.com/doi/10.1201/b10636-9 (visited on 11/14/2019).

[29] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach*. John Wiley & Sons, 2012.

[30] M. Focchi, G. A. Medrano-Cerda, T. Boaventura, M. Frigerio, C. Semini, J. Buchli, and D. G. Caldwell, "Robot impedance control and passivity analysis with inner torque and velocity feedback loops," *Control Theory and Technology*, vol. 14, no. 2, pp. 97–112, May 2016, ISSN: 2095-6983, 2198-0942. DOI: 10.1007/s11768-016-5015-z. [Online]. Available: http://link.springer.com/10.1007/s11768-016-5015-z (visited on 11/13/2019).

[31] A. Ninu, S. Dosen, S. Muceli, F. Rattay, H. Dietl, and D. Farina, "Closed-loop control of grasping with a myoelectric hand prosthesis: Which are the relevant feedback variables for force control?" *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 5, pp. 1041–1052, Sep. 2014, ISSN: 1534-4320, 1558-0210. DOI: 10.1109/TNSRE.2014.2318431. [Online]. Available: http://ieeexplore.ieee.org/document/6807741/ (visited on 11/18/2019).

[32] S. Dosen, M. Markovic, K. Somer, B. Graimann, and D. Farina, "EMG biofeedback for online predictive control of grasping force in a myoelectric prosthesis," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, no. 1, p. 55, Dec. 2015, ISSN: 1743-0003. DOI: 10.1186/s12984-015-0047-z. [Online]. Available: http://www.jneuroengrehab.com/content/12/1/55 (visited on 11/13/2019).

[33] R. G. Radwin, S. Oh, T. R. Jensen, and J. G. Webster, "External finger forces in submaximal five-finger static pinch prehension," *Ergonomics*, vol. 35, no. 3, pp. 275–288, Mar. 1992, ISSN: 0014-0139, 1366-5847. DOI: 10.1080/00140139208967813. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00140139208967813 (visited on 11/14/2019).

[34] R. W. Gregory, "Biomechanics and control of torque production during prehension," 2002.

[35] K. J. Bretz,. Jobbgy, and K. Bretz, "Force measurement of hand and fingers," *Biomechanica Hungarica*, Apr. 2010, ISSN: 20600305, 20604475. DOI: 10.17489/biohun/2010/1/07. [Online]. Available: http://biomechanica.hu/index.php/biomech/article/viewFile/86/141 (visited on 11/14/2019).

[36] E. Burdet, D. W. Franklin, and T. E. Milner, *Human robotics: neuromechanics and motor control*. Cambridge, Massachusetts: The MIT Press, 2013, 277 pp., ISBN: 978-0-262-01953-8.

[37] A. Ajoudani, S. B. Godfrey, M. Bianchi, M. G. Catalano, G. Grioli, N. Tsagarakis, and A. Bicchi, "Exploring teleimpedance and tactile feedback for intuitive control of the pisa/IIT SoftHand," *IEEE Transactions on Haptics*, vol. 7, no. 2, pp. 203–215, Apr. 2014, ISSN: 1939-1412. DOI: 10.1109/TOH.2014.2309142.

[38] Q. Fu and M. Santello, "Improving fine control of grasping force during handobject interactions for a soft synergy-inspired myoelectric prosthetic hand," *Frontiers in Neurorobotics*, vol. 11, Jan. 10, 2018, ISSN: 1662-5218. DOI: 10.3389/fnbot.2017.00071. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnbot.2017.00071/full (visited on 03/01/2019).

Appendix A

EMG-driven musculoskeletal model description, scaling and calibration

A.1 Model description

The generic OpenSim model used in this work was a full body biomechanical model. For the use in this work, the model was reduced to the right arm muscles described in Table I in Chapter 2. The model was restricted to only allow pronation-supination and index finger flexion-extension movements. The MTUs (under 'Muscles' in the 'Navigator' view in OpenSim) implied in index finger flexion-extension where: extensor digitorum communis indices (EDCI), extensor indicisproprius (EIP), flexor digitorum sublimis indices (FDSI), flexor digitorum profundus indices (FDPI) and flexor digitorum profundus (FDPM). Therefore, the joints used in the model where the wrist and the metacarpophalangeal (MCP) joint of the index finger. The rest of arm joints, i.e. elbow, shoulder and the MCP joints of the rest of the fingers were welded. The resulting torque predicted by the MTUs of finger flexion-extension was used for prosthetic hand open and closing movements.

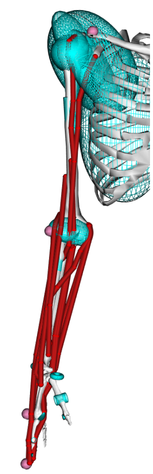


FIGURE A.1: OpenSim model used in this work.

Two important processes to accomplish a subject-specific EMG-driven musculoskeletal model are: 1) model scaling and 2) model calibration. Although these two process are briefly described in Chapter 2, a more profound explanation of both processes is given in this appendix. Note that the explanations given are according to OpenSim 3.3, where the name and placement of the tabs and tags can vary from version to version.

A.2 Model Scaling

Model scaling consists on scaling a generic OpenSim model to the anthopometric size of the subjects. Because the model used in this work comprised of the right arm, only subject's right arm was measured to scale the generic model.

The anatomical position of the shoulder joint, elbow, wrist, MCP joint of the middle finger and index finger length were considered to scale the model. For this, the first step is to mark the measuring points in the generic model of OpenSim. This can be done by going to the 'Navigator' view, then 'Markers', right click 'Add new marker'. The markers are place in the anatomical landmarks described before. Because we will measure the anatomical distance from one landmark to another, it is important to put all the markers in the same reference coordinates. In the model used in this work, all landmarks were set to the reference coordinates of the body ’ground’. This can be set by clicking the marker and going to ’*name of marker*-Properties’ and setting the ’body’ property to ’ground’. The positions of the each marker should appear in the ’position’ tag in the properties of the marker. The positions of each marker are written down for posterior computation of the distances between markers. The distance between the markers is the length of each body segment described above. Fig. A.2 shows the placement of the markers in the generic model.

Afterwards and by using the placement of the markers in the generic model as reference, the distances between the anatomical positions of the shoulder joint, elbow, wrist, MCP joint of the middle finger and index finger length are taken for the subject. The generic model will be scaled by computing the factor between the segment lengths of the generic OpenSim model and the segment lengths of the subject. This can be done in the custom created Matlab function ’computeScalingFactors.m’. This function takes as input the position of each marker and the length of the segments of the subjects in array form. The values are input manually and the output is the factor for each segment. Note that the factor computed for the segment between the wrist and the middle finger MCP landmarks (i.e. the metacarpal bones) is applied to the metacarpal bones of the rest of the fingers. This case is the same for the length of the index finger: the rest of the fingers are scales with the same factor. Therefore, the output of the function give the factors to scale: the humerus, radius and ulna, metacarpal bones and phalanxes.

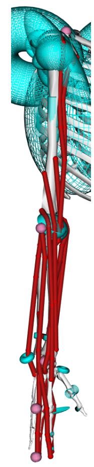


FIGURE A.2: Position of anatomical landmarks marked with pink markers. Note the marker in the wrist.

Finally, the scaling of the model is performed by going to the tab ’Tool’, the ’Scale Model’, ’Scale Factors’, select the body part(s) we want to scale, and press ’Use Manual Scales’ to enter the scaling factor. Click ’Run’ and the new scaled model will appear in the same window. The scaled model is saved with a different name from the original containing subjects name or reference.

A.3 EMG calibration

First, EMG was calibrated for each subject by measuring the maximal EMG and baseline signal across all 8 electrodes channels. Remember that the 8 electrodes have to be correctly placed according to the EMG positions shown in Table I, Chapter 2. This means that each channels will always correspond to the same muscle. EMG calibration needs to be done *each time* the prosthesis is used or the musculoskeletal model is calibrated.

For the calibration, the subject follows a sequence of postures where he/she has to try to achieve maximal force, or Maximal Voluntary Contraction (MVC). The sequence of movements is as follows: wrist flexion, wrist extension, wrist pronation, wrist supination, ulnar deviation, radial deviation, open, close and rest (no movement). The subject is seated in a chair and he/she has to perform the movement against two clamp bars mounted in the table. For the closing exercise the subject is given a non-breakable objects to grasp, such as hard foam. The subject is given visual feedback and instruction by plotting the activity of each EMG channel in a simple GUI (Graphical User Interface). The GUI and the processing of the EMG is done in the function 'emgCEINMSCalibration.m'. Once the sequence of movements is finished, the function will take the maximal value and baseline value registered through each channel. The maximal value for each channel is computed across all movements while the baseline is computed only from the 'rest' phase. The output values are stored in a 8x2 matrix in text form containing MVC values and baselines for each of the 8 electrodes. At the end, the GUI asks for a file name to save the text matrix, which could be named as '*subject name_date.txt*' (without the txt extension).

To normalize the EMG, during any running program to control the Michelangelo hand, the incoming EMG values must be: baseline subtracted and divided by the MVC values before any other processing is made. For this, the txt containing the baseline and maximum MVC levels of each channels needs to be read at the beginning of any control program as shown in Listing A.1.

LISTING A.1: Code to open a txt calibration file for 8 channels

```
%% Open calibration file. Paste this wherever you need to open the calibration file
if ~exist('filename','var')
filename = uigetfile('*.*');
end
formatSpec = ['%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n%6.2f %6.2f\n'];

fid = fopen(filename);
data = textscan(fid,formatSpec);
fclose(fid);
calibration = cell2mat(data);
calibration = reshape(calibration,[2,8]);%results in 3 (mvc, rest) x NumberOfChannels (which is 8)
clear data;
```

A.4 Model calibration

Overview

The calibration of the EMG-driven musculoskeletal model implies following predefined motions of the prosthetic hand. This is, the prosthesis is programmed to follow certain path and the subject has to mimic this path.

More specifically, model calibration establishes the relationship between MVC level (stracted from normalized EMG), torque production and prosthetic hand position. That is, certain MVC level produces certain torque according to MTU properties at certain muscle position. The position of the prosthetic hand is then an estimation of the position of the phantom limb. The velocity input given to the prosthetic hand to perform the predefined movements is taken as an approximation of the torque 'produced'. This torque is the experimental torque τ_{exp}. For each DOF to be calibrated, a different calibration experiment has to be done according to the motion of the DOF to calibrate: e.g. if hand open-close needs to be calibrated, the prosthesis is commanded to open and close; if wrist pronation-supination needs to be calibrated, the prosthesis is commanded to pronate and supinate.

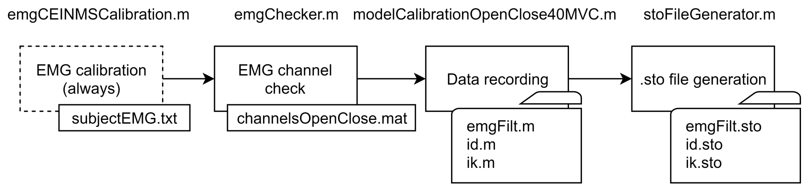


FIGURE A.3: Process of data recording for model calibration

Calibration data recording process

First of all, Fig. A.3 provides a simple overview of the calibration data recording process, the Matlab functions involved and the output files created by of each of them. The explanations following these lines are made according to that scheme.

For the calibration of the open-close movement, the function 'modelCalibrationOpenClose40MVC.m' is used, while 'modelCalibrationProSup40MVC.m' is used for pronation-supination calibration. In this program, the prosthetic hand is commanded to open and close at a *constant* speed. This constant speed is an approximation of τ_{exp}. A GUI is presented to the subject were he/she is asked to match 40% of MVC level marked by a read line in the GUI plot. The constant speed is set to 40% of the maximum velocity that can be commanded to the prosthesis. Therefore, a 40% MVC level would correspond to 40% of the maximum prosthesis open-close velocity, which gives an approximation of τ_{exp}. Hence, a 40% MVC level corresponds to 40% of torque production. The ratio of 40% was found to be a good balance between the effort the subject has to make to reach 40% of MVC and the constant speed commanded to the prosthetic hand.

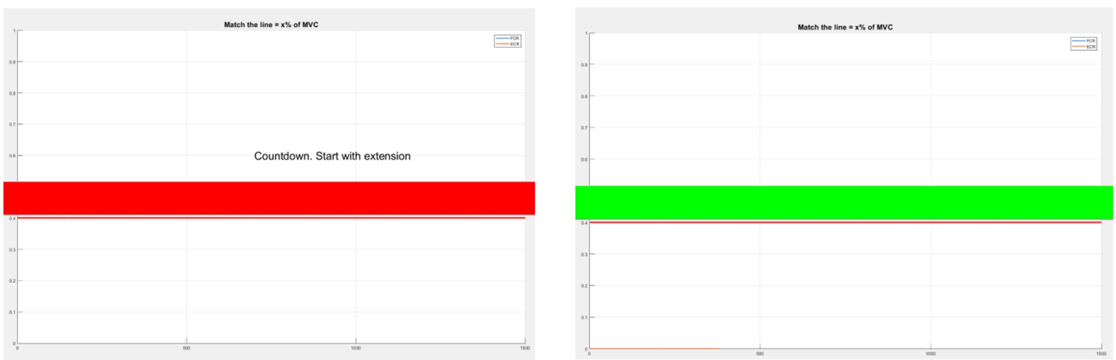


FIGURE A.4: Calibration GUI

During the task, the subject needs to mimic the movement of the prosthetic hand for 30 seconds. However, it is hard to maintain certain MVC level during the whole range of the open-close movement. That is why the subject is asked to reach the open or the close position as fast as possible and maintaining the contraction level at the extremes of the motion range. That is, if the prosthesis is *about to* close, the subject suddenly fully closes his/her hand and maintain the MVC level at fully closed hand. For this, the subject is given a visual queue in the GUI: a green queue tells the user to fully open, while a green queue tells the user to fully close. The timing on when to fire the queues was experimentally chosen. This is because activation precedes torque production, that is why the visual queue to fully open or close is given *just before* any of those movements happen. This takes into account reaction time to visual queue and movement production.

Furthermore, because the position of the aperture of the hand is given in values from 0 to 100 % (100% means fully open), this range needs to be transformed into radians for a realistic measure of the aperture which makes sense with the aperture range in the musculoskeletal model. That is, the aperture range in the musculoskeletal model and the prosthetic hand needs to be coherent. For this, the incoming value of the aperture of the hand is transformed to the aperture range of given in the musculoskeletal model, which is between -1.221730476396 (fully open) and 1.221730476396 radians (fully closed). The transformation is done as in A.1, where 'real aperture' is the aperture values of the prosthesis between 0 and 100%.

$$aperture\ rads = (-real\ aperture * ((2 * 1.221730476396)/100)) + 1.221730476396 \quad \text{(A.1)}$$

In the same lines, if pronation-supination is desired, the incoming values of the Michelangelo need to be changed to radians. For this, the pronation-supination of the Michelangelo is limited to -90 to 90 degrees. The pronation-supination values registered by the Michelangelo are given in the range to [-100%, 100%]. Then, the range that corresponds to -90 to 90 degrees is [-56.25%, 56.25%]. Therefore, the formula to convert percentage to radians is A.2.

$$\begin{aligned} rotation\ rads\ &= ((real\ rotation - (-56.25))/(56.25 - (-56.25))) \\ &\quad (1.57 - (-1.57)) + (-1.57) \end{aligned} \quad \text{(A.2)}$$

During the calibration, normalized EMG, position and τ_{exp} (approximation by input constant speed) are recorded at 100 Hz. After the calibration experiment is completed, the arrays are synchronized by interpolation; that is, an ideal time vector is used to interpolate the samples of each array. Then, the program plots the result of the synchronization: the resulting EMG, τ_{exp} and position are plotted together. Here, the researcher needs to assess if the calibration was good enough by judging if the EMG indeed preceded τ_{exp} (i.e. the constant velocity commanded to the hand). Furthermore, the EMG needs to precede the position where the prosthetic hand started to change from open to close or vice-versa. An example is provided in Fig. A.5. The calibration lasts 30 seconds to have enough time to judge which were the best intervals following the reasoning explained. The researcher directly selects two periods from the plot by clicking the two points defining each period (the Matlab function 'ginput' is used for this). Then, the program automatically cuts and saves the resulting EMG, position and τ_{exp} arrays in a .mat file named 'aperture*date*' under an automatically created folder (in current Matlab path) named 'SubjectCalibration'.

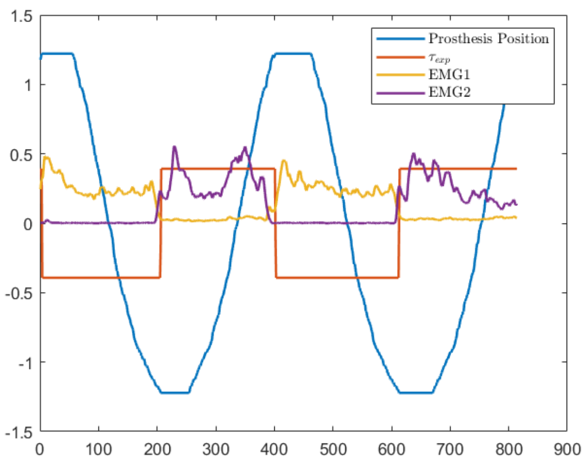


FIGURE A.5: Example of calibration for hand open-close.

Further details during calibration

During the calibration experiment, a 40% MVC level needs to be. However, not all muscles implied in opening and closing will reach 40% MVC as the have different contributions. For this reason, only 2 EMG channels are plotted during the calibration. These 2 channels are selected to be representative enough of when the subject is opening and closing. As this may vary between subjects, the function 'emgChecker.m' can help on deciding which channels may be appropriate. This function program just plots all channels and the same 40% MVC lines as during the calibration, but no data array is recorded during this check. The subject is asked to try to reach the MVC level and the researcher decides which channels are more representative of each action. At the end of the simulation, a command line prompt asks to enter which two channels are chosen for opening and closing, which are saved in a .mat file under the name 'channelsOpenClose.mat'. This process will help to have a clearer vision in the moment to assess which intervals to chose during the calibration process.

Subject-specific parameter adjustment procedure

The subject-specific EMG-driven musculoskeletal model is calibrated for each subject by fine tuning parameters that vary non-linearly across subjects. These parameters include: muscle twitch activation/deactivation time constants, EMG to-activation non-linearity factor, muscle optimal fiber length, tendon slack length, and muscle maximal isometric force [29]. Initial parameters values are adjusted as part of a least-squares optimization procedure, so that the error between τ_{exp} and τ_{pred} (torque predicted by the musculoskeletal model) is minimized. For this procedure, there is a specific C++ calibration program: 'calibrate.exe', which opens though the command line in Microsoft Power Shell. As explained in Chapter 2, section II-B, the calibration takes as input the EMG, τ_{exp} and position angles of the prosthesis that were obtained during the offline data recording explained in appendix section A.4. The calibration program takes its input in the form of *.sto* or *.mot* file extension, similar to a formatted .txt file. To convert the .mat array of EMG, τ_{exp} and aperture angles obtained during data recording session, the custom function 'stoGenerator.m' is used. Here, the program asks the researcher to select the .mat file he/she wants to convert. The researcher shall navigate to folder 'SubjectCalibration' to find the desired .mat folder. The output of the program are 3 .sto files: emgFilt.sto, id.sto and ik.sto. The last two contain τ_{exp} and aperture angles, respectively. The 3 .sto files are automatically saved in one folder with subject's name and date and in the path from which the calibration program will read the files.

The calibration comprises of 3 steps: spline calculation, pre-scaling and parameter calibration. These 3 steps need different inputs for each subject. Each input file needs to be placed correctly in 3 different xml files: subjectAnnealing.xml (see Fig. A.7), executionRT.xml and executionIK.xml. These files will be mentioned in the following lines and special attention must be taken to *what things need to be changed* for each subject. An overview of the process is shown in Fig. A.6 and a complete explanation on how each component is related to each other can be found in [31].

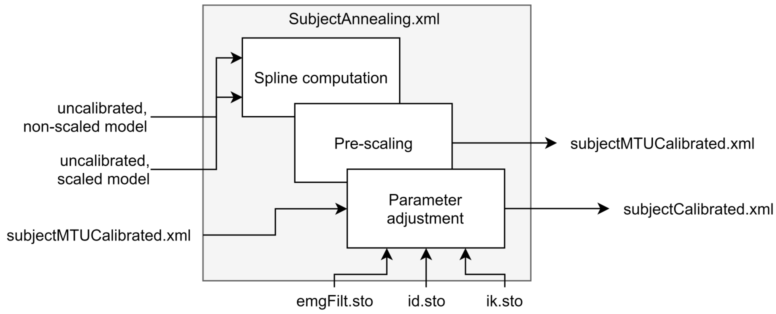


FIGURE A.6: Model calibration process and files involved on each step.

Spline computation

During spline calculation, the moment arms r^{mt} and the length l^{mt} of each MTU are calculated out of the aperture angles and the output are the coefficients of the spline. This process needs of: 1) an uncalibrated, non-scaled generic OpenSim model (already containing only the MTUs of interest); 2) the uncalibrated, subject-scaled OpenSim model. This two models are loaded in the subjectAnnealing.xml file. Model number 1) needs to be placed under the <SubjectXML> tag; model number 2) needs to be placed under <OSimfile> tag. Then, under the <NameOfSubject> tag a name for the output spline coefficients must be given, in the likes of '*subject name + date*'. Check that in the example provided in Fig. A.7, the DOF that will be calibrated is the one under <dofToCalibrate> tag. In this case, the DOF to calibrate is index finger flexion-extension (which is mapped to prosthetic hand open-close), where the specific name of the DOF is '2mcp_flexion'. The naming and available DOFs are specified in the OpenSim file under the 'Coordinates' tab. Afterwards, the name given to the computed spline coefficients needs to be loaded in the executionRT.xml file under the tag <NameOfSubject> (see Fig. A.8), and the correct scaled model needs to be loaded in executionIK.xml under the tag <OsimFile>. The model *must be the same* as the one in line 20 in subjectAnnealing.xml (Fig. A.7).

```xml
<?xml version="1.0"?>
<simulatedAnnealing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="../../XSD/simulatedAnnealing.xsd">
    <EMGProccesing>cfg/Arm/executionEMG_10K.xml</EMGProccesing>
    <!-- <subjectXML>cfg/Arm/Right_arm.xml</subjectXML> -->
    <subjectXML>cfg/Arm/Right_arm_OpenCloseMCP.xml</subjectXML>
    <!-- very first model, unscaled and uncalibrated-->
    <NameOfSubject>splineScaledModelCARLOS_thirdmc_MCP_14112019</NameOfSubject>
    <dofToCalibrate>2mcp_flexion</dofToCalibrate>
    <computeSpline>
        <use>true</use>
        <osimFile>cfg/Arm/LeftArm_scaled_Spline_changes_35_addedPTmuscle _MARKERS_2MCP_4 - GoodMarkers_CARLOS_SCALED.osim</osimFile>
        <TranslateFile>cfg/Arm/LeftArm_scaled_Spline.txt</TranslateFile>
        <printingOption>3</printingOption>
        <numberOfNode>9</numberOfNode>
    </computeSpline>
    <preScaling>
        <use>true</use>
        <unscaledOsimFile>cfg/Arm/LeftArm_scaled_Spline_changes_35_addedPTmuscle _MARKERS_2MCP_4.osim</unscaledOsimFile>
        <osimFile>cfg/Arm/LeftArm_scaled_Spline_changes_35_addedPTmuscle _MARKERS_2MCP_4 - GoodMarkers_CARLOS_SCALED.osim</osimFile>
        <TranslateFile>cfg/Arm/LeftArm_scaled_Spline.txt</TranslateFile>
    </preScaling>
    <calibration>
        <ObjectiveFunction>StrengthCoefficients_ShapeFactor_TendonSlackLength_single</ObjectiveFunction>
        <use>true</use>
        <filterEMG>false</filterEMG>
        <dofCalibrationSequence>
            <Dof>2mcp_flexion</Dof>
            <!-- <Dof>pro_sup</Dof> -->
        </dofCalibrationSequence>
        <trials>
            <directory>cfg/Arm/</directory>
            <trial>
                <trialName>CarlosapertureNovember14</trialName>
                <cropMinTime>0</cropMinTime>
                <cropMaxTime>7.47</cropMaxTime>
            </trial>
        </trials>
        <option>
        <noEpsilon>4</noEpsilon>
            <epsilon>1.E-3</epsilon>
            <maxNoEval>300000</maxNoEval>
        </option>
    </calibration>
</simulatedAnnealing>
```

FIGURE A.7: Example code of the subjectAnnealing.xml file

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<execution xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../XSD/execution.xsd">
  <NMSmodel>
     <type>
        <realTime>
            <openLoop/>
        </realTime>
     </type>

     <tendon>
          <stiff/>
     </tendon>

     <activation>
          <exponential/>
     </activation>
     <curve>
        <online/>
     </curve>
  </NMSmodel>

  <ConsumerPlugin>
    <EMGDevice>plugin/EMG_UDP_Simulink/bin/Win/Debug/EMG_UDP_Simulink.dll</EMGDevice>
    <EMGDeviceFile>cfg/Arm/executionEMG_ankle_knee_new_filter.xml</EMGDeviceFile>
    <AngleDevice>libangle_0_plugins.dll</AngleDevice>
    <AngleDeviceFile>cfg/Arm/executionIK.xml</AngleDeviceFile>
    <TorqueDevice/>
    <TorqueDeviceFile/>
    <ComsumerPort>1234</ComsumerPort>
    <AngleAndComsumerDevice>plugin/Torque_Position_UDP_Simulink/bin/Win/Debug/Torque_Position_UDP_Simulink.dll</AngleAndComsumerDevice>
  </ConsumerPlugin>
   <NameOfSubject>splineScaledModelCARLOS_thirdmc_MCP_14112019</NameOfSubject>
</execution>
```

FIGURE A.8: Example code of the executionRT.xml file

Pre-scaling

During pre-scaling, the initial guesses for subject-specific parameter for the parameter adjustment procedure are found. The pre-scaling takes as input the same two models as in spline computation. The output is an xml file called 'subjectMTUCalibrated.xml'.

Parameter adjustment

In this process, subject-specific parameters are found. The input for this are: subjectM-TUCalibrated.xml, emgFilt.sto, id.sto and ik.sto, which are all contained in a file with the name of the subject (e.g. in Fig. A.7 this would be 'subjectapertureNovember14'). Such folder needs to manually be loaded under the tag <trialName>. The <Dof> under <dof-CalibrationSequence> tags marks which DOF(s) will be calibrated. If more than one DOF needs to be calibrated, this should be specified as in line 29 and another specific calibration file should be another in a subsequent <trial> tag. The output of this process is a file named 'subjectCalibrated.xml' which contains the scaled and calibrated subject-specific musculoskeletal model. One may change the name of this file by substituting 'subject' by subject's name of reference and save the file properly for future usage.

How to start the calibration

The calibration program (calibration.exe) is started on Microsoft PowerShell, as shown in Fig. A.9. The result of the calibration, plotting the fitted polynomial to τ_{exp}, is shown in a GUI (see Fig. A.10). The researcher may judge the result of the calibration visually and that the calibration algorithm could actually converge to one result. The approximate calibration for 1 DOF should be less than 1 hour.

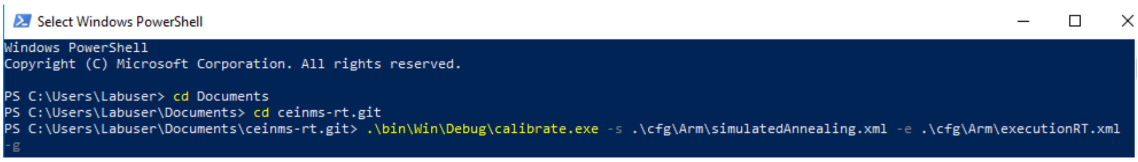


FIGURE A.9: Example of how to start the calibration program. Note that calibration.exe should be opened from the correct path. Furthermore, the -s tag define the simulated annealing file, -e the execution file and -g open the GUI. Type -h for further help.

FIGURE A.10: An example of the result of calibration of the index finger flexion-extension.

General notes

The rest of the parameters that were not mentioned in the previous lines shall not be modified. An important note is that the <use> tag marks which of the 3 calibration process will be performed. For example, if we had already computed the splines, the <use> tag under <computeSpline> should be set to <false>. Another note is that the <cropTimeMaxTime> defines the length of the .sto files loaded in the <triaName> folder. This number is given in the command line by the 'stoFileGenerator.m' after .sto files are created. Otherwise, it can be found in the time array of any of the .sto files. However, the value of <cropTimeMax> should be *at least* 0.05 seconds less than the length of the .sto files. This is because the program shifts the EMG to make sure it precedes τ_{exp}.

Appendix B

System communication framework

The communication between the EMG-driven musculoskeletal model (referred as CEINMS in the following lines for convenience), control program and prosthesis is explained in this appendix. Furthermore, the program used to control the prosthetic hand, both via admittance and non-admittance, is explained.

B.1 Framework overview

The framework, shown in Fig. B.1, consists of three parts: CEINMS model, Matlab processing and high-level control program and the graphical user interface (GUI) of the Michelangelo hand. The connection between the three parts is made possible via User Datagram Protocol (UDP) connection. The communication pipeline works as follows:

1. The Michelangelo GUI sends EMG and prosthetic angle *in percentage* values to the high-level control program implemented in Matlab.
2. Matlab sends EMG and prosthetic angles *in radians* to the CEINMS plugin.
3. The CEINMS plugin computes join torques based on EMG and prosthetic angles and sends this torque to Maltab.
4. Matlab high-level control program computes commands and sends them to the low-level controller of the Michelangelo hand via Michelangelo's GUI.

In the next sections, each of the components will be explained. This work did not develop the CEINMS plugin and therefore cannot provide the insights of the program; however, it will be important to explain how the UDP connection between CEINMS and MATLAB is made.

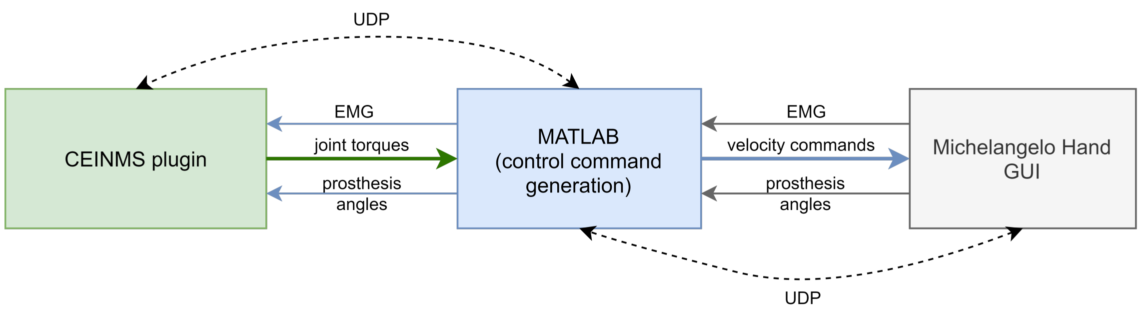


FIGURE B.1: Overview of the communication framework between the main 3 components.

B.2 Michelangelo hand interface

A scheme of the interface is shown in Fig. B.2. The main components are the electrodes, the AxonMaster central control unit, the Michelangelo prosthetic hand and the PC to which Ottobock's bluetooth dongle is connected. The AxonMaster has implemented lower-level controller by Ottobock, which is not accessible. Therefore, Matlab is used as a higher-level controller by sending and receiving data packets by a UDP protocol from and to the prosthesis. More specifically, the AxonMaster recieves processed EMG signals from the electrodes and sensor values derived from sensors integrated within the Michelangelo. The AxonMaster processes the data and sends them to the PC as datagrams for each sample. The PC reads the datagrams within the Matlab environment and processes them into actuation commands that are sent to prosthesis via the AxonMaster.

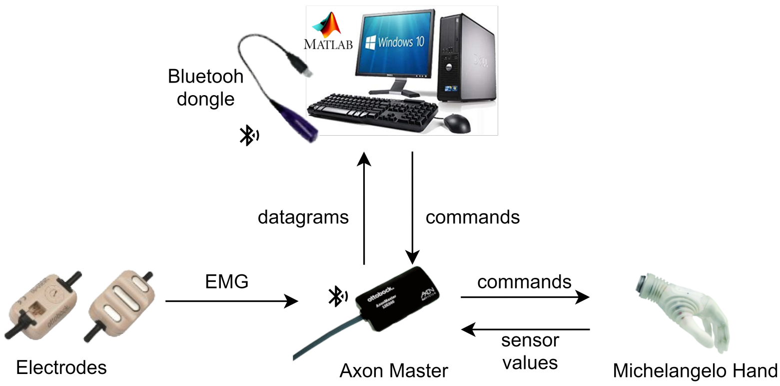


FIGURE B.2: Communication setup between the Michelangelo hand and high-level control in Matlab.

Datagram packages

Bluetooth datagram packages is the way the AxonMaster shares information with the PC. The AxonMaster samples the values of the sensors within the Michelangelo hand (position, force, etc.) and root mean square (RMS) values obtained from the raw EMG from the electrodes. For each sample, a datagram containing 35 bytes is send. Each of the bytes carries different information. Two bytes are used for each RMS, which for 8 channels makes a total of 16 bytes. The bytes corresponding to the RMS values need to be converted into uint16 datatype after sending and receiving the datagram. The rest of the bytes carry sensor values and other control variables' information, which need to be converted into bytes in either int8 or uint8.

Sensor information

The Michelangelo hand has several embedded sensors, such as position sensors for each servomotor and a force sensor in the base of the thumb that measures the closing force. The information of such sensors comes in percentage values, i.e., a fully opened hand will give an aperture angle of 100% and fully closed 0% aperture. The same holds for the force sensor, which gives maximum closing force of 100%. Because the measured force is not in physical units (Newtons), the embedded force sensor was calibrated with a load cell to gain insight of the real force range in Newtons (see Appendix E).

Michelangelo GUI

The Michelangelo GUI serves as a high-level control graphical interface itself and a mediator that allows high-level control from other platforms, such as Matlab or Python based control programs. This way, it receives bluetooth datagrams from the AxonMaster and allows to retrieve the datagrams for further usage in the custom control program. In this work, the datagrams are retrieved for processing in Matlab via UDP connection with the Michelangelo GUI. The usage and how to install all Michelangelo hand related installation files etc. is provided in the bitbucket repository CTW-BW/NMSTool/MMInterface/source code. Further usage of this GUI is explained in Appendix C.

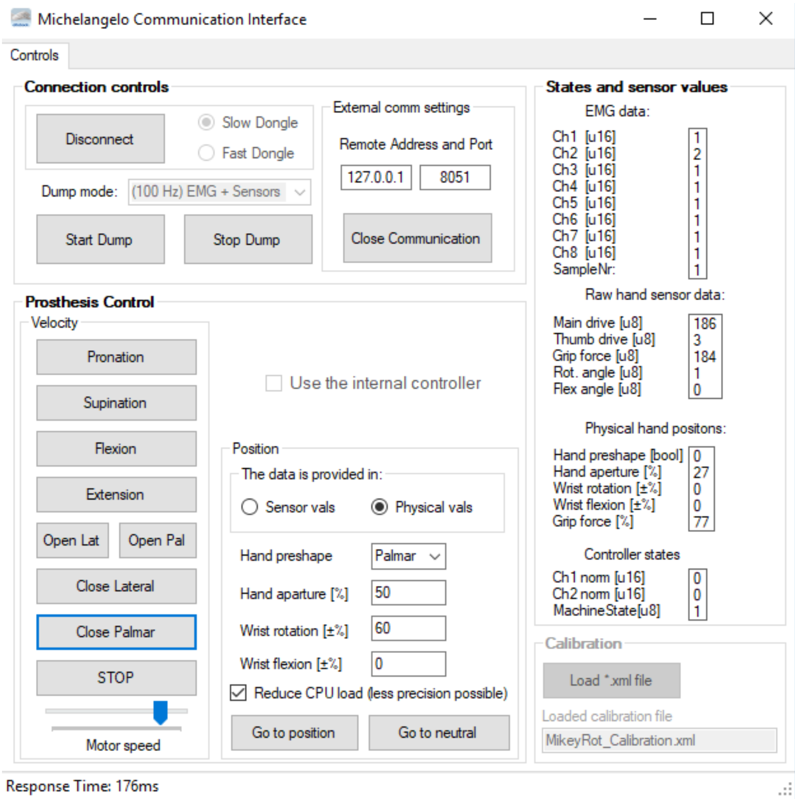


FIGURE B.3: Interface of Michelangelo GUI

B.3 Processing in Matlab

To retrieve datagram packages from the Michelangelo GUI in Matlab a UDP object is used. The datagrams are collected in an input buffer from which data can be read by the use of *fread* Matlab command. The DatagramTerminateMode property is disabled to continuously receive datagrams. The amount of datagrams received on each iteration depends on the number of samples chosen to be read (i.e. a window of samples). This is done to save processing time of the *fread* function. The reading frequency of *fread* is one of the bottlenecks of the whole framework. Because of this, a java-based UDP connection was implement with the thought that it would accelerate the UDP connection, although this method did not give better results. Furthermore, the received bytes are converted into the correct data type values as explained in section Datagram packages.

The next step in the pipeline is to generate appropriate commands based on the EMG and sensor information read from the datagrams. EMG is normalized and sensor information is converted into its respective units (percentage values of position are transformed into radians such as in A.1, Appendix A, and force values into Newtons such as in E.1, Appendix E).

To compute joint torques from EMG and prosthesis position angles (as explained in Chapter 2 section II-B), these are forwarded to the musculoskeletal model (i.e. CEINMS) via a UDP-based communication plug-in. Two different UDP connections are created: one for sending normalized EMG and another one to send aperture angles. Normalized EMG from 8 electrodes and aperture angles in radians are sent to CEINMS by the use of the *fwrite* function in Matlab by assigning the correct UDP object. In both cases we need to declare that we are sending 'double' formatted information via the UDP connection. Note that when sending position we need to do this in array format, where 3 position values are specified: hand aperture, hand rotation and elbow flexion. We do not use elbow flexion in this work so is set to a value of zero. Furthermore, if we just want to control 1 DOF, as in this work, we should set the 'rotation' value *always* to zero. A code example to send information to CEINMS is provided in Listing B.1.

LISTING B.1: Code snippet to sent prosthetic angles and EMG values to CEINMS plug-in

```
%Send position to CEINMS
aperture = (-app.posMickey*((2*1.221730476396)/100)) + 1.221730476396;
rotation =((rotation - (-56.25)) / (56.25 - (-56.25)))* (1.57 - (-1.57)) + (-1.57);%up to +-90

%Check if I want to control 2 dofs: open-close + pro-sup
if strcmp(oneORtwoDof,'2 DOF')
fwrite(udp_position,[rotation aperture 0],'double');%[rotation aperture elbow_flexion]
else %1 dof
fwrite(udp_position,[0 aperture 0],'double');%[rotation aperture elbow_flexion]
end

%Send normalized EMG to CEINMS
fwrite(udp_EMG,[emgNorm(1) emgNorm(2) emgNorm(3) emgNorm(4) emgNorm(5) emgNorm(6) emgNorm(7) emgNorm(8)],'double');
```

Furthermore, another code snippet to create the necessary UDP connections of EMG and position is given in Listing B.2.

LISTING B.2: Code snippet to create UDP objects

```
%% Create udp objects
function createCEINMSconnection()
 %------------EMG-----
 if ~exist('udp_EMG','var')
  local_IP = '0.0.0.0';
  EMG_port = 1233;
  udp_EMG = udp(local_IP,EMG_port, 'LocalPort', EMG_port); %Must be on to receive data.
  udp_EMG.InputBufferSize = 1000000;
```

```
        udp_EMG.DatagramTerminateMode = 'off'; %Can receive more than 35 bytes data at the same time
        udp_EMG.ByteOrder = 'LittleEndian';
        fopen(udp_EMG);
    end
    %------------Position----
    if ~exist('udp_position','var')
        local_IP = '0.0.0.0';
        position_port = 1234;
        udp_position = udp(local_IP,position_port, 'LocalPort', position_port); % Must be on to receive data.
        udp_position.InputBufferSize = 100000;
        udp_position.DatagramTerminateMode = 'off'; %Can receive more than 35 bytes data at the same time
        udp_position.ByteOrder = 'LittleEndian';
        fopen(udp_position);
    end
end
```

Note that the receiving port of each variable in the CEINMS plug-in is static and will always use the same port.

After sending normalized EMG and position values, CEINMS computes joint torques for the specified DOF. Torque is mapped into velocity via the admittance model or by direct, proportional mapping to velocity (i.e. non-admittance-based control). Velocity commands are sent for each DOF of the Michelangelo via its specific UDP and the *fwrite* command.

Appendix C

Prosthesis Control In Matlab

Due to restricted access to the low-level controller from Ottobock, Matlab was used to provide higher-level control of the prosthesis. A custom GUI was built for the control of the Michelangelo hand with the two control strategies presented in this work: admittance and non-admittance-based EMG-driven musculoskeletal modelling control. Therefore, all the processing, admittance model, connection to CEINMS and (velocity) command generation is implemented on this GUI.

C.1 Matlab GUI for prosthesis control: overview

During the development of this work, two versions of the GUI were created: one for testing and one for experiments. The testing version was made prior to the experimental one, which allowed to track the performance of the admittance coupled to the CEINMS model. Moreover, this version of the app allows to use the admittance model *without* the CEINMS model. Hence, the prosthetic hand can be controlled with conventional 2 channels and using admittance. However, this version *is not* as optimized as the one used in the experiments and special attention must be taken with velocity normalization. Furthermore, due to the amount of plotted information, the update frequency of the system is low. Therefore, the GUI that should be used for experiments is the experimental one presented after these lines.

The GUI was developed in Matlab under the App Designer environment, which provides a simple interface to create GUIs. This application can be saved as a Matlab add-on or a stand-alone desktop program. If any modification should be made, the app must be opened in Matlab by right click 'open'. This will open the app in the App Designer environment, which provides visual design and code view. The app can also be executed by pressing the 'run' button. The code mentioned in previous appendices may be found in the code view as implemented functions. It is of special interest the function 'eom' which has the *equation of motion* of the pendulum. If any parameter or admittance equation needs to be changed, it should be within this function. This prevents from altering the rest of the control code.

C.2 Description and steps for prosthesis control

In the following lines, as description of the app is given. More interestingly, it provides the steps the researcher shall follow to control the prosthesis through the CEINMS model.

Main page: running an experiment

The main page provides intuitive guidelines to control the prosthesis. The functionalities implemented and the steps to start prosthesis control are described below.

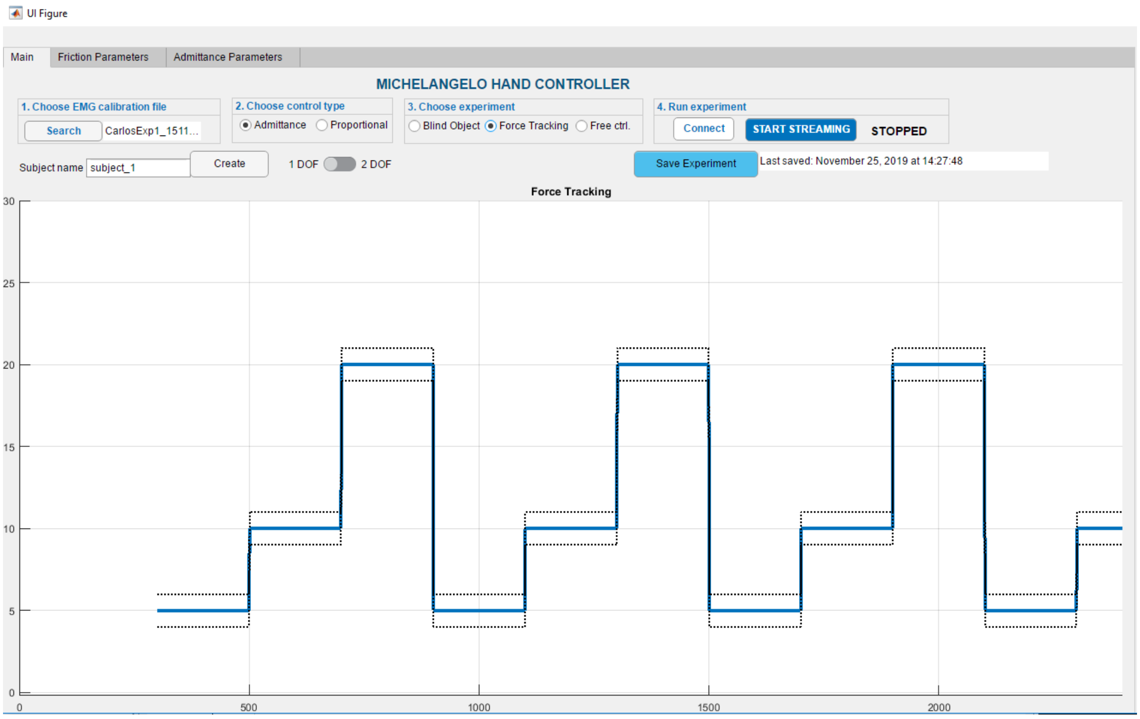


FIGURE C.1

1. Choose an EMG calibration file
The first thing is to load subject-specific EMG .txt calibration file. By clicking the search button a window of the current Matlab path will appear, where the desired EMG .txt file should previously be allocated. The name of the file will appear in the small window.

If any experiment is intended to be recorded, it is advised to create an specific experimentation session for the subject by entering subject's name or reference in the 'Subject name' edit field and pressing 'Create'. This will create a folder containing all experimental variables and data recordings. The file system and data storage system is explained in section C.4 in this appendix.

2. Choose control type

The GUI gives the option to control the prosthesis with admittance-based or non-admittance-based EMG-driven musculoskeletal model control. In the GUI, the latter is called 'Proportional', just because velocity commands will be 'just proportional' to the torque (perhaps a better naming would have been *direct* control).

Furthermore, in this moment the researcher shall chose between 1 DOF or 2 DOF control. By default, 1 DOF is active, which allow to control prosthetic hand opening and closing. 2 DOF allows this, plus prosthetic pronation-supination. Note that a proper calibration of the subject-specific CEINMS model for the 2 DOFs should have been previously made (refer to Appendix A).

3. Choose experiment

In case the researcher needs to perform the 'blind object' or 'force tracking' experiment, the option should be chosen accordingly. This will change the shape of the plot according to the experiment. If modifications in the shape of the reference line needs to be changed, this should be done in the 'generatePattern' function included in the code view. The control of the prosthesis is limited to the duration of each experiment: 120 seconds.

The option 'Free ctrl.' gives the option of just controlling the prosthesis with no time limitations. Then, this options should be selected to conduct functional tasks.

4. How to run an experiment

To control the prosthetic hand, first, a UDP connection with the Michelangelo hand GUI and a UDP connection with the CEINMS model plug-in must be established, as explained in Appendix B. The 'Connect' button will create the UDP connection to the Michelangelo and CEINMS model. Afterwards, the 'Start Experiment' button will allow control of the prosthesis. Some explicit steps need to be followed for successful connection of both UDPs and prosthesis control:

a) Connect to Michelangelo hand

To control the Michelangelo hand, the bluetooth dongle must be connected to the PC. Then, press 'Connect' *in the Michelangelo GUI*. A calibration file for the Michelangelo hand must be loaded by pressing 'Load *.xml file' in the 'Calibration' zone of such GUI. Select 'MikeyRot_Calibration.xml'. Then, click 'Start Communication'.

b) Click 'Connect' in the Matlab GUI

Once connect is pressed, the button will be disabled. In this point, the CEINMS model must be started. The order of the steps is important because Matlab must retrieve the unknown IP address and the connection ports of the CEINMS plug-in *before* trying to connect to it.

c) Start the CEINMS model

In the same way as the calibration, the CEINMS model must be started from the command line in PowerShell as shown in Fig. C.2. It is important to note that each subject has his/her own 'subjectCalibration.xml' and this must be placed in the right path (i.e. .\cfg\Arm\). After few seconds, the GUI of the real-time CEINMS model will open. In this point, the 'Connect' button of the Matlab GUI should be green indicating successful connection to both the Michelangelo hand and to the real-time CEINMS-model.

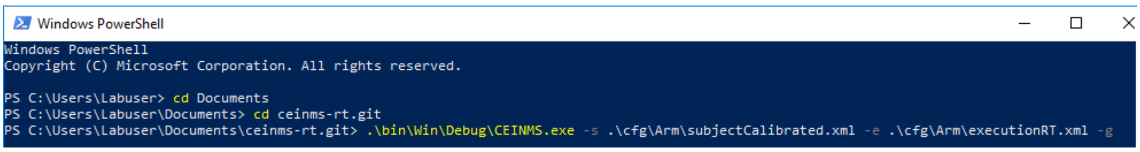


FIGURE C.2: Starting CEINMS model.

d) Click 'Start Experiment' in Maltab GUI

Once UDP connections are ready, press 'Start Experiment'. In should say 'waiting to Start Dump'.

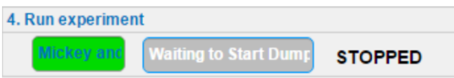


FIGURE C.3: Change of state of the 'Start Experiment' button

e) Click 'Start Dump' in Michelangelo GUI

The 'Start Dump' button is the most important to start control of the prosthesis. By pressing this button, the streaming of datagram packages from the AxonMaster will start. In this moment, in the Matlab GUI, the 'waiting to start dump' should change to 'Stop experiment' coloured in red.

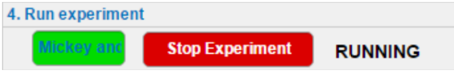


FIGURE C.4: Change of state of the 'Start Experiment' button to 'Stop experiment'

f) Stopping an experiment
To stop the experiment, click 'Stop experiment' red button in the Matlab GUI. This will end the stream of data and will close all UDP connections. If another experiment needs to be conducted, steps from b) to e) should be repeated.

C.3 Friction parameters

In this window, the different variables of the friction model can be changed. The plot is used as a reference to see how each variable affects the friction profile. Note that the damping value is included in this window to see how it affects the friction profile. The value of the parameters is adjusted with the slider and for Coulomb, Stribeck and damping value, specific values can be added in the edit field next to the sliders. Note that the friction model will only run in 'admittance' mode. *Values can be changed during a running experiment to see the change in performance.*

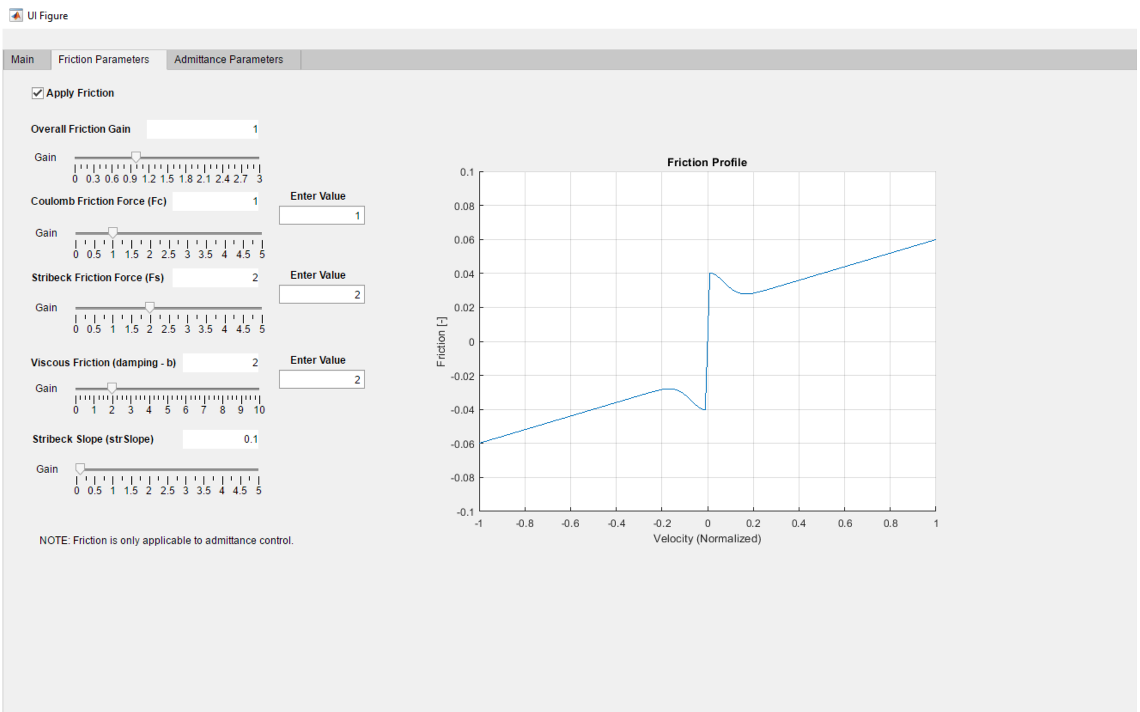


FIGURE C.5: Friction parameters window.

C.4 Admittance parameters

Different admittance parameters can be modified offline or *during a running experiment* to see how the performance changes. Here, we can change gain of the predicted torque

(τ_{pred}) by the CEINMS model and the sensitivity of the force sensor, where a gain of 1 will take the whole range of the sensor (74 N). Specific gains values can be added in the edit fields.

Furthermore, the maximum velocity values used to normalize velocity commands both from admittance and non-admittance can be added here. These values are the maximum velocities for opening and closing. Each subject will generate different torques and therefore will achieve higher or lower velocities. Therefore, these values should be adjusted prior to any experiment by performing a quick trial of subject's torque-velocity generating capabilities. This can be done by running a free control experiment where the subject is asked to perform maximal activation for closing for few seconds and the same for closing. This way we can see which are the peak velocities for each movement. It is also in this moment where subject preferences of admittance parameter values can be adjusted. Note that by changing those parameters the maximum velocities will change and normalization should be done according to those last maximum velocity values. This procedure is not implemented in the GUI for pronation-supination, which is left as future work.

On the other side, if normalizing velocities is not sufficient for smoother control, the overall velocity gain can be adjusted through its slider. Lastly, the virtual mass of the pendulum (m_v) and length of the rod (l_{rod}) can be adjusted in this screen

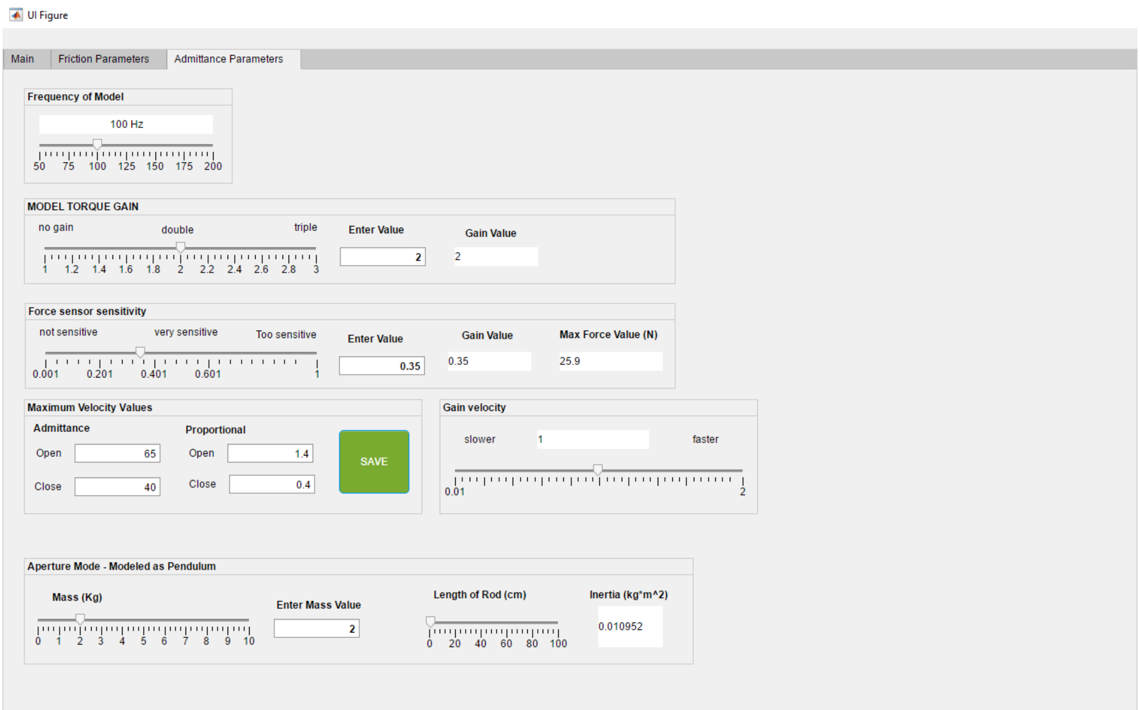


FIGURE C.6: Admittance parameters screen containing adjustable parameters of the control

File system

In this section, how to save experiment data and how this is stored is explained. First, on the Main screen of the app there is a 'Save experiement' button. This will save the experiment at any moment, even during a running experiment. However, this last practice must be avoided as much as possible, as it takes time for Matlab to process the saving and this could generate additional delays. Furthermore, the experiment is saved *every* *time* we stop it through the 'Stop experiment' button. This ensures to never miss any data.

Furthermore, when the app is opened for the first time it will create a folder named 'Admittance Experiments Data' in its current path, and every time is opened it will create a folder with the date, e.g. November26. After creating a subject session in the Main page of the app, this creates a folder for the subject with three subfolders: 'Blind Object', 'Force Tracking' and 'Functional Task' (this folder will contain any data recording during free control). Here, each time data is saved a new .mat file with data and time is created in the appropriate subfolder. See Fig. C.7 for an example.

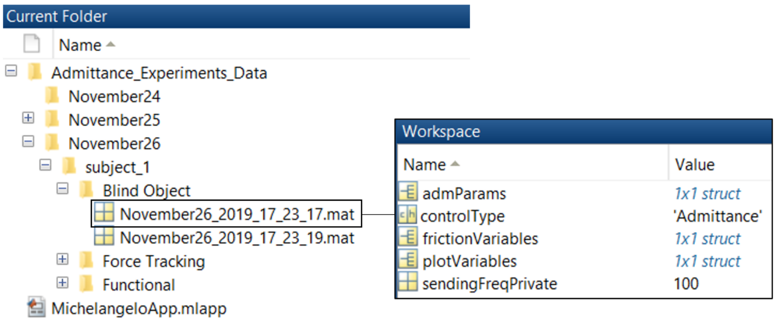


FIGURE C.7: File system of the control app.

Each .mat file contains all the relevant variables for analysis in form of structure files. The structure 'admParams' contains the value of all parameters that were used on each admittance control-based experiment, and 'frictionVariables' the friction values. The control strategy used in each experiment is detailed under 'controlType'. In 'plotVariables' we will find all relevant data: external forces (with and without applying any gain), predicted torques, virtual velocity, EMG data, etc. See Fig. C.7 for an example.

Appendix D

Admittance Model Equations

In this appendix, the details on the equations of the dynamics of the admittance model used in this work are provided. As explained in Chapter 2, the admittance model captures the biomechanics of the hand during interaction with the environment. The dynamics of the hand are simplified to model the dynamics of the index finger with an external opposing force. As shown in Fig. D.3, the index finger can modelled as a 1 DOF hanging pendulum. That is, we do not consider finger adduction and abduction for this model.

The motion of the pendulum is ruled by its forward dynamics, which are obtained from its equation of motion (EOM). The forward dynamics convert force into motion, while inverse dynamics extract force from motion. We are interested in how the balance between external forces and predicted forces by the musculoskeletal model move the pendulum. Therefore, we chose to model the pendulum on its forward dynamic equations.

The considerations to resolve the dynamics of the pendulum are the followings:

- The interphalangeal joints are considered welded, the thumb is fixed and the pendulum has only 1 DOF at the MCP joint.
- The rod of the pendulum is massless and is attached to a mass point.
- The external force is always perpendicular to the mass point.
- There is friction at the MCP joint. The friction model is the Stribeck friction model as described in Chapter 2.
- There is no gravitational force acting on the pendulum. The reasons for this are explained in Chapter 2.
- Due to the rotational nature of the pendulum, the physical variables are rotational too.
- θ defines the angular position and its time derivative, ω, angular velocity.

The torques accelerating the pendulum are:

- The predicted torque from the model τ_{pred}
- An inertial torque due to the point mass. Virtual inertia I_v is governed by the virtual mass m_v in the point mass (for a pendulum $I_v = m_v l_{rod}$).
- The friction torque at the MCP joint always opposes the motion of the pendulum.
- An external torque τ_{ext} opposes motion, where $\tau_{ext} = F_{ext} l_{rod}$

D.1 Friction model details

The Stribeck friction model is shown in Fig. D.1. The total friction of the Stribeck model is usually represented on its linear form as in D.1, where, F_f is the total friction force, F_c is the Coulomb friction, $F_{vis}(v)$ is the viscous friction dependent on velocity (i.e. damping effect) and F_s is the Stribeck force, v_s is the Stribeck velocity and δ_{v_s} controls the decay of the exponential. The *sign* function will determine the direction of the velocity and therefore the sign of F_{fr}.

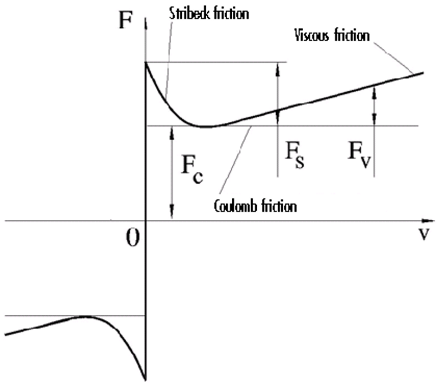


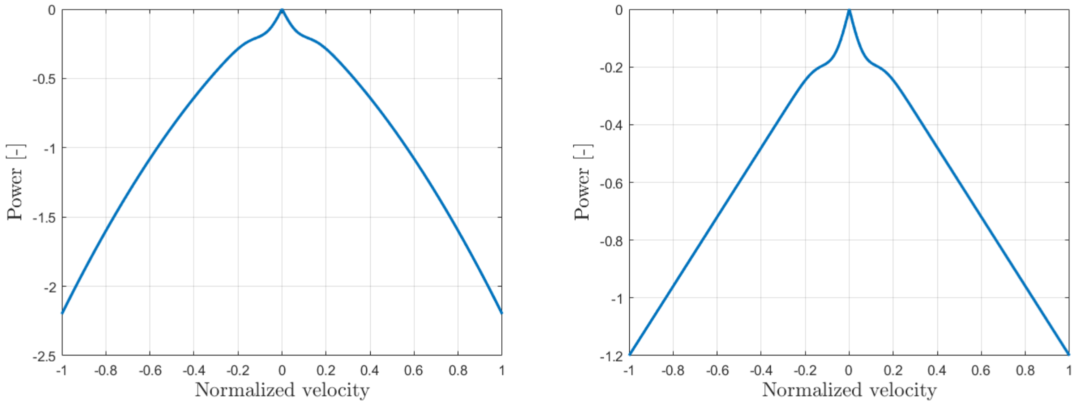
FIGURE D.1: Stribeck friction model

$$F_{fr} = \mathrm{sgn}(v)(F_c + (F_s - F_c)e^{(\frac{|v|}{v_s})^{\delta v_s}}) + F_{vis}(v) \tag{D.1}$$

However, due to the rotational nature of the pendulum, we deal with rotational units (i.e. torques, *angular* velocities, etc.). Therefore, a frictional *torque* must be opposing the angular motion instead of a linear friction force. If we consider the pendulum to be attached to a hinge joint, the area of contact is at a fixed distance R from the axis of rotation. Then there is no need to calculate the contact area and there is no need to consider that the friction force varies across the contact area.

On the other side, it could be interesting to separate the viscous friction from the rest of the friction model. A viscous element dependant on velocity dissipates energy. If we revise the passivity concept explained in Chapter 2, it is based on the amount of energy flowing in the system. If there is dissipative element which subtracts energy, it is interesting to see its effect on the passivity and the stability of the whole controlled system. We can see if we can isolate $F_{vis}(v)$ from the rest of the friction model if this is energy passive. If it is passive, with and without $F_{vis}(v)$, then no energy is generated and we are safe isolating $F_{vis}(v)$.

It is possible to demonstrate that the Stribeck friction model in D.1 is passive by checking if it generates any power for a range of normalized velocities. That is, if we consider the Stribeck friction model to be a power-conjugated system, where velocity is the input and the output is force, there should be no power generated at any moment of time considering all the parameters to be constant. Then, we can compute the power generated by multiplying the total output friction force F_{fr} by a range of normalized velocities. As seen in Fig. D.2a, no power is generated, but rather it is extracted from the system, due to the negative power values. However, this could be because the dissipative nature of the viscous friction. Nevertheless, by setting $F_{vis}(v)$ to zero we obtain the same conclusion: the friction model is passive also when no viscous friction is applied (Fig. D.2b).



(A) (B)

FIGURE D.2: Power profile of friction model with a) viscous friction b) no viscous friction

The viscous friction F_{vis} is given by $F_{vis} = \sigma_{vis} v$, where σ_{vis} is the viscous coefficient. Then we can consider σ_{vis} as the *damping* coefficient of our pendulum. In the reminder of the appendix we call the damping coefficient b_v, which is the *virtual damping* of the (virtual) dynamics of the pendulum. Again, as we are dealing with rotational units, we consider rotational damping and angular velocity, such that $\tau_{vis} = b_v \omega$.

D.2 Equations of Motion

After the some clarifications of the friction model used on the pendulum, the equations of motion can be computed. Note that we separate the damping force from the friction forces as explained above.

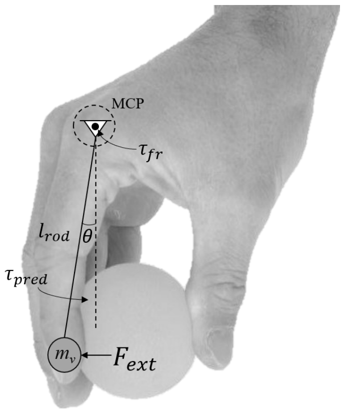


FIGURE D.3: Virtual index finger dynamics modelled as a pendulum.

The equations of motion for the presented pendulum are the followings, where we omit the dependence of the variables on time for clarity:

$$\tau_{ext} = F_{ext} l_{rod} \tag{D.2}$$

$$\tau_{ext} = \tau_{pred} + I_v \frac{d\dot{\theta}}{dt} - b_v \frac{d\theta}{dt} - \tau_{fr}(\dot{\theta}) \tag{D.3}$$

$$I_v \dot{\omega} + b_v \omega = \tau_{pred} - \tau_{ext} - \tau_{fr}(\omega) \tag{D.4}$$

The last equation (D.4) represents the virtual dynamics Y_v of the pendulum in time domain.

Rearranging (D.4) we get the following ODE representing the rate of change for the angular velocity of the pendulum, or **angular acceleration**:

$$\frac{d\dot{\omega}}{dt} = \frac{\tau_{pred} - \tau_{ext} - b_v\omega - \tau_{fr}(\omega)}{I} \tag{D.5}$$

$$\frac{d\dot{\omega}}{dt} = \frac{\tau - b_v\omega}{I} \tag{D.6}$$

By integrating (D.6) forward in time, the desired angular velocity ω_d of the prosthesis can be obtained. This can be done using Euler's method of numerical integration by following the general formula $y_{n+1} = y_n + hf(t_n, y_n)$, where h is the step size and $f(t_n, y_n)$ a function dependent on initial conditions. We can use the classic definition for velocity, based on Euler's method, and where ω_{di} is the initial condition and function f is substituted by (D.6) in (D.7) and (D.8) .

$$\omega_{d_{i+1}} = \omega_{d_i} + (\frac{d\dot{\omega}_{d_i}}{dt})dt \tag{D.7}$$

$$\omega_{d_{i+1}} = \omega_{d_i} + (\frac{\tau - b_v\omega_{d_i}}{I})dt \tag{D.8}$$

The Laplace transform of (D.6) is as follows:

$$\mathscr{L}\{\tau(t)\} = T(s) \tag{D.9}$$

$$\mathscr{L}\{\omega(t)\} = s\Omega(s) \tag{D.10}$$

$$T(s) = sI_v\Omega(s) + b_v\Omega(s) \tag{D.11}$$

$$Y_v = \frac{\Omega(s)}{T(s)} = \frac{1}{I_v s + b_v} \tag{D.12}$$

The last equation (D.12) represents the virtual dynamics Y_v of the pendulum model in the frequency domain.

In the Matlab GUI presented in Appendix C, (D.8) is used to compute the desired or reference angular velocities (ω_d) that are used as input for the controller of the prosthetic hand. Check the 'eom' function in the code view of the app.

Finally, the virtual position θ_d can be obtained in the same way as ω_d where θ_{d_i} is the initial condition:

$$\theta_{d_{i+1}} = \theta_{d_i} + \omega_{d_i}dt \tag{D.13}$$

Anti-windup mechanism

All actuators have physical limitations: the prosthetic hand cannot be more opened or more closed than what it *physically* can. For this same reason, the velocity reference ω_d can never command higher velocities that what the prosthesis drive can handle. For both reasons, the prosthesis would infinitely open or close, which is not possible. To prevent this, the virtual position θ_d is constrained to be always between the range of motion of the prosthetic hand (see B.2). When θ_d is greater or smaller than the upper and lower bounds of the range of motion of the prosthetic hand, ω_d is set to zero. Because θ_d is normalized, upper angular position bound is 1 and lower bound is 0.

Appendix E

Michelangelo hand's force sensor calibration

E.1 Overview

Michelangelo hand's embedded force sensor gives raw sensor values ranging from 0 to 100 %. As published by Ottobock, their prosthetic hand can achieve 70 N in palmar grasp, approximately. However, experiments carried out by the research community stated that it can deliver up to 78 $\pm$ 4 N . As external force measurement is key part of this work, reliable force readings are needed in physical units (Newtons), so that the whole admittance-based control system proposed is coherent on its units.

Because of these reasons, an experiment was carried out to find out which was the force range delivered by the specific Michelangelo hand model used for this work. The force readings of the Michelangelo hand were compared to the values given by an external load cell at different grasp levels. For this, the load cell was placed between the index finger and the thumb so that compressive forces were register in the point where the load cell and the embedded force sensor meet. After recording measurements from both the embedded force sensor and the load cell, a regression between the force values measured by both sensors was performed to get the coefficients of the fitted polynomial. This coefficients will map the force readings of the Michelangelo hand in percentage to Newtons.

The force sensor is embedded in the thumb and can only register forces in one direction. That means that if the contact is not fully perpendicular, the force sensor will not be that responsive. Therefore, a pin was attached to the load cell to ensure that forces were laying only the force sensor.

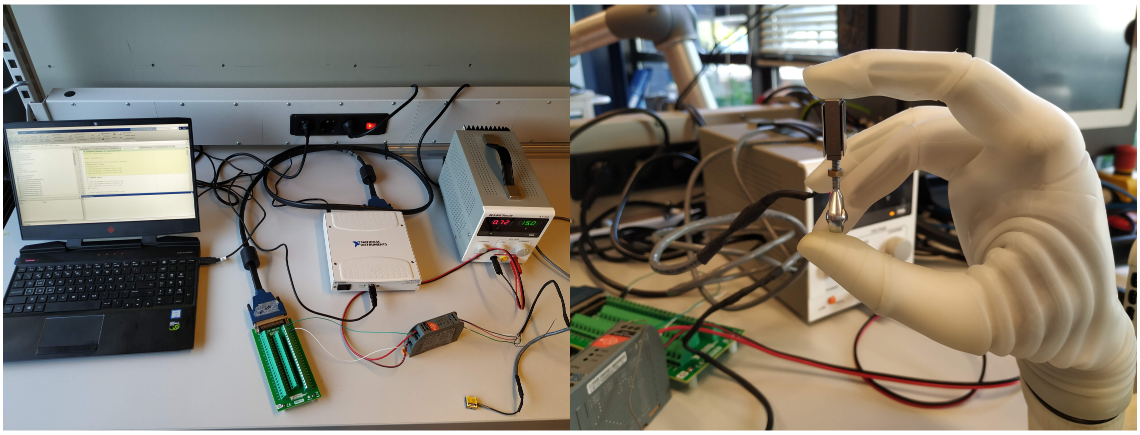
E.2 Calibration experiment

Set up

The set up used for calibration is shown in Fig. E.1. The materials used are:

- Isolated Strain Gauge Input Module SG-3016.
- Futek S-Beam Load Cell, model LSB200 50 lb (calibrated)
- Data acquisition device (DAQ) NI-USB-6259 (National Instruments) sampling at 2048 Hz (default).
- Laptop with Matlab and NI-DAQmx Support package from Matlab's Data Acquisition Toolbox.
- Power supply: BASEtech BT-305.

- Matlab programs: getDataLoadcellVSMickey.m, getLoadcellToMickeyCoeffs, voltsToNewtonsCoeffs.mat.



(A) (B)

FIGURE E.1: a) Materials and set-up for calibration and b) load cell placement in Michelangelo hand

The load cell used in this experiments was calibrated with known weights before any experiment with the embedded force sensor in the prosthetic hand was performed. The result of load cell calibration were the polynomial coefficients mapping the voltage read by the load cell to force in Newtons.

For the regression between Michelangelo force sensor values and load cell readings, 7 increasing force points were measured. The process and the Matlab programs used to perform Michelangelo hand's embedded force sensor calibration are the following.

Process

1. Put the load cell just between the thumb and index finger of the Michelangelo hand. For the first reading, the load cell should be barely squeezed to get low force readings.
2. Open Matlab and turn on the DAQ. Matlab should automatically recognise it. This can be checked by going to the 'Analog Input Recorder' app in Matlab.
3. Turn on power supply and put it at 15 V. Check the 'output' button is on, otherwise no current will flow through the circuit.
4. Read value of the load cell in Matlab with 'getDataLoadcellVSMickey.m'. The program makes 3 recordings of 2 seconds each. It computes the mean of *each* recording and, afterwards, the overall mean of the three means. The values are in volts.
5. Read the force value of the Michelangelo on the GUI, under the 'Grip force' tag and write it down. There is an overshoot, so it is recommended to wait few seconds for the value to decrease to a steady state.
6. Decrease the aperture of the Michelangelo hand through the GUI, or any other control program available. The force value should increase.
7. Repeat points 4 to 6 several times (e.g. take 7 measured points).
8. Use 'getLoadcellToMickeyCoeffs.m' to get the coefficients mapping force percentage values of the Michelangelo to Newtons. For this, put the force values from step 5 in the array `michelangeloForcePercentage`. The program will automatically load

the force measurements from the load cell if the correct Matlab path is specified. It will load 'voltsToNewtonsCoeffs.mat' to convert load cell voltage values to Newtons. Then it fits a polynomial for the force points obtained by the Michelangelo and the load cell. The output is an array `percentageMickeyToNewtonsCoeffs` containing the polynomial coefficients to map Michelangelo force percentage values to Newtons.

Outcome

If the experiment was successful, the plot of the fitting done by 'getLoadcellToMickeyCoeffs.m' should be a straight line, meaning that higher Michelangelo force percentage values linearly correspond to higher load cell force Newton values. For analytical results, the *goodness of fit* can be checked (output of the `fit` Matlab function). Check that the R^2 value is close to 1 and that the size of the root mean square error (RMSE) is coherent.

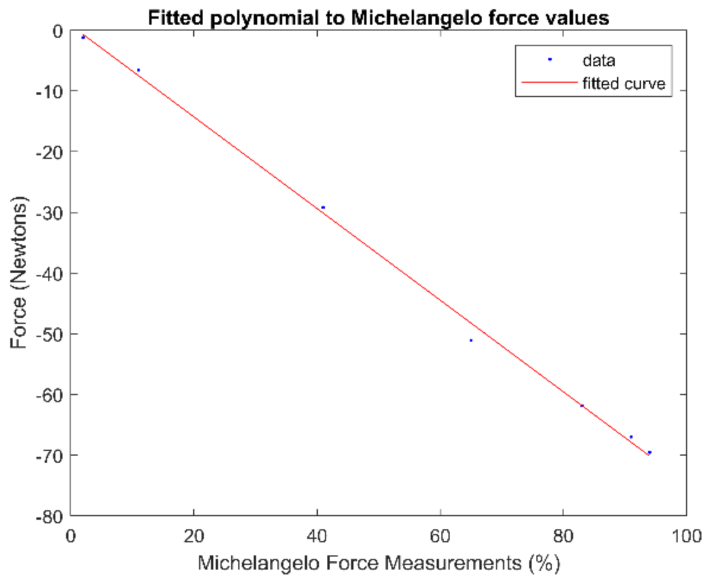


FIGURE E.2: Example of regression between Michelangelo force percentage values and load cell force Newton values.

Usage

The output coefficients can be used during online experiments to convert percentage values to Newton values as in E.1.

$$\text{force Michelangelo in Newtons} = coeff(1) * \text{force Michelangelo percentage} + coeff(2). \tag{E.1}$$

Appendix F

Prosthesis Dynamics Identification details

To check if the apparent dynamics Y_a are actually passive or not, and to assess the ez-width of the system, the dynamics of the prosthetic hand need to be identified. These dynamics contain the dynamics of the controller H_c and the plant H_p, as in Fig.

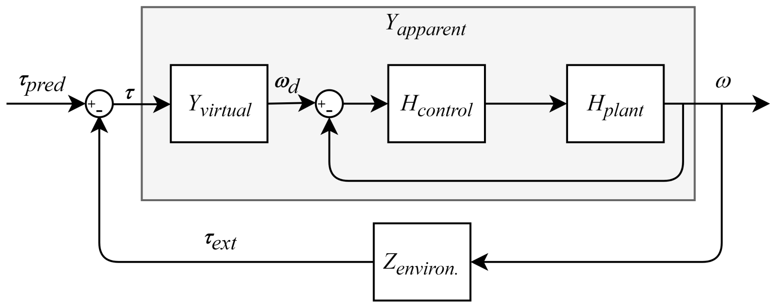


FIGURE F.1: Scheme of the apparent dynamics containing prosthesis dynamics.

Neither H_c nor H_p are known. Therefore, we can model prosthesis dynamics as a black box containing those two systems. Then, what we are identifying is the combination of H_c and H_p. We will call this combination H_{mic}, which stands for the overall transfer function of the **Mic**helangelo hand. Note that we are identifying a closed-loop system, however, we can *estimate* the closed-loop dynamics by identifying the open-loop dynamics.

A system identification process is based on perturbing the system with an input signal ($u(t)$) and comparing it with the output signal ($y(t)$). If the system is linear time invariant (LTI), the output should only change based on the frequency of the input signal. If the output varies with other parameters, such as the amplitude of the input signal, it will be a non-linear system.

F.1 Identification experiment

A system identification experiment was carried out with the Michelangelo hand to know its dynamics. The input signal ($u(t)$) was a multisine containing position commands. The multisine was generated with the provided function 'multisineGenerator.m'. Here, several parameters can be adjusted such as period length, excitation frequency limits, etc. By default, the excited frequencies are 80 frequencies ranging from 0 ot 5 Hz, with 6 periods of 20 seconds each. As explained in Chapter 2, 3 multisines with different amplitudes were used in 3 experiments. An amplitude of 10, 25 and 50 were used, which correspond to 20%, 50% and 100% prosthesis aperture.

The system identification experiment can be conducted with the 'sysidExperiment.m' program. For this, connect the prosthesis through the Michelangelo GUI. The program will automatically create a UDP connection. *Important: put the hand completely closed before starting each experiment*. The output ($y(t)$) of the experiment is the position of the Michelangelo hand.

F.2 Computing the frequency response function

A transfer function is defined by the ratio between the input and the output signal. If we want to get the transfer function of dynamics of the Michelangelo hand in the frequency domain, one way is to do it by transforming $u(t)$ and output $y(t)$ into the frequency domain variables $U(s)$ and $Y(s)$ with the Fourier transform. Therefore, we can compute the frequency response function of $H_{mic}(s)$ by taking the relation between $U(s)$ and $Y(s)$. Note that we consider no correlation between the input signal and any noise source in the system.

In the frequency domain, this is done by computing the relation between the cross-spectral density $S_{yu}(f)$ and auto-spectral density $S_{uu}(f)$, both frequency dependent. Spectral density is the Fourier transform of the time-domain cross-correlation. From this it can be inferred that we can get the response of a system by looking at the correlation between the input and the output. Therefore, we get the frequency response function (FRF) of H_{mic} as in F.1.

$$H_{mic} = \frac{S_{yu}(f)}{S_{uu}(f)} \tag{F.1}$$

In this work, spectral densities where computed through Welch averaging method, where frequency domain data was divided in segments, then the spectral density for each segment is calculated and finally the average of all segments is computed.

It is also interesting to compute the coherence γ^2_{yu}, which is similar to the correlation coefficient on the time-domain, and has a value between 0 and 1. A value of 1 indicates full correlation between input-output. Therefore, coherence indicates if two signals are linearly related. Coherence is reduced by additional signals (noise) and non-linearities. It is calculated as in F.2.

$$\gamma^2_{yu}(f) = \frac{\left|S_{yu}(f)\right|^2}{S_{yy}(f)S_{uu}(f)} \tag{F.2}$$

For the three amplitudes, the resulting FRFs are shown in Fig. F.2. The FRF and coherence of H_{mic} based on raw $u(t)$ and $y(t)$ is computed in the provided 'frequencyAnalysis.m' and 'estfrf.m'. Plots are made through 'plotFrfCoherence.m'.

From the output of the FRFs, we can see that the response changes at different multisine amplitudes. In linear systems, the output should only be affected by the frequency of the input signal, which is not the case. This means that the Michelangelo hand has non-linear dynamics. However, for the ease of the stability analysis, we consider linear dynamics, where the non-linear behaviour is left for further research. For parameter identification, the FRF of the amplitude 20% was used, so as to preserve the widest bandwidth the Michelangelo hand can deliver.

Furthermore, from the figures below, it can be seen that the coherency is smooth and very high in the whole bandwidth. This means that the signal-to-noise ratio is low and

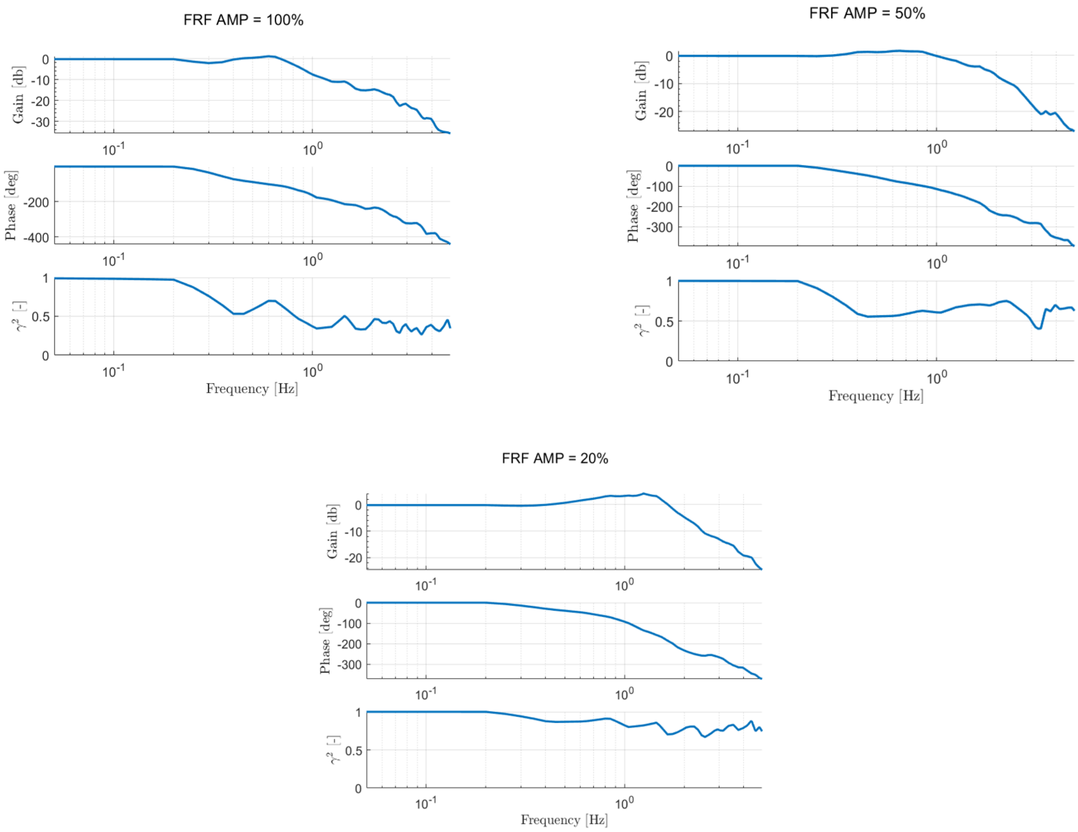


FIGURE F.2: Obtained FRFs for different amplitudes.

allows for a cleaner FRF response. The coherence reduces at high frequency where measurement noise has greater impact. Therefore, we can conclude that our FRF estimation was good enough and *coherent*.

F.3 Estimating the parameters of the FRF

To get the transfer function H_{mic}, the parameters of the curve of the identified FRF must be identified. This can be done through a least-squares routine in Matlab. This can more easily be done through the `tfest` Matlab function or through the system identification app in Matlab. The result obtained for H_{mic} after parameter identification is the following:

$$H_{mic} = \frac{3.43s + 60.65}{s^2 + 4.828s + 61.92} \cdot e^{-0.15s} \tag{F.3}$$

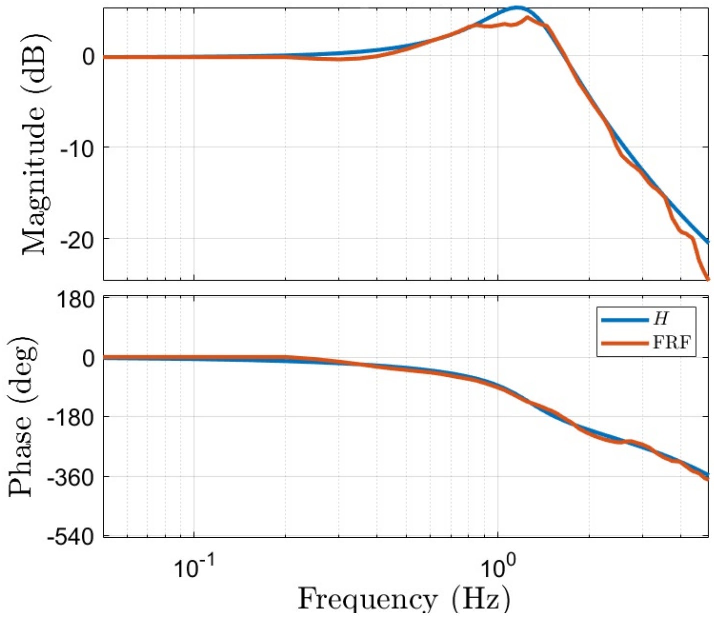


FIGURE F.3: Frequency response of identified H_{mic} and experimental FRF of amplitude = 20%.

The source for the explanations on this appendix was [32].

Appendix G

Stability Analysis

Coupling an admittance controlled device with external environment dynamics (Z_e) creates a negative force feedback loop (Fig. G.1). The stability of the coupled system depends of the dynamics of apparent dynamics Y_a and Z_e (see Fig. F.1). Because there is no knowledge of Z_e, one way of ensuring stability in our system is to ensure it is energy passive, as explained in Chapter 2.

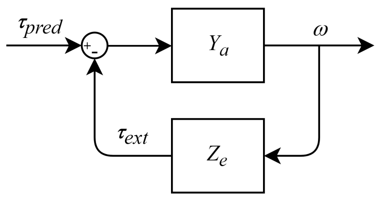


FIGURE G.1: Schematic of coupled dynamics of prosthesis and environment dynamics

However, passive behaviour of a controlled device cannot always be achieved due to weak dynamic performance of the controlled device. Then, Y_a can be stable but nonpassive. Y_a is composed of virtual dynamics Y_v and Michelangelo prosthesis dynamics H_{mic}. Y_v is designed to be passive, but, *is* H_{mic} *passive?*
A system is passive if *and only if* it is positive real, that is, H_{mic} will be passive iff:

1. $H_{mic}(s)$ has no poles in $\Re(s) > 0$
2. the phase of $H_{mic}(s)$ lies between $-90°$ and $90°$.

The phase plot of the identified dynamics of the Michelangelo, H_{mic}, shows an absolute phase lag greater than $90°$, therefore, it is *not* passive (see Fig. F.3). Then, the apparent dynamics Y_a will not be passive either, as Y_a is just the multiplication, in frequency domain, of Y_v with H_{mic}. If Y_a is not passive, *how do we check if the coupled system can actually be stable?* To answer this question, we can assess what *range* of environment dynamics Z_e could *complementarily stabilize* Y_a for a stable prosthesis-environment interaction by following the *Nyquist stability criterion* for closed-loop systems.

The closed-loop transfer function of the coupled system in Fig. G.1 is $Y_a/(1 + Y_a Z_e)$. From the Nyquist criterion we know that any interconnection of stable systems is always (marginally) stable if the absolute phase of the loop-gain (L) is smaller than $180°$. In our case, the loop-gain of the closed-loop transfer function is $L = Y_a Z_e$. Then, we can assess if the closed-loop system is stable by looking at the phase of the loop-gain. More specifically, an easy check for stability is to look at the phase margin: if it is positive, $|\angle L| \leq 180°$ and therefore, stable; if negative, we confirm that $|\angle L| \nleq 180°$, and therefore the closed-loop system is unstable.

Because real objects consist of inertias, damping and stiffness and these are passive elements, we can model Z_e as $Z_e = k_e + b_e/s$, where k_e and b_e will be a range of values of environment stiffness and damping, respectively. The range of values of k_e and b_e that complementarily stabilise Y_a is called *ez-width*.

The computation of the ez-width in this work is done in the program 'getEZwidth.m'. This program loads the identified transfer function H_{mic}, computes the loop-gain and assesses stability based on phase margin for a range of values of k_e and b_e. Finally, it builds the plots for the ez-width.

The sources for the explanations on this appendix are [33], [34].

G.1 A note for coherence

Note that the identified dynamics of the Michelangelo H_{mic} are identified for *position* input-output. That means that in our scheme (Fig. F.1) we should integrate ω_d into θ_d, and use desired (angular) position as input for the prosthetic controller. For coherence with identified H_{mic}, the output should be angular position θ. Therefore, we should differentiate output θ into ω, which will be the input for Z_e. However, this does not affect the loop-gain of the closed-loop transfer function and the ez-width analysis and results would be unaltered.

Furthermore, note that the input for Z_e is linear velocity and that the range of values for k_e and b_e is given in linear units and not rotational (see Chapter 2). To transform angular velocity ω into linear velocity v, one should multiply by the radius of the index finger (l_{rod}) such that $v = \omega\, l_{rod}$.