

POLITECNICO DI MILANO



Scuola di Ingegneria Industriale e dell'Informazione
Master of Science in Biomedical Engineering

Analysis and Optimization of Hannes prosthetic hand: a Multi-Body approach

Relatore:

Prof.ssa Elena De Momi

Correlatore:

Ing. Nicolò Boccardo

Dott. Andrea Marinelli

Dott. Federico Tessari, PhD

Autori:

Giulia Bruni
Matricola: 918437

Anna Bucchieri
Matricola: 916184

Anno Accademico 2020-2021

Aknowledgments

Ringraziamo la Professoressa De Momi per averci dato la possibilità di conoscere la realtà dell'IIT e per i suoi preziosi consigli.

Vorremmo ringraziare tutto il laboratorio di Rehab per aver reso questa esperienza di tesi spensierata e formativa, nonostante il periodo particolare in cui ci troviamo. Durante questi mesi non solo abbiamo avuto modo di imparare tantissime cose nuove, ma anche di sentirci parte di un ambiente creativo e stimolante.

Vorremmo ringraziare Lorenzo, Dario, Paolo e Sam per l'aiuto che ci hanno dato in questi mesi, Alessia per le chiacchierate e le risate in Acquario.

In particolare ci teniamo a ringraziare Nicolò per averci dato la possibilità di prendere parte al progetto Hannes e per la sua disponibilità nonostante i mille impegni.

Ringraziamo Andrea per la pazienza dimostrata soprattutto durante la stesura di questa tesi e per le battute sdrammatizzanti nei momenti di panico.

Ringraziamo Federico per averci seguito in ogni piccolo passo, per averci insegnato come affrontare i problemi durante il lavoro, per le frasi di incoraggiamento quando tutto sembrava non funzionare e l'infinita pazienza nei nostri confronti.

Vi ringraziamo soprattutto per aver sempre creduto in noi e nelle nostre capacità.

Abstract

Hand amputation greatly affects the ability of a person to perform Activities of Daily Living (ADL). For this reason, prosthetic hands should present anthropometric and functional characteristics to allow the manipulation of objects of different shapes and dimensions. These requirements are often difficult to translate in an ad-hoc design process of mechatronics components especially for a prosthetic application where multiple variables are very difficult to be studied and investigated from a bio-engineering point of view. This is the case of the Hannes prosthetic hand, an under-actuated myoelectric prosthesis where the actuation is provided by a single motor and the available sensors are particularly limited due to weight and cost constraints.

In such a scenario, models are a powerful tool to decompose, and hence approach, a very complex mechatronic system into smaller sub-assemblies which appear to be easier to study. Other very important aspects which induce researchers to invest their efforts toward a modelling process are the time and cost reduction during the design and prototyping phase.

The aim of our work was first the mechatronic analysis and optimization of the Hannes hand, and secondly, the implementation of novel control strategy capable of performing object stiffness recognition with the limited sensors information available on the device. In order to achieve these goals, a multi-body model of Hannes has been developed.

The multi-body model was realized in the Simscape environment, developing all the mechatronic components of the prosthesis, ranging from the mechanical components to the actuation and the control system. To verify the correct operation of the different subsystems, an analytical analysis was performed. The whole behavior of the model, instead, was validated by means of experimental tests performed on the actual device.

With this model, an estimation of the frictions, not quantifiable directly from the real prosthesis, was performed. A mechanical optimization of the mechanism responsible for the hand opening was proposed. Furthermore, the model was exploited to implement a stiffness-based objects recognition by using the only available information coming from the sensors already integrated within the Hannes system.

The results showed how the implemented multi-body model is comparable to the Hannes prosthetic hand in terms of both kinematic and dynamic behaviors, indicating this tool as a valid substitution of the real environment. Particularly, the relative error percentage between Hannes closure time and model closure time was of +0.79%, meanwhile the relative error percentage between the two in terms of force exerted by the fingers on grasped objects was -16.7%.

The proposed stiffness-based objects recognition implementation also provides promising results in terms of F1Score (>85%). These outcomes hence suggest a successful solution for a future real-time haptic feedback implementation directly on Hannes.

Sommario

Le persone affette da amputazione trans radiale presentano difficoltà nello svolgere attività della vita quotidiana. Per questa ragione le protesi di mano dovrebbero presentare caratteristiche antropometriche e funzionalità adatte per permettere la manipolazione di oggetti di diverse forme e dimensioni.

Questi requisiti sono spesso difficili da tradurre in un design specifico dei componenti meccatronici, specialmente nell'applicazione protesica dove studiare e investigare diverse variabili da un punto di vista bioingegneristico risulta complesso. Questo è il caso della mano protesica Hannes, una protesi mioelettrica sotto-attuata (l'attuazione è fornita da un unico motore) e sotto-sensorizzata a causa di limitazioni in termini di peso e costi.

In questo scenario, i modelli risultano essere un valido strumento per decomporre un sistema meccatronico complesso in sotto-assiemi che si rivelano essere più semplici da studiare. Altri aspetti importanti che inducono i ricercatori a sfruttare un processo modellistico sono la riduzione dei tempi e dei costi durante le fasi di design e prototipazione.

L'obiettivo del nostro lavoro è di analizzare e ottimizzare la meccatronica della mano Hannes e implementare una nuova strategia di controllo capace di riconoscere la rigidità degli oggetti sfruttando solo le informazioni fornite dai sensori già presenti nel device. Per raggiungere questi obiettivi abbiamo sviluppato un modello multi-body di Hannes.

Il modello multi-body è stato realizzato nell'ambiente Simscape, sviluppando tutti i componenti meccatronici della protesi, a partire dai componenti meccanici, al sistema di attuazione e il sistema di controllo. Per verificare la corretta implementazione dei diversi meccanismi abbiamo effettuato un'analisi analitica. Il comportamento complessivo del modello, invece, è stato validato tramite test sperimentali condotti sul dispositivo.

I risultati mostrano come il modello multi-body implementato sia comparabile alla mano protesica Hannes in termini di comportamento cinematico e dinamico, indicando come questo strumento sia un valido sostituto del vero device. In particolare, l'errore relativo percentuale tra il tempo di chiusura di Hannes e il tempo di chiusura del modello è del +0.79%, mentre l'errore relativo percentuale tra le due in termini di forza esercitata dalle dita durante una presa è del -16.7%.

L'implementazione del riconoscimento della rigidità degli oggetti proposta presenta risultati promettenti in termini di F1Score (>85%). Questo risultato suggerisce come questa soluzione sia interessante per lo sviluppo futuro di un feedback aptico real-time direttamente su Hannes.

Table of contents

Aknowledgments	i
Abstract.....	iii
Sommario	iv
Table of contents	vi
List of Figures	viii
Introduction	1
CHAPTER 1 State of Art.....	4
1.1 Prostheses overview.....	1
1.2 The role of the model.....	6
1.3 Multi-body models.....	10
CHAPTER 2 Materials and methods	15
2.1 Hannes	16
2.1.1 Overview	17
2.1.2 Hannes mechatronic components.....	21
2.1.2.1 Fingers.....	22
2.1.2.2 Differential mechanism	29
2.1.2.3 Actuators: theoretical background	30
2.1.2.4 Control strategy	35
2.1.3 Multi-body model implementation.....	38
2.1.3.1 From CAD to Simscape	43
2.1.3.2 Characterization of Simscape blocks.....	45
2.1.3.3 Fingers return mechanism	49
2.1.3.4 Master-slave system.....	55
2.1.3.5 DC motor	58
2.1.3.6 Double PID control.....	60
2.1.3.7 Frictions estimation.....	60

2.1.3.8 Object interaction	62
2.1.4 New implementation: Object stiffness recognition	65
2.1.4.1 Exploited sensors	66
2.1.4.2 Tool MATLAB: classification learner	70
2.2 Experimental set-up and validation	73
2.2.1 Validation metrics	77
CHAPTER 3 Results	80
3.1 Validation	81
3.1.1 Drop time	81
3.1.2 Return-wire mechanism	81
3.1.3 Closure time: Hannes vs Model	82
3.1.4 Motor current: Hannes vs Model	83
3.2 Optimization	87
3.2.1 Torsional springs	87
3.2.2 Force implementation	88
3.3 Application	90
3.3.1 Classifier	90
CHAPTER 4 Discussion	93
CHAPTER 5 Conclusion	97
Bibliography	100
Appendix I	104
Appendix II	108
Appendix III	111
Appendix IV	114
Appendix V	116

List of Figures

Figure 1 Hand prostheses available on the market.....	2
Figure 2 (On the left) SoftHand, (on the right) Hannes	3
Figure 3 Scheme of haptic feedback implementation	4
Figure 4 Explorative model scheme workflow.....	7
Figure 5 Simulation of the robotic head following the object displacement.....	8
Figure 6 MPPS CAD model exploited for FE simulation	9
Figure 7 Example of two link robot.....	12
Figure 8 Humanoid robot.....	12
Figure 9 Grasp performance of a four fingers robotic hand	13
Figure 10 DoFs comparison between Hannes and the real hand.....	16
Figure 11 Gloved device performing ADLs:	17
Figure 12 Close up on Hannes socket and Hannes hand.....	18
Figure 13 Hand, Wrist and Stump scheme	18
Figure 14 Control scheme:.....	19
Figure 15 Supplied accessories:.....	19
Figure 16 Hannes Master-Slave system for the control of the fingers movement.....	21
Figure 17 Finger structure:.....	23
Figure 18 Finger scheme:	25
Figure 19 Different thumb configurations	26
Figure 20 Hannes thumb:	27
Figure 21 Thumb scheme:	28
Figure 22 Differential mechanism scheme.....	29
Figure 23 ON/OFF actuator scheme.....	30
Figure 24 Actuators	31
Figure 25 Diagram of Brushed DC motor vs Brushless DC motor	31
Figure 26 DC motor scheme.....	32
Figure 27 Examples of PWM:	33
Figure 28 Motor Drive H-Bridge circuit schematic.....	34
Figure 29 H-Bridge modes of operation	34
Figure 30 (On the left) Brushless motor, (on the right) Brushed motor	35
Figure 31 Magnetic encoder scheme.....	36
Figure 32 Hannes Control and Driver scheme:.....	37

Figure 33 Solver configuration block.....	38
Figure 34 World frame block.....	39
Figure 35 Mechanism block.....	39
Figure 36 Rigid transform block.....	39
Figure 37 Revolute joint block.....	40
Figure 38 Solver profiler interface.....	43
Figure 39 Complete CAD model of Hannes:.....	44
Figure 40 Links and joints.....	45
Figure 41 Hannes Multi-Body model in Simscape:.....	46
Figure 42 Hannes Simscape: Blocks:.....	46
Figure 43 Double-inverted pendulum:.....	48
Figure 44 Compression spring block of the “Forces and Torque” library on Simscape.....	49
Figure 45 “Belt and Cables” Simscape library.....	50
Figure 46 Forces and moments acting on the middle part of a finger.....	51
Figure 47 Return-wire simulation.....	52
Figure 48 Torsional spring.....	53
Figure 49 Torsional spring in simulation.....	53
Figure 50 Applied force profile.....	54
Figure 51 master-slave system in simulation.....	55
Figure 52 Prismatic joint block.....	56
Figure 53 Slave-wire Simscape implementation.....	56
Figure 54 Example of a pulley system:.....	57
Figure 55 Scheme of Forces implementation:.....	57
Figure 56 DC motor model.....	58
Figure 57 Realistic control scheme.....	59
Figure 58 Simplified model of motor and PID controller system.....	60
Figure 59 Example of motor current.....	61
Figure 60 Blocks from Multi-body Multiphysics library:.....	61
Figure 61 Variation of the friction torque wrt angular velocity:.....	62
Figure 62 Interaction between Base Frame and Follower Frame.....	63
Figure 63 Solid Contact Force block.....	63
Figure 64 (On the left) ball, (on the right) hand dynamometer.....	64
Figure 65 Hannes pilot at Cybathlon 2020.....	65
Figure 66 Hannes motor current: Soft objects (blue line) vs Rigid object (red line).....	66

Figure 67 (Up) Graph of RefPos vs Encoder output, (Down) Hannes motor current:	67
Figure 68 RefPos vs Encoder output: Void closure	68
Figure 69 RefPos vs Encoder output: Stiff object	69
Figure 70 RefPos vs Encoder output: Soft object	69
Figure 71 Training Dataset example:	70
Figure 72 Experimental Set-up:	73
Figure 73 Testbench:	74
Figure 74 Two-cuffs system.....	74
Figure 75 Viener hand dynamometer	75
Figure 76 GUI Vernier Hand Dynamometer.....	75
Figure 77(On the left) Hannes and testbench, (on the right) Position of Hannes thumb	76
Figure 78 Object employed for testing the classifier	77
Figure 79 Comparison between reference input, Hannes and Model encoder outputs	83
Figure 80 Void currents:.....	84
Figure 81 Normalized current scheme	85
Figure 82 Forces exerted by fingers	86
Figure 83 Normalized angular excursion RW vs TS scheme.....	87
Figure 84 Normalized Angular excursion pulleys vs force implementation scheme.....	89
Figure 85 Loose wire	94
Figure 86 Index, middle Drop time.....	107
Figure 87 Ring, little Drop time	107
Figure 88 Index, middle RW vs TS	108
Figure 89 Ring, little RW vs TS.....	109
Figure 90 Thumb RW vs TS, applied force	110
Figure 91 Index, middle pulleys vs Fext.....	111
Figure 92 Ring, little pulleys vs Fext	112
Figure 93 Thumb pulleys vs Fext, Force applied	113
Figure 94 Current rigid object	114
Figure 95 Current soft object	115
Figure 96 Classification (1)	117
Figure 97 Classification (2)	117
Figure 98 Classification (3)	117
Figure 99 Classification (4)	117
Figure 100 Classification (5)	117

Introduction

Technological and academic developments have introduced poly-articulated prosthetic hands to the world with particular attention to amputees' case of study. The aim is to develop a poly-articulated prosthetic hand capable to reproduce human-like grasping behaviours. Nevertheless, there is still an important gap between amputee desires and what a prosthetic system can offer to the patient. For this reason, prosthetic users are not fully able to perform most of the activities of daily living. Thus, prosthesis abandonment rate remains high due to poor embodiment.

The project specifications, divided in three main subsystems: anthropometric features, human-like grasping and biomimetic performance, make the design process a long-standing challenge [1]. Therefore, the main mechatronic components sizing process needs to be in deep investigated. This requires a precise identification of all the involved variables, which some of them often appears to be difficult to measure and hence compensate.

Poly-articulated prosthetic hands, as all the complex systems, need to be analysed in smaller sub-assemblies to be completely understood and optimized.

Sometimes, designing a model after the prototype is already realized, still represents a powerful tool to make the observed phenomenon easier to understand [2]. The reverse engineering approach allows to investigate aspects in different context ranging from an optimization process to an industrialization phase. The model hence allows researchers and engineers to realize and test new device implementations according to the user needs. Once an implementation leads to promising results, the system could be modified according to its related model [3].

Furthermore, multi-body models are very promising strategies when the computation of the kinematic and dynamic analysis of complex systems is required. For this reason, the differential equations of motions and forces need to be computed to study how variables and parameters interact in a physical environment. A multi-body computational software allows to automatically integrate them with the purpose of simulating the systems behaviour guiding the developer to a more automated parameters optimization.

In this thesis we hence proposed a multi-body model of the Hannes prosthetic hand with the aim to study non-measurable variables on the real device, to optimize mechanical components and to test new artificial intelligent software implementations.

Specifically, the work focused on modelling the Hannes prosthetic hand. Hannes is a poly-articulated myoelectric prosthesis capable of restoring over 90% of functionality to upper limb amputees. Hannes was realized through a collaboration between Istituto Nazionale Assicurazione Infortuni sul Lavoro (INAIL) and Istituto Italiano di Tecnologia (IIT).

The poly-articulated prosthetic hand can automatically adapt to any object of any shape allowing 95% of the grasping possibilities. This is possible by means of its embedded differential mechanism which couples a master wire, connected to the power train, to two slave wires, running within the fingers in a tendon-driven custom mechanism [4].

The Hannes multi-body model might support the related research activity to estimate the non-measurable frictions affecting the real system. Afterwards, we propose an alternative return-wire mechanism, fundamental for the reopening of the hand, in order to save space and weight on the device.

In conclusion, a new method to recognise the hardness of grasped objects will be proposed by using the model without any additional force sensors employment.

The thesis thus is organised as follows:

- CHAPTER 1 offers an overview of the State of the Art of prosthetic hands and their limits, with a focus on the haptic feedback implementation, then we proceeded with a brief introduction to the importance of creating a model, and lastly, we focused on the utility of multi-body models.
- CHAPTER 2 presents the materials and methods used to develop this work, in particular:
 - The mechatronic components of Hannes prosthesis;
 - The multi-body model of Hannes;
 - The object-stiffness recognition implementation;
 - The experimental set-up used to acquire data and the validation metrics employed.
- CHAPTER 3 collects the obtained results, in particular:
 - The results obtained to validate the model;
 - The results related to the optimization of mechatronic components;
 - The results of the new application.
- CHAPTER 4 presents a critical discussion of obtained results.
- CHAPTER 5 provides the conclusive remarks and future developments.

CHAPTER 1 State of Art

In this section firstly we reviewed the State of Art of trans-radial prostheses and their limitations (see 1.1), then we proceeded with the importance of creating a model (see 1.2) and lastly we introduced Multi-body models in the paragraph 1.3.

1.1 Prostheses overview

The hand is fundamental for the interactions of humans with the world. It allows to manipulate objects of different shapes and dimensions by assuming several different configurations and applying the right amount of force thanks to its physiological tactile feedback.

The combination of the human hand great kinematic and dynamic complexity, together with its multiple way of utilization, still haven't found an adequate substitute in prosthetic devices.

The most important skills, that a prosthetic hand should have, are:

- High number of degrees of freedom (DoFs)
- Different functionalities
- Precision
- Low weight
- Reliability

Most of these requirements are not fulfilled, or partially fulfilled, by the available prostheses on the market and in research. Consequently, the majority of unilateral amputees prefers to use the healthy limb to perform most of their activities of daily living (ADL) [32].

Amputees have several types of devices among which they can choose, ranging from passive prostheses, such as cosmetic hands (Figure 1 a)), whose function is only aesthetic, to active ones which require power supplies and actuators with different levels of complexity.

Active prostheses restore partially the functionality of a human hand. They are mainly based on myoelectric control which allows the patients to interact with the world and manipulate objects directly controlling their prosthesis through the residual muscle contractions. This is possible thanks to the superficial electromyography (sEMG) which involves the application of non-invasive superficial electrodes on the patient's stump to record the muscles' activity. A wide variety of control scheme are currently used to translate the information given by the EMG signal.

Currently, the on/off control, based on EMG thresholds of two antagonist muscles (Flexor and Extensor of the wrist, typically), determines the hand movement in the open or close direction. This is the most used strategy in ADL scenario, even if most advanced methods are developed to control multi-DoFs hand prostheses. The pattern recognition strategy, for example, employs machine learning algorithms to find a correlation between the EMG signals and the desired hand pose [33], [34].

The choice of the right strategy depends on the number of DoFs to control.

In the past, devices with a single DoF were the most common. They are characterized by a single motor able to move two fingers simultaneously with the thumb to perform the tri-digital grip like the Ottobock SensorHand Speed [35] in Figure 1 b).

However, the evolution in research has led to the introduction of poly-articulated hands with multiple DoFs for more human like movements. These prostheses can be classified in two categories: fully actuated and under actuated.

A mechanism can be defined fully actuated if an actuator is present for each DoF like in the case of the i-Limb [36] (Figure 1 c)) and of Bebionic [37] (Figure 1 d)) which are characterized by the presence of five individual powered digits, one for each finger.

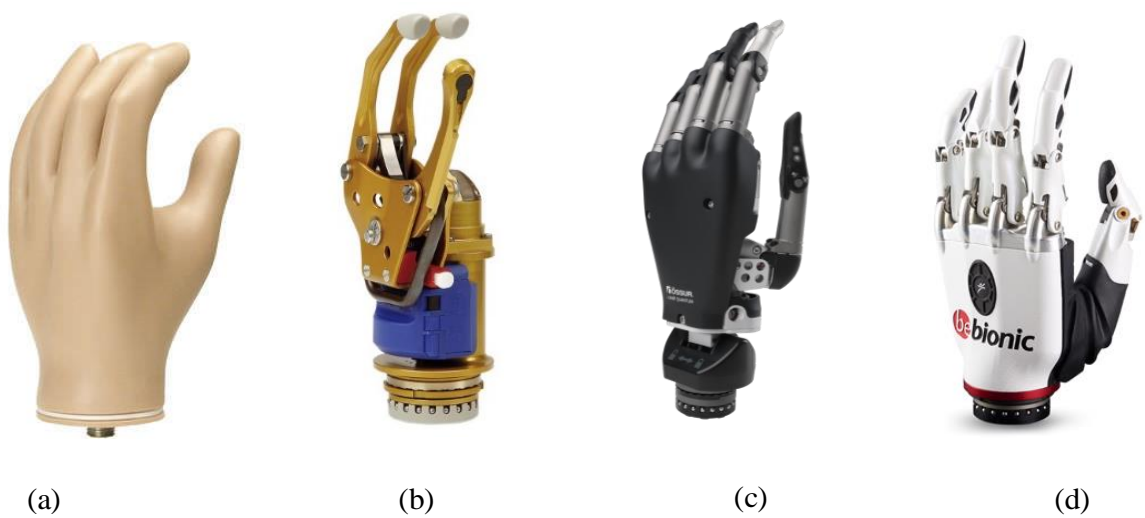


Figure 1 Hand prostheses available on the market

a) Aesthetic glove, b) Ottobock sensor Hand Speed, c) i-Limb, d) Bebionic

Although fully actuated systems seem to be performant, they present some disadvantages regarding weight, cost and complexity in the control [38]. Consequently, under actuated systems might represent a valid alternative where the actuators are less than the number of DoFs and they are substituted by passive elastic elements and mechanical stops, which create a mechanism allowing adaptive grasps more similar to the human ones [39]. Two examples of this type of system are the SoftHand [40] and the Hannes Hand [4] reported in Figure 2.



Figure 2 (On the left) SoftHand, (on the right) Hannes

Another important skill that a prosthetic hand should have is the tactile sensation.

The haptic feedback, defined as the tactile sense, is a powerful source of observations facilitating the refined control of the hand movements when in contact with a surface. In particular this kind of feedback is provided when the hand applies a pressure on a side of an object, while another part of the object is either stabilized or exerts an opposing force. [41]

When using a myoelectric prosthesis, the user relies solely on visual and auditory information [42]. This reduces the feeling of connection with the non-physiological hand and brings the subject not to be engaged in active grasping and exploration. [43] Some studies reported how the ability to control a prosthesis would be considerably improved for the user if a haptic feedback would be provided to them, leading to a faster and easier acceptance of the prosthetic hand [43-47].

Different studies propose the implementation of a tactile sensation by employing force sensors coupled with vibrating units.

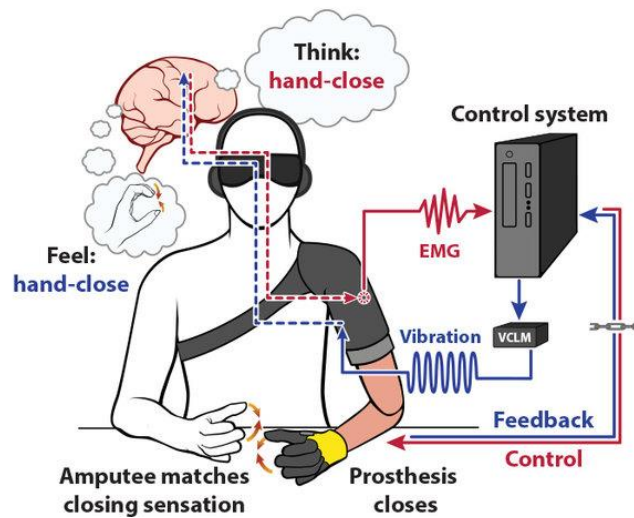


Figure 3 Scheme of haptic feedback implementation

Chatterjee et al. [43] developed a haptic feedback simulator integrated with visual information. Their study demonstrated how such simulator helps the prosthesis user to better control the device leading to an improvement of functionality and comfort. The user study contemplated eight able-bodied subjects who wore the prosthetic hand over their limb, simulating the conditions actual amputees would face.

The technology exploited to sense the interaction between the prosthesis and the object was a force sensor made by strain gauges mounted at carpal-metacarpal thumb joint where maximum bending moment is observed; meanwhile for the vibrotactile feedback it was used a C2 factor mounted to the upper arm with an elastic band. Overall, the study suggested that haptic feedback was not perceived with the same resolution of the visual feedback. However, it was interesting how in the first repetitions of interaction the subjects controlled the hand grasp almost relying solely on visual feedback meanwhile, on the last trials, they tended to rely to the more unfamiliar vibrotactile system [43].

Another study conducted by Pylatiuk et al. [44] aimed at demonstrating the effectiveness of the force feedback system in myoelectric prosthesis operations without the help of the vision system. Five habitual myoelectric prosthesis users were involved for the user study. Their task was to grasp, hold and lift a hand dynamometer with a five-finger pinch grip with and without vibrotactile feedback application.

The results of their experiments suggested how subjects provided with haptic feedback exerted a lower grasping force with respect to subjects without the feedback, in particular the grasping force reduced by 30% to 77% when vibration was applied directly to the skin of the residual limb.

In conclusion a refined control of a prosthetic hand, along with a realistic sensation by means of a haptic feedback, is still a very complex reality that needs to be further investigated [46]. However, the powerful contribution to prosthetic users, provided by the addition of such information, has been largely confirmed. In particular, according to Sensinger et al. [42], the beneficial aspects of haptic feedback are mostly present in new prosthetic users, meanwhile subjects who already have experience with myoelectric prostheses don't seem to appreciate it as much.

In this thesis we focused our studies on the prosthetic hand Hannes, explained in detail in the paragraph 2.1. Hannes is an underactuated prosthetic hand characterized by a complex mechatronic structure of difficult investigation. For this reason, we employed a modelling approach to analyse the device and test a specific method to recognise the stiffness of a generic object when interacting with the hand prosthesis. The proposed strategy can be further exploited in the implementation of haptic feedback, not currently present in Hannes.

1.2 The role of the model

The word 'model' originates from Latin *modulus* which means measure, pattern, example to be followed.

It is defined as the representation of something, often idealized or simplified, in order to make the observed phenomenon easier to understand [5]. It captures not all attributes of the represented system, but rather only those seeming relevant. [6]

The main criteria of a model should be [7]:

1. Mapping criterion: the model should represent a real object or phenomenon;
2. Reduction criterion: the model should at least present one of the properties of the real object;
3. Pragmatic criterion: the model should be a tool useful for prediction, problem-solving and optimization of the object.

Modelling an object or a process helps not only in the development and improvement of the system design but also in finding several solutions operationally, economically and environmentally speaking.

The model is a powerful tool to decompose a complex system into smaller parts, easier to understand ('reductionist approach') [2], in order to facilitate the decision-making process in how the system will be deployed.

Given a theoretical hypothesis, the idea is starting from a given target, create a related model to investigate its characteristics, and demonstrate such hypothesis.

The info extracted from such analysis are converted into claims about the target system [7].

Models can be classified in different ways according to various factors. The relationship between the target and its model can be *descriptive*, meaning that the model reflects the exact characteristics of a target, *prescriptive*, where the model provides a specification of something to be created, and *transient*, which represents a descriptive model enriched with new features.

Another way to classify a model is its purpose. *Educational models* are a representation of a target for ethical or practical reasons; for example human body models are used by medical students to better understand anatomy [8] or flight simulators are used to train pilots [9]. These are descriptive model that imitate the original reference.

Another class is the *Formal model*. Mathematical models belong to this category: they are descriptive models which don't resemble the reality they describe but allow to analyse phenomena of the real world. The use of mathematical models has proven to be very efficient in improving and accelerating the development of lower extremity neuroprostheses

[2]. Particularly for the prediction of specific outcomes of the system [10] or for the optimization of open-loop and closed-loop controllers [11].

Lastly *explorative models* are transient models used when the consequences of a change are to be evaluated. Rather than applying a change in the real system, the model is modified according to the need. If the changes lead to positive results, the modifications are then implemented into to the original system [7].

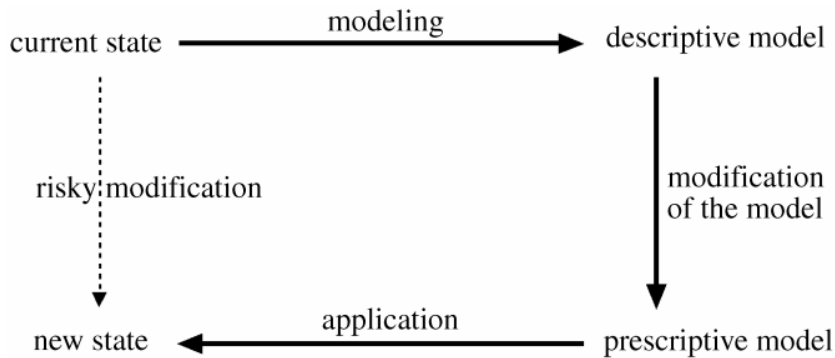


Figure 4 Explorative model scheme workflow

Conventional engineering transforms concepts and models into real systems. The power of such approach is particularly appreciated when large amounts of capital is to be invested. Indeed, the model represents an economic function reducing the financial loss as much as possible [3].

An example of model-based development of a system is the simulation head developed by Muthugala et al. [12]. The research team designed a MATLAB model of an interactive robot head with human-like movements and human-friendly interactive features (Figure 5).

Thanks to the model, it was possible to study the neck actuators movements to design a proper control strategy to drive the interactive head. This was implemented by tracking the movements of a human hand.

Once tests and analysis on the model were performed, a prototype was created.

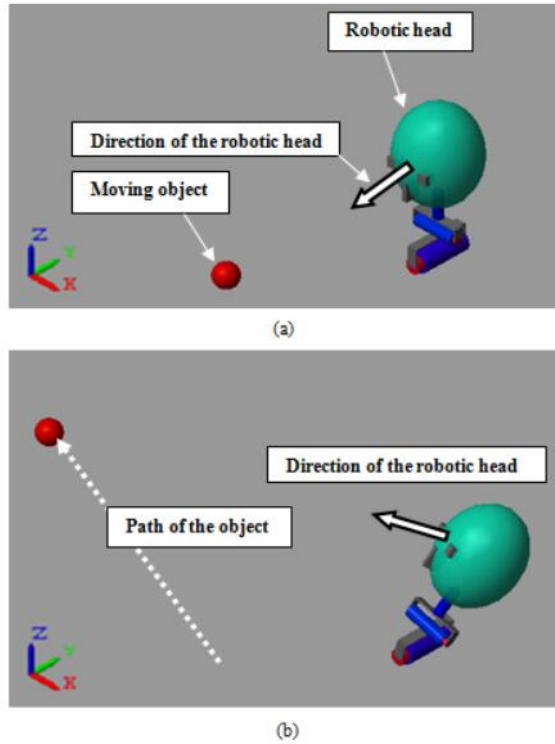


Figure 5 Simulation of the robotic head following the object displacement.

Another interesting purpose of models is to provide a tool for measuring given quantities which otherwise wouldn't be easily derived from the real system. In the study conducted by De Lorenzo et al. [13] the purpose was to perform an analysis on intraoperative forces and moments on patient head clamp during awake brain surgery. Due to unavailability of the Mayfield Patient Positioning System used to restrain the head of the subject, a CAD model of MPPS was created on which FE analysis was performed in order to find the best configuration to instrument the real system.

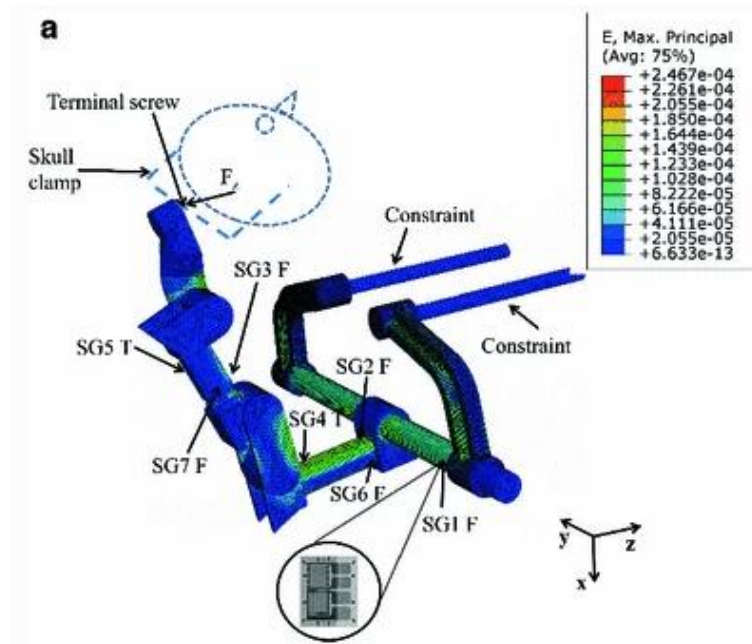


Figure 6 MPPS CAD model exploited for FE simulation

Sometimes reverse engineering real objects into models has proven to be useful too. This approach starts from measuring an existing object so that a model can be created exploiting the best features of the real object [14]. Dallali et al. [15] developed a simulator for the humanoid robot COMAN including the dynamic characteristics of the system and realistic ground models. The simulator was used to test and develop new control strategies for the joints and walking task.

In our thesis we decided to focus our studies on multi-body models.

1.3 Multi-body models

A multi-body system is an assembly of rigid or flexible bodies connected through kinematic joints and subjected to force elements.

A body can be considered rigid if its deformations are small enough not to influence the global motion produced by the body, thus it can rotate and translate without changing its shape. Instead, a flexible body has an elastic structure which allows a certain deformability. In the three-dimensional space the rigid body motion can be described by 6 generalized coordinates, three linear positions and three angular positions, one for each DoF, while flexible bodies require additional coordinates to describe the deformations as well.

The kinematic joints constrain the relative motion of the bodies connected to them reducing the number of Degrees of Freedom (DoFs) of the system. The number of constraints depends on the type of joint. The most common are prismatic and revolute joints which allow respectively one translation and one rotation with respect to a given axis.

Force elements applied on the multi-body systems can be the result of springs, dampers, actuators or external forces. In particular, the external forces are commonly used to model the system interaction with the environment.

Multi-body modelling can be used to study the kinematic and the dynamic characteristics of the movement of a wide variety of systems [16], [17].

However, in order to perform the kinematic and dynamic analysis of complex systems, thus finding how the bodies move together and the forces generated in the process, it is necessary to write and integrate the differential equations of motion. Due to the complexity of these equations, several multi-body computational software have been introduced to automatically generate and integrate them with the purpose of simulating the systems behaviour and finding possible parameters optimizations.

The differences between existing multi-body computational programs, such as ADAMS [18], SIMPACK [19], DADS [20], stand in the principle used (e.g., principle of virtual work, principle of virtual power, Newton–Euler’s approach), the type of coordinates (e.g., Cartesian coordinates, Lagrangian coordinates), the type of approach used in the generation of the equations of motion (e.g., numerical or symbolical) and the selected method to handle the kinematic constraints (e.g., coordinate partitioning method, augmented Lagrange formulation). In the case of constrained multi-body systems the Lagrange multipliers technique can be used as a solution, which leads to the generation of differential and algebraic equations (DAE) solved frequently by converting them into ordinary differential equations (ODE) [21], as it is done by most of the Simscape Multi-body solvers. For further information look at the paragraph 2.1.3.

The inputs required by multi-body programs are [22]:

- Rigid bodies (mass, centre of mass, inertia tensor);
- Constraints (joints, prescribed motions);
- Forces (springs, dampers, frictions, contact forces, gravity, external forces);
- Initial conditions (initial position, initial velocity, initial orientation, initial angular velocity);
- Integration parameters (choice of solver, step size, tolerance, simulation time).

While the data given as output are:

- Motions of rigid bodies (position, velocity, orientation, angular velocity);
- Joint reaction forces;
- Other forces.

In this way multi-body programs allow the simulation of mechanical systems with multiple DoFs interconnected with each other. Some of them, like Simscape, also integrate the characteristics of multiphysics software, thus allowing the interaction between systems ruled by different physical principals (e.g., hydraulic, electronics, mechanics, etc.).

Moreover, the most advanced programs provide useful tools such as plots, graphical animation, signal analysis and CAD connectivity.

As a consequence, they can be considered a valid resource in different application fields like automotive (e.g., for the optimization of vehicles suspension design [23]), biomechanics (e.g., for the reconstruction of accidents injuries [24], [25]) and aeronautic (e.g., for wings design [26]).

Specifically in robotics, our field of interest, in literature there are several applications employing this type of software, ranging from the simplest to the most complex ones.

An example of a simple application is the one reported in L. Ángel's study, where a manipulator with two DoFs was realized through the multi-body program ADAMS with the aim of finding the appropriate control strategies for the motors present at the level of the two joints [27]. In this case, since the structure was very simple, it was possible to design the parts and the two joints, composing the manipulator, directly by using ADAMS blocks.

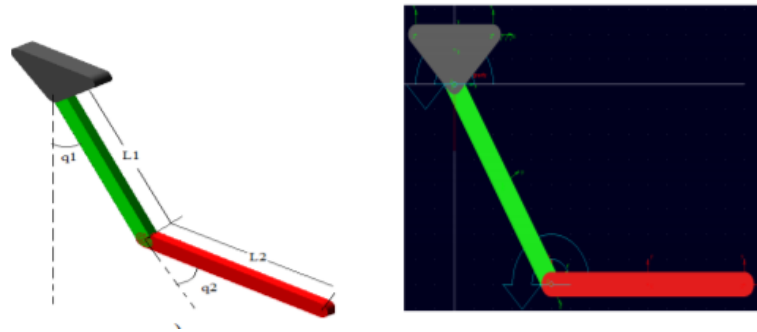
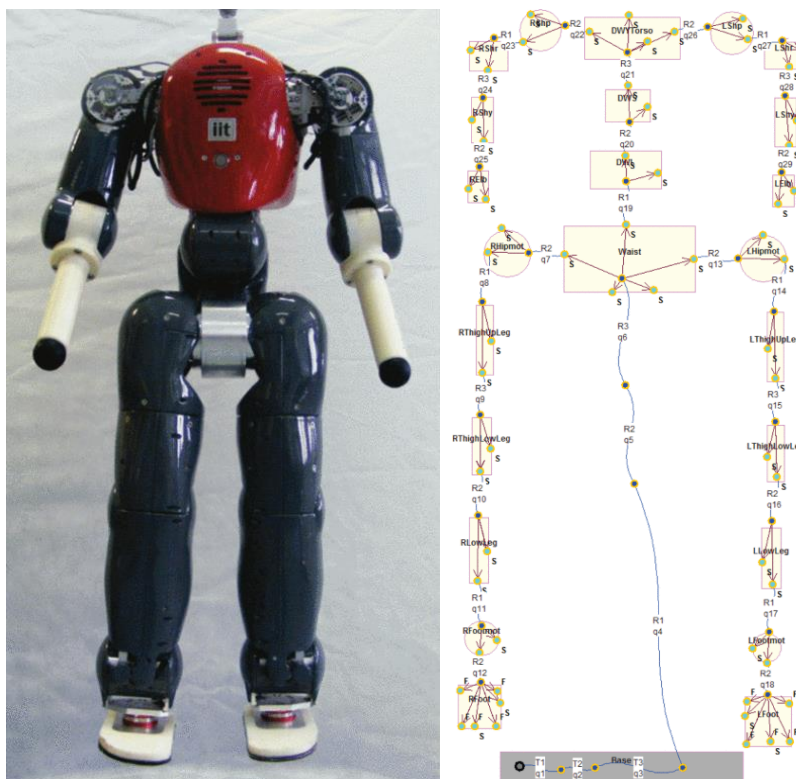


Figure 7 Example of two link robot.
 (On the left) mechanical structure, (on the right) ADAMS model

However, in other cases in which the structures are more complex, it is more convenient to import the CAD model whose parts are directly converted in multi-body blocks in this way. This is the strategy adopted in the realization of the IIT humanoid robot C-cub model, which was developed through the symbolic multi-body software Robotran [28] to design, test and compare different control systems [29]. Robotran resulted also to be a useful tool in simulating the interaction of a robot with the ground allowing for example to tune walking trajectories for 15 DoFs of the COMAN robot, the humanoid robot derivative from the iCub and cCub [15].



The aspect related to the interaction with the environment in multi-body models was investigated with other types of robots as well. For instance, the multi-body program SimMechanics [30] (Simscape previous version), was employed to simulate the grasp performance of a four fingers robotic hand. The different angular motions, joint velocities, accelerations, and torques were measured from the model, interacting with three cylindrical objects and a sphere, in order to evaluate the stability in grasping objects of different size [31].

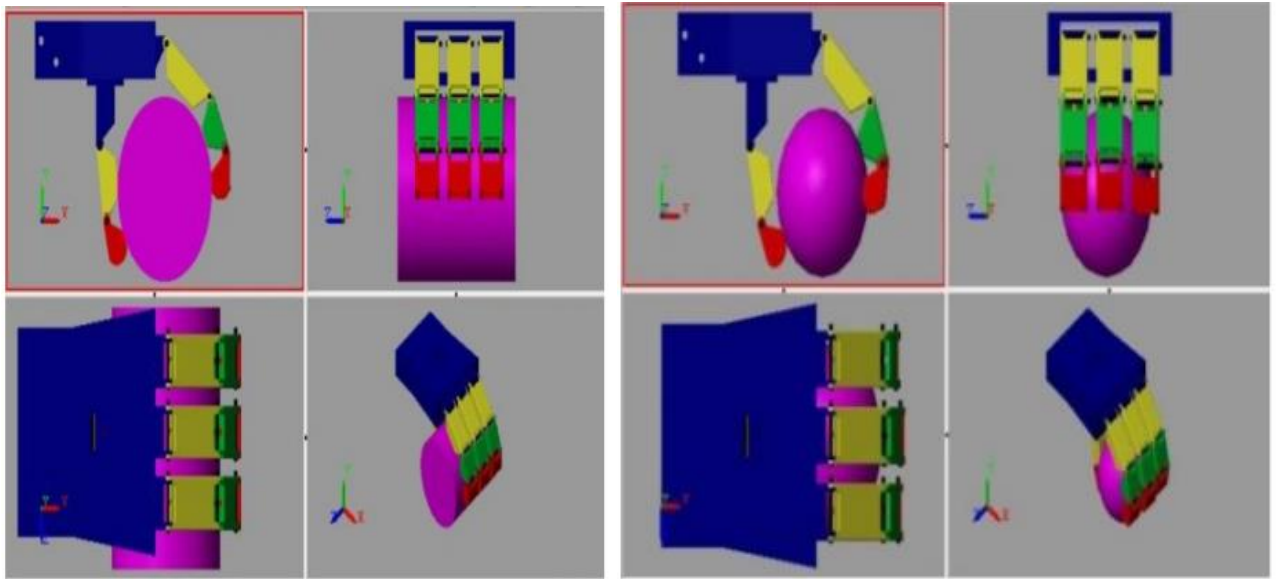


Figure 9 Grasp performance of a four fingers robotic hand

(On the left) cylindrical object grasp, (On the right) spherical object grasp

Considering the advantages given by multi-body programs in obtaining kinematic and dynamic analysis of complex systems and the possibility to simulate object interactions, in this thesis we employed Simscape to model Hannes prosthetic hand.

CHAPTER 2 Materials and methods

This section is divided into two macro sections: In 2.1 an introduction of the prosthetic hand Hannes is presented, followed by an overview of its working principles, then the mechatronics of the system is specifically described in 2.1.2; In particular the paragraph 2.1.2.1 presents the fingers mechanism, while paragraph 2.1.2.3 shows a brief theoretical background about actuators. This is followed by Hannes motor considerations. At last, in paragraph 2.1.2.4 the control strategy is presented.

The paragraph 2.1.3 starts with an overview of the software Simscape, then the various steps to create the Multi-body model of Hannes are presented: in 2.1.3.1 we described how we exported the Hannes CAD model from Creo to Simscape, then in 2.1.3.3 the implementation of master-slave and return wire mechanism is presented, following the discussion about friction modelling in 2.1.3.7. Then, in 2.1.3.5 the DC motor implementation is presented, such as the control strategy in 2.1.3.6. In 2.1.3.7 we explained how the interaction between the fingers and objects of different stiffness was modeled, meanwhile in 2.1.4 we presented our idea for the implementation of object stiffness recognition.

Lastly, in 2.2 we described the experimental set-up and validation metrics used.

2.1 Hannes

The Hannes hand prosthesis was realized thanks to a collaboration between Istituto Nazionale Assicurazione Infortuni sul Lavoro (INAIL) and Istituto Italiano di Tecnologia (IIT) started in January 2014 and ended in December 2017, when the device was officially presented to the public.

Hannes is a poly-articulated and myoelectric prosthetic hand capable to restore over 90% of functionality to upper limb amputees [4]. It is available in two dimensions, small and large, and it can be covered with an aesthetic glove resembling a human hand.

Hannes was designed to realize anthropomorphism, biomimetic performance, and human-like grasping thanks to the underactuated mechanism which is described in detail in the following paragraphs.

As shown in Figure 10 all the human fingers DoFs are implemented in Hannes except for the distal interphalangeal (DIP). Moreover, the thumb interphalangeal (IP) and metacarpal (MCP) joints are locked, while the abduction is actuated and rotation is passive.

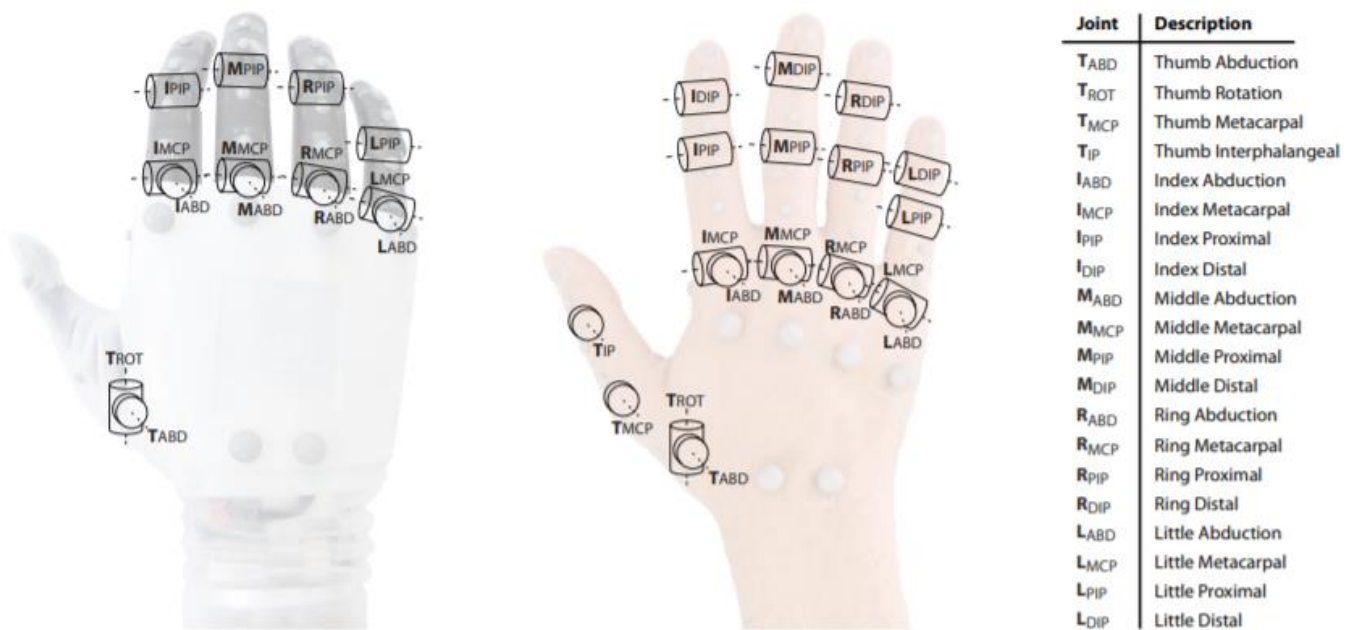


Figure 10 DoFs comparison between Hannes and the real hand

The thumb can assume three different positions thus replicating three types of grasps:

- The precision grasp to pick up small objects;
- The lateral grasp for thin objects;
- The power grasp to grasp and move heavy objects.



Figure 11 Gloved device performing ADLs:

A precision grasp of a pen, a power grasp of a compliant object, a power grasp of a tool, a lateral grasp of a business card, a power grasp of a soft ball, a lateral grasp of a marker.

The device is suitable for trans-radial amputation since it also includes an active wrist in prone/supination and a passive wrist in flex/extension.

These characteristics make Hannes appreciated by upper limb amputees. [4]

2.1.1 Overview

Hannes is a trans-radial prosthesis which comes with a socket (Figure 12) containing the entire dedicated electronics in a watertight and insulated compartment.



Figure 12 Close up on Hannes socket and Hannes hand

The electronic components of Hannes can be sorted into three macro groups between Hand, Wrist and Stump. These elements communicate together by means of a CAN protocol.

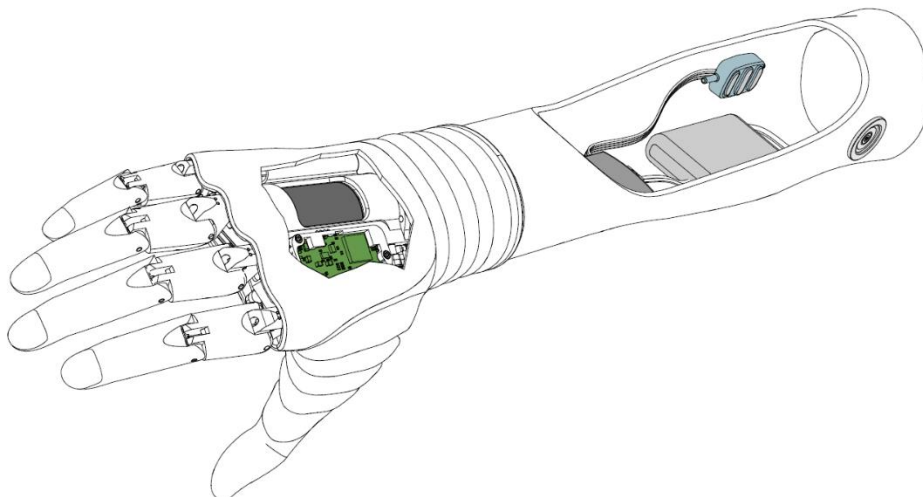


Figure 13 Hand, Wrist and Stump scheme

As we can see in Figure 13, the Stump provides a surface EMG sensors system (Figure 15 g)) that can be used to acquire electromyography signals with a sample frequency of 300Hz. The recorded electrical activity of the muscles is then processed by the electromyography-master (EMG-Master).

The EMG-Master is the core of the system. It is composed by a microprocessor TIVA C microcontroller and EEPROM flash memories. These can elaborate the signals coming from the sensors system allowing fundamental operations such as control of active joints, choice of the control strategy and management of position and velocity references to control the motors.

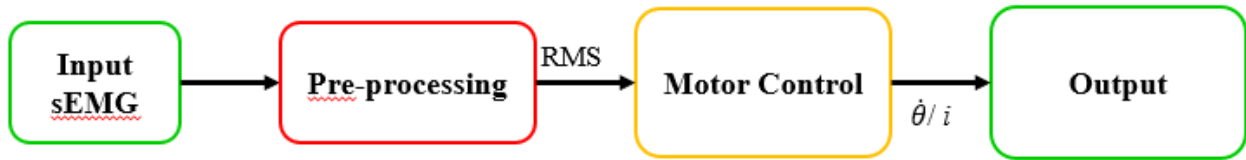


Figure 14 Control scheme:

The input sEMG signals are processed by the EMG-master into the correspondent control strategy used to control the motor.

The whole system is powered through a battery management system which provides 12V. The battery is composed by three lithium prismatic cells connected in series and charged through a plug-in. (Figure 15 c))

Hannes wrist (Figure 15 b)) is actuated by a motor allowing the active prone/supination of the hand. Hannes wrist can rotate about 360° with a continuous movement along both directions.



Figure 15 Supplied accessories:

a)Hannes hand, b) wrist motor, c)battery, d) battery charger, e)power supply AC/DC, f)EMG master, g)EMG sensors

Hannes hand powertrain is composed by a Faulhaber motor with a 19:1 hypocycloid drive reducer. This motor (Figure 15 a)) is controlled by a (Scheda Controllo Motore Mano) SCMM board that transforms the position references coming from the EMG-Master into an electric potential difference to drive the motor.

The board is also instrumented with an encoder that measures the output position of motor gearbox, used to control the motor in reaching the desired grasp position of the hand.

2.1.2 Hannes mechatronic components

Hannes mechanics is the core of the poly-articulated hand thanks which it is able to adapt to any object of any shape allowing 95% of the grasping possibilities.

Hannes system is characterized by a master-slave wire configuration used to control the fingers' movements; particularly, the peculiarity of the prosthetic hand is the differential mechanism (further discussed in 2.1.2.2) [4]. The leader wire on Hannes palm is actuated by the power train: the DC motor rotation causes the wire to roll-up on the motor pulley thus transmitting a force and a speed throughout the entire master system ending up in the thumb.

On the other hand, the four fingers are actuated passively by means of slave wires: The first one connects the index and middle fingers meanwhile the second one connects the ring and little fingers.

The follower wires are coupled with the master system thanks to two differential systems mounted onto two linear guides, each made of custom-made bush bearing and two rails along which the bushing moves.

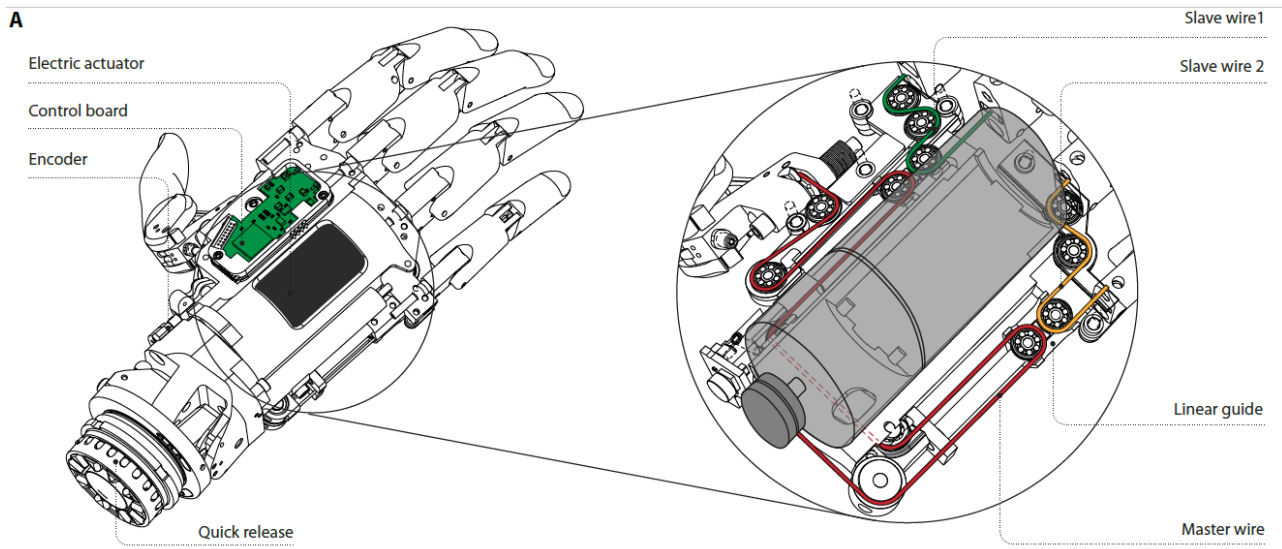


Figure 16 Hannes Master-Slave system for the control of the fingers movement

2.1.2.1 Fingers

Hannes fingers are characterized by 3 DoFs as previously introduced in paragraph 2.1:

- Flexion/extension of the MCP;
- Flexion/extension of PIP;
- Adduction/abduction of the proximal phalanges.

As explained before the four fingers are actuated passively by means of two wires constituting the slave system.

For each pair of fingers, the two extremities of the slave wire are fixed at the level of the distal phalanx; then the cable routes inside each finger thanks to the guidance of dowel pins of negligible friction.

The wire, then, passes through three rotational pulleys: two fixed to the metallic support and the other one hinged to the differential system, further discussed in section 2.1.2.2.

When the linear guide is pulled down due to master wire tension, the force acting on the slave wire generates a moment at the level of MCP and PIP joints.

The kinematic of the fingers is modulated thanks to a return mechanism (each finger is provided with their own return system) which plays a lead role in both keeping the fingers open when no actuation is provided and in gradually let the fingers re-open after closure.

The return mechanism is characterized by three elements such as:

- Wire
- Compression spring (each finger has a different spring stiffness)
- Cylindrical tap

One extremity of the return wire is fixed on the cylindrical tap where the compression spring is inserted too.

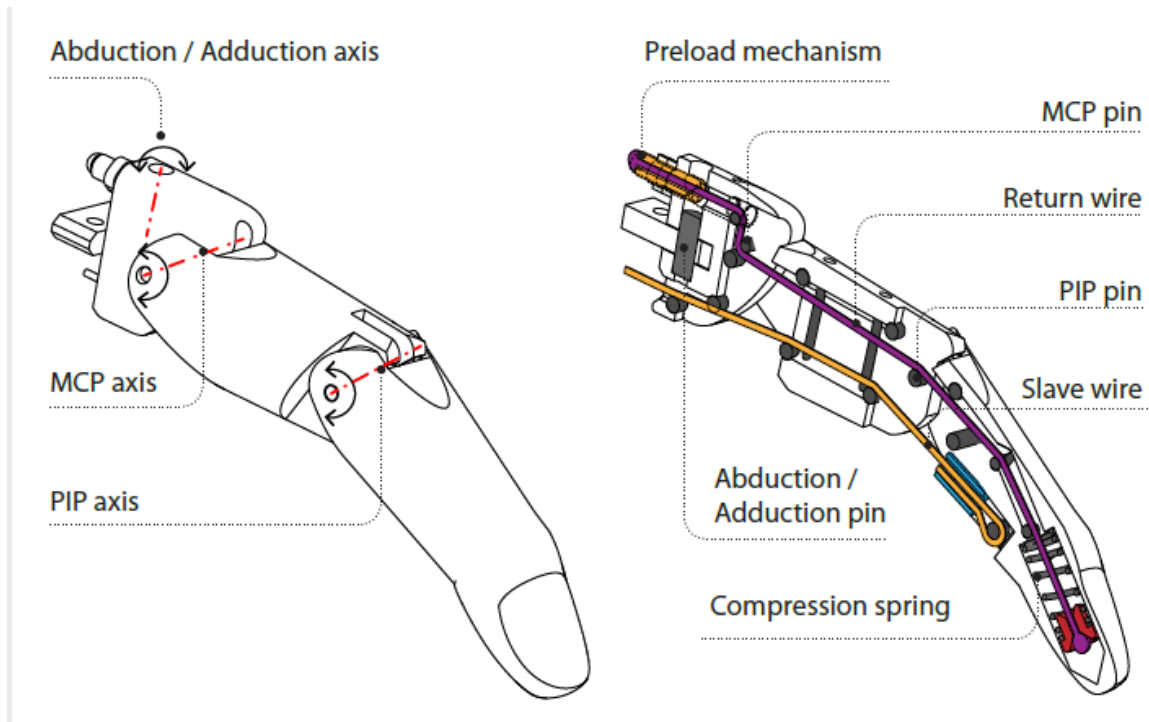


Figure 17 Finger structure:

(On the left) Joint axis names, (On the right) Finger slave (in yellow) and return wire (in violet)

The more the finger flexes, the more the spring generates an increasing force due to its compression.

When the motor unwinds the master wire, the follower wire slackens, and the elastic element extends thus facilitating fingers' re-opening.

For the fingers to stay open when no actuation is provided, the compression springs are pre-tensioned of a certain amount thanks to a screw-based mechanism situated at the other extremity of the return wire, which is fixed on the back of the metallic frame at the level of the knuckles.

The return wire mechanism is tuned such as the fingers close as together as possible. This is especially true when the aesthetic glove is worn; this is because the glove represents an additional stiffness to the return mechanism.

The equilibrium of the moments at the MCP joint (Figure 17) can be expressed as the following equation where F_{A^j} is the force acting on the follower wire for j -th digit and R_i^j is the interaction force at i -th time instant [4].

$$\sum_{i=1}^n R_i^j (b_i^j(\theta_{PIP}^j)) = F_A^j h^j(\theta_{MCP}^j) + f^j n^j(\theta_{PIP}^j) - F_A^j a^j(\theta_{PIP}^j) - f^j y^j$$

$$\sum_{i=1}^n R_i^j (b_i^j(\theta_{PIP}^j)) = F_A^j (h^j(\theta_{MCP}^j) - a^j(\theta_{PIP}^j)) + f^j (n^j(\theta_{PIP}^j) - y^j)$$

$$\{R_1^j \quad \dots \quad R_n^j\} \begin{Bmatrix} b_1^j(\theta_{PIP}^j) \\ \dots \\ b_n^j(\theta_{PIP}^j) \end{Bmatrix} = \{F_A^j \quad f^j\} \begin{Bmatrix} h^j(\theta_{MCP}^j) - a^j(\theta_{PIP}^j) \\ n^j(\theta_{PIP}^j) - y^j \end{Bmatrix}$$

Where:

- b_i^j ... Lever arm relative to the point of application of the interaction force R_i^j and the MCP joint;
- f_i ... Force generated by the return wire;
- Θ_{MPC}^j ... MCP joint angle;
- Θ_{PIP}^j ... PIP joint angle;
- $h^j(\Theta_{MPC}^j)$... Lever arm relative to the point of application of the force F_A^j to generate torque on MPC joint;
- $a^j(\Theta_{PIP}^j)$... Lever arm relative to the point of application of the force F_A^j to generate torque on PIP joint;
- $n^j(\Theta_{PIP}^j)$... Lever arm relative to the point of application of the force f_i to generate torque on PIP joint;
- $y^j(\Theta_{MPC}^j)$... Lever arm relative to the point of application of the force f_i to generate torque on MPC joint.

The adduction/abduction of the proximal phalanx is a passive movement which allows the fingers to better suit the grasping configuration. This is implemented by custom-made hinges provided with rubber insert.

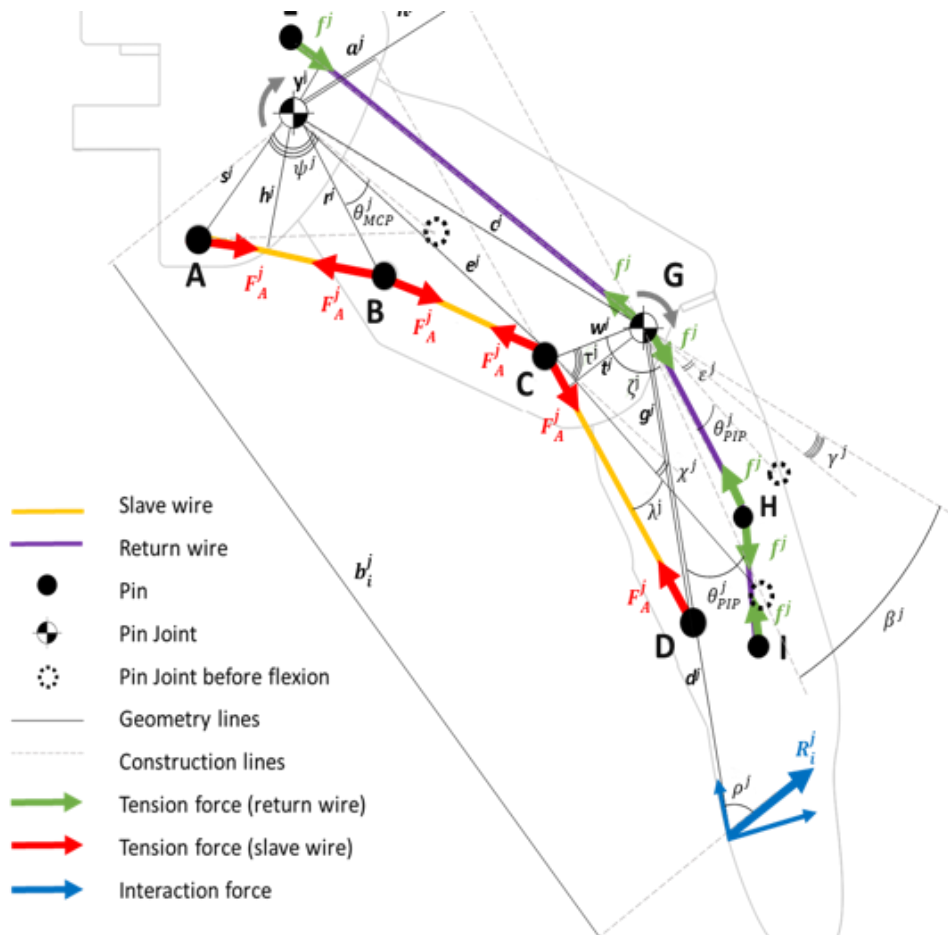


Figure 18 Finger scheme:

Forces acting along the slave and the return wire

Differently from the other four fingers, the thumb is characterized by two DoFs:

1. Rotation (ROT), implemented by a custom-made spring-based plunger mechanism that can lock the finger in three equally spaced positions as shown in Figure 19. This system allows different grasp styles.

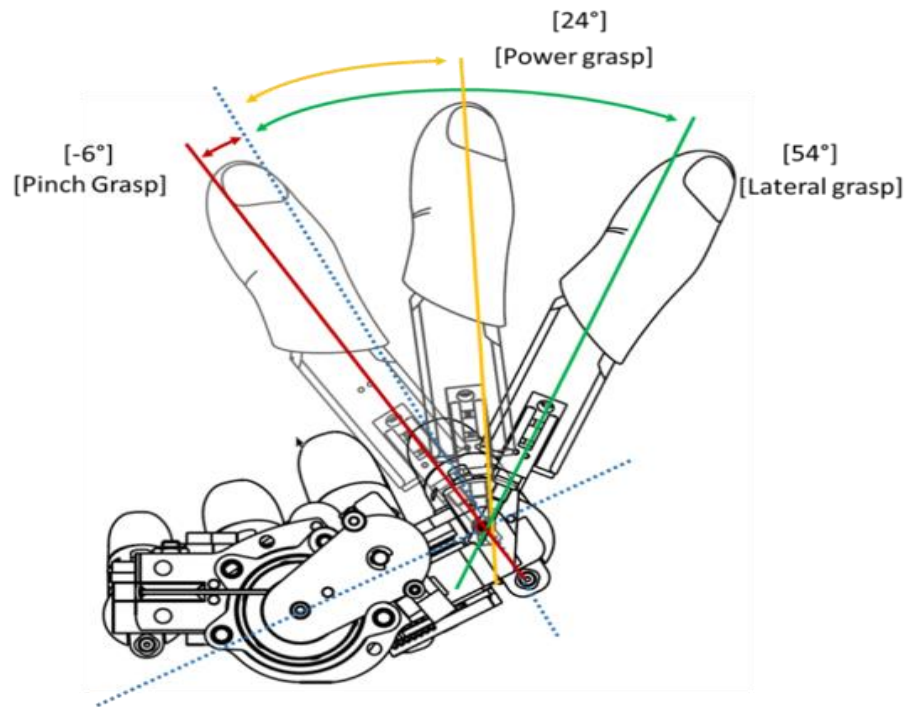


Figure 19 Different thumb configurations

2. Adduction/abduction (ABD), implemented as the MCP joint in the fingers.

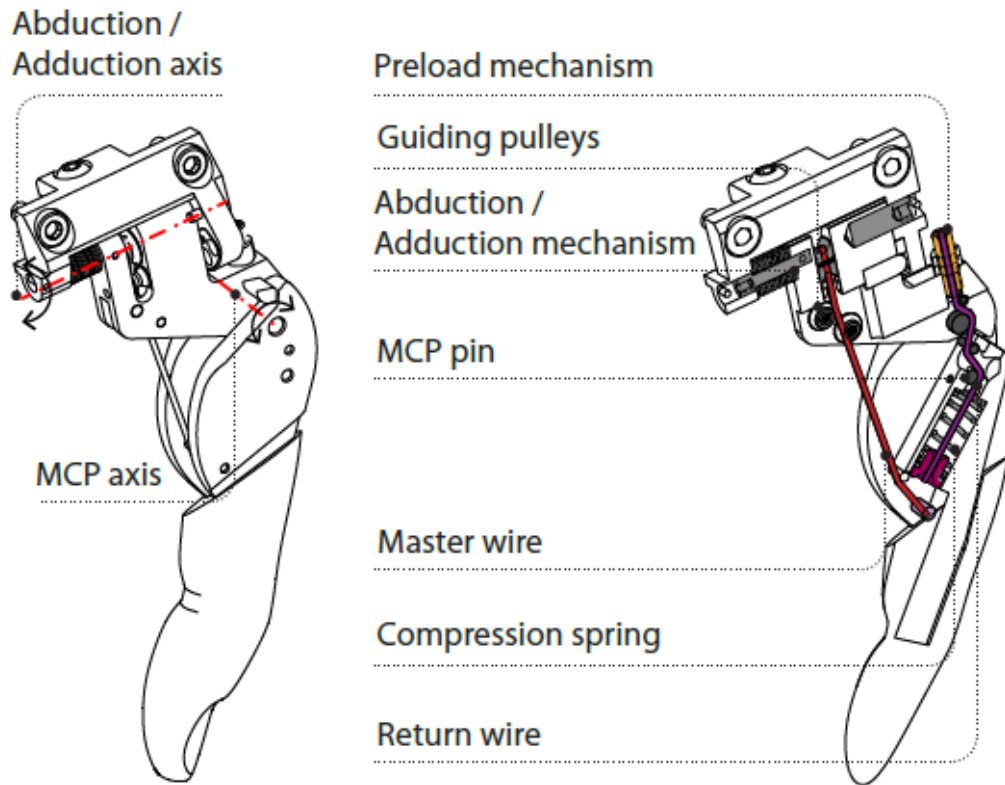


Figure 20 Hannes thumb:

(On the left) Joint axis names, (On the right) thumb master (in red) and return wire (in violet)

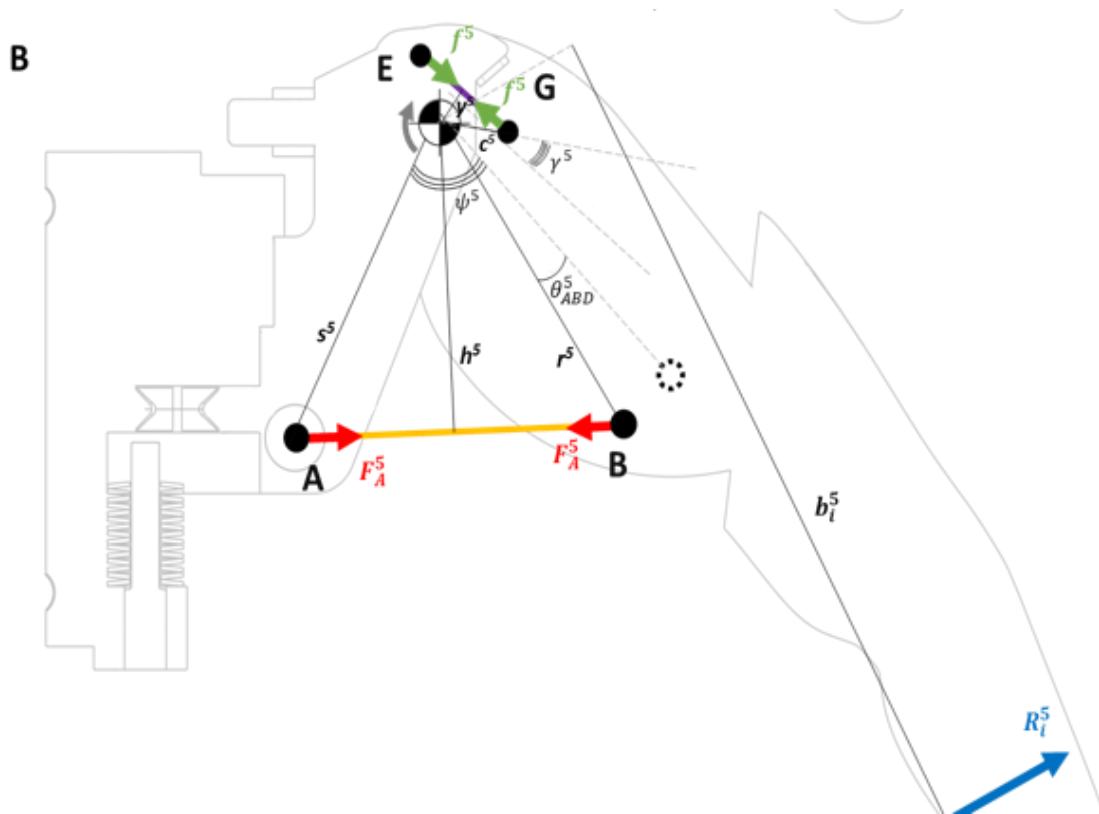


Figure 21 Thumb scheme:

Forces acting along the master and the return wire

In Hannes thumb, the IP and PIP joints are locked.

When the motor winds the leader wire, a force F_A is transmitted throughout the entire master system up to the thumb causing the flexion of the finger. One extremity of the wire is knotted to the motor pulley, then it passes through eight pulleys, and finally reaches the thumb where the other extremity of the wire is knitted at the level of the distal phalanx.

The extension of the finger is implemented by means of a return mechanism like the fingers. Thanks to the screw-based mechanism that acts on the return wire it is possible to tune the preload of the compression spring so that the thumb can be synchronized with the fingers.

An additional torsional spring is added at the level of the ABD joint to make the extension of the finger more natural.

The equilibrium of the moments for the thumb can be expressed as follow where Θ_{MPC}^j is replaced with Θ_{ABD} and Θ_{PIP}^j is not present anymore because constant.

The overall equation results as [4]:

$$\{R_1^5 \quad \dots \quad R_n^5\} \begin{Bmatrix} b_1^5 \\ \dots \\ b_n^5 \end{Bmatrix} = \{F_A^5 \quad f^5\} \begin{Bmatrix} h^5(\theta_{ABD}^5) \\ -y^5 \end{Bmatrix}$$

2.1.2.2 Differential mechanism

As cited in paragraph 2.1.2, the peculiarity of Hannes hand is the differential mechanism: this system couples the master wire at the hand palm to the slave wires at the four fingers which are paired as index – middle and ring – little.

The differential mechanism, represented in Figure 22, is implemented as:

- Custom-made linear cart;
- Two idle pulleys hinged on the carts. The lower pulley routes the master wire meanwhile the upper one routes the slave wire to actuate the two fingers;
- Two rails along which the cart moves. Two compression springs were inserted on the rails in order to reduce the possible slackness of the master wire.

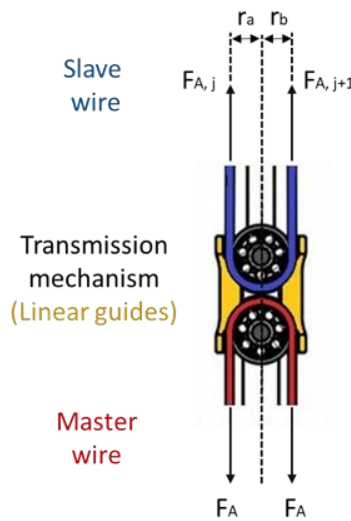


Figure 22 Differential mechanism scheme

When the motor actuates the system, two forces F_A^j a F_A^{j+1} act on the slave wire at the level of the upper idle pulley as a reaction to the force F_A transmitted over the master wire. The relationship between these forces for the j -th finger ($j = \{1, \dots, 4\}$) and k -th differential mechanism ($k=\{1,2\}$) can be described in function of the transmission matrix T_k as [4]:

$$\mathbf{F}_{A,k} = \begin{Bmatrix} F_A^j \\ F_A^{j+1} \end{Bmatrix} = \begin{bmatrix} r_b \\ (r_a + r_b) \\ r_a \\ (r_a + r_b) \end{bmatrix} 2 F_A = \mathbf{T}_k F_A$$

Considering $[S_{Hand}]_{1 \times 5}$ the full transmission matrix, the generated force F_A^j actuating the j -th finger is

$$\mathbf{F}_A^j = \mathbf{S}_{hand} F_A = \begin{bmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \\ 1 \end{bmatrix} F_A$$

Under the assumption of negligible frictions, it's possible to assume that the actuation force F_A acting on the master wire is transmitted unchanged to the slave wires, regardless of the state the hand is operating.

2.1.2.3 Actuators: theoretical background

A motor is a system which either develops energy or imparts movement by transforming a variable from a physical space to a different physical space.

The family of actuators can be divided into two macro-groups: the ON/OFF motors and the proportional ones.

The ON/OFF motors are electromechanical devices based on electromagnetic inductance principles. They are characterized by a coil run by current which creates an attractive force on a movable armature. The resting configuration of the movable armature is generally considered as the OFF position meanwhile when the core is energized the armature shifts toward the other side of the contacts reaching the ON position [48].

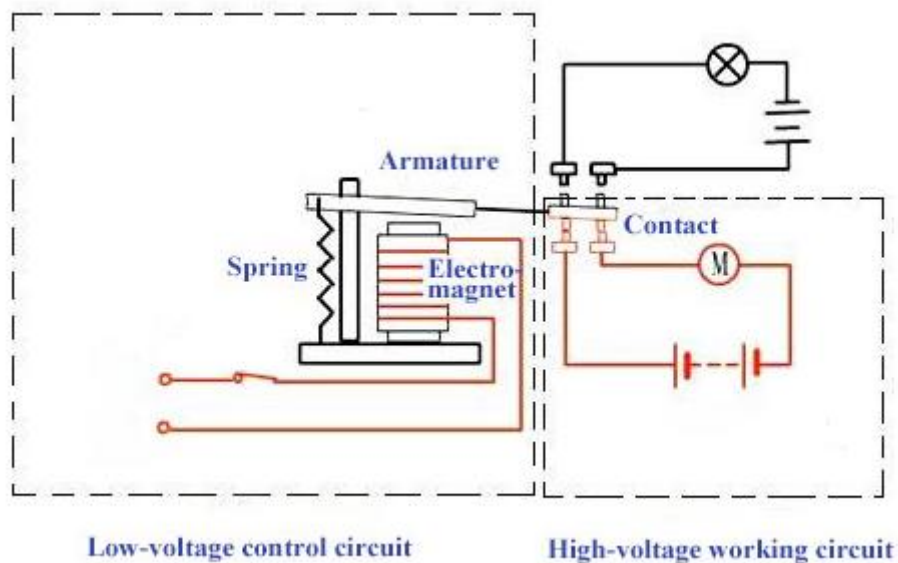


Figure 23 ON/OFF actuator scheme

This kind of actuator is usually utilized when a system is needed to be either always switched ON or always switched OFF and can't be controlled in velocity and position; for this reason, electromechanical actuators are not the best choice for biomedical applications.

On the other hand, DC motors are very used in the biomedical field where efficient speed and position control is required. The working principle behind these kind of actuators lays on a current-carrying tool experiencing a mechanical force (Lorentz's force) when placed in a magnetic field.

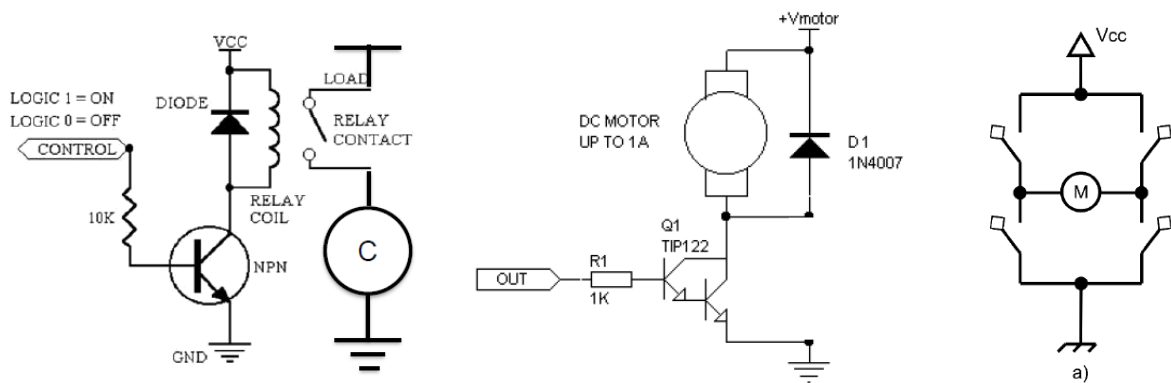


Figure 24 Actuators

(First scheme on the left) Relè actuator: No speed, no direction control. (Middle scheme) Single transistor DC motor: Speed control, no direction control. (Right scheme) DC motor in an H-bridge configuration. Both speed and direction

DC motors can be divided between Brushed and Brushless motors.

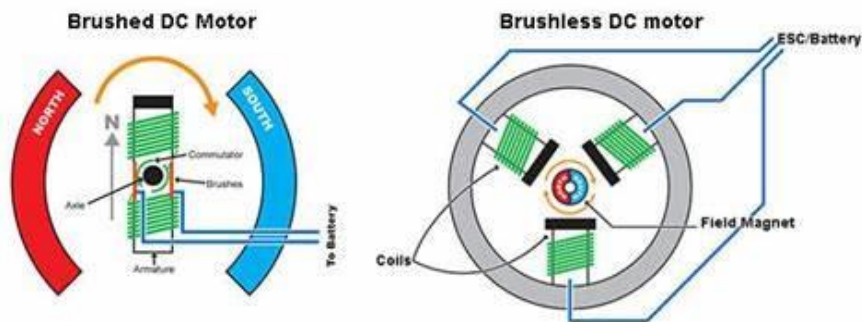


Figure 25 Diagram of Brushed DC motor vs Brushless DC motor

These two families of proportional actuators are characterized by different implementations. The main difference between the two, as one can easily deduce from the names, is that Brushed motors presents brushes meanwhile the Brushless don't.

Also, the Brushed motor is directly powered by means of a battery and an H-bridge system, this allows to apply a regulated tension to the motor. On the other hand, Brushless motors, which also present a battery providing a constant voltage supply, presents a three-phase inverter generating a field of tensions (either trapezoidal or sinusoidal) useful to generate a rotating magnetic field in the stator, which leads the rotor to rotate as well.

Another difference between the two lays on the commutation: Brushed motor commutation happens periodically changing the direction of current between the rotor and external circuit, thus acting as a switch causing a unidirectional torque in the DC motor. The brushes supply the current to the commutator from the power supply.

The commutation of Brushless motors, instead, is made electronically by supplying voltage to 2 out of 3 windings of the stator in order to create a rotational magnetic field. The rotor starts rotating to align its magnetic field to the stator's one. Usually three digital Hall-effect sensors are employed to read the magnitude of the magnetic field, thanks which the rotor position is estimated with respect to the stator, thus controlling the motor.

A DC motor can be modeled as an electric equivalent circuit:

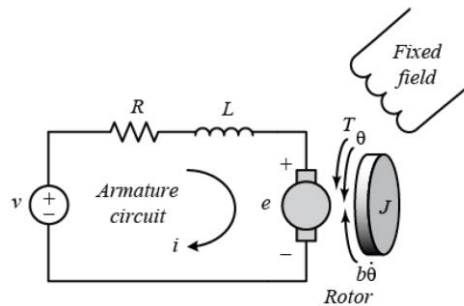


Figure 26 DC motor scheme

Electric equivalent circuit of the armature and a free-body diagram of the DC motor.

The input of the system is the voltage supply while the rotational speed of the shaft is the output. R is the armature resistance and L is the armature inductance.

Under the assumption of constant magnetic field, it's possible to assume that the torque generated by the DC motor is proportional to the armature current by a constant factor motor torque K_t :

$$T = K_t * I$$

Looking at the scheme in Figure 26 it's possible to find the set of equations describing the system, based on Newton 2nd Law and Kirchhoff's voltage law:

$$J\ddot{\theta} + b\dot{\theta} = K_t i$$

$$L \frac{\partial i}{\partial t} + Ri = V - K_t \theta$$

The **speed** of a DC motor is proportional to the voltage applied to the motor [49].

$$\omega = K_t * (V - I_a R_a)$$

Where:

- R_a = armature circuit resistance;
- I_a = armature current;
- V = motor input voltage = machine's total flux;
- K_t = Constant factor

Changing the voltage will change the motor speed.

When considering a digital control, a Pulse Width Modulation (PWM) signal is used to generate an average voltage depending on the Duty Cycle.

$$V_{AVERAGE} = DutyCycle * V_{SUPPLY}$$

PWM is a technique based on the modulation of the Duty Cycle of a square wave defined by amplitude, frequency and DutyCycle.

It basically consists in the modulation of the signal by partializing in time the signal itself. This is obtained by activating or deactivating the signal through an electronic switch (e.g. transistor).

T represents the period of PWM, divided between T_{ON} (duration of activation signal) and T_{OFF} (duration of deactivation signal). The ratio (T_{ON}/T) is called DutyCycle and it represents the percentage of activation time of the signal with respect to the total period T.

Setting a proper PWM frequency is important: too low frequency results in both a noisy motor at lower speeds and a lazy response to changes in DutyCycle meanwhile too high frequency lessens the efficiency of the system.

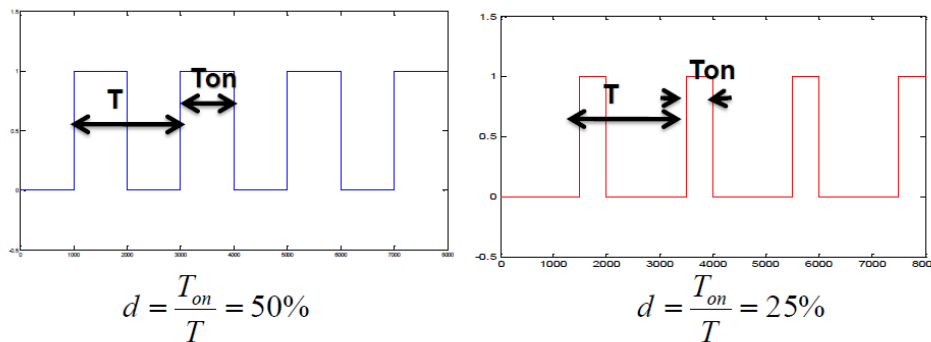


Figure 27 Examples of PWM:

(On the left) PWM with a duty cycle of 50%, (On the right) PWM with a duty cycle of 25%

The **direction** of rotation of a DC motor can be controlled by means of an H-Bridge motor drive. This system (Figure 28) is able to move current in either direction through the motor windings. Q1 and Q2 make up one half of the bridge and Q3 and Q4 the other half. Each of the half-bridge connects the motor to either the power supply or ground.

Particularly, in forward mode one half of the bridge will be switched to the power supply meanwhile the other half-bridge will be switched to ground. If the DutyCycle is maximum (100%) the motor rotates at maximum speed in both configurations [50].

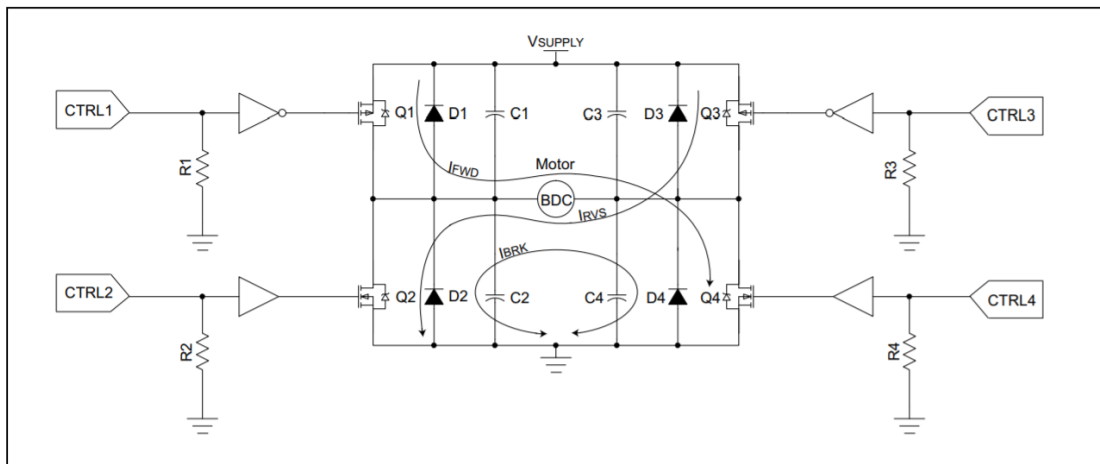


Figure 28 Motor Drive H-Bridge circuit schematic

	Q1 (CTRL1)	Q2 (CTRL2)	Q3 (CTRL3)	Q4 (CTRL4)
Forward	on	off	off	on
Reverse	off	on	on	off
Coast	off	off	off	off
Brake	off	on	off	on

Figure 29 H-Bridge modes of operation

Opening and closure of Q1, Q2, Q3, Q4 switches is modulated by PWM.

As we can see in Figure 29 it's essential for adjacent MOSFETs not to be switched ON at the same time; this could be very dangerous because it leads to a short-circuit between voltage supply and ground, leading to permanent damage to MOSFETs.

In general, the Brushless motors allow a wider range of possible velocities with respect to Brushed ones, other than a better electrical efficiency, the small dimensions of the rotor lead to better thermal efficiency and also the absence of brushes doesn't bring problems related to frictions and wear. On the other hand, the electronic control of Brushless motors is more complex and expensive [49].

As introduced in paragraph 2.1.1, Hannes utilizes a Faulhaber DC motor.

The Hannes motor is reversible which means that two directions are allowed. Also, it is characterized by a high efficiency.

The choice of a Brushed motor over a Brushless was made especially over economical reasons: One of the main project specifications requirements was for Hannes to be limited inside a range proposed by INAIL, making the purchase of Hannes as affordable as possible for amputees. Also, due to geometrical constraints, a Brushed DC motor better fit Hannes specifications.



Figure 30 (On the left) Brushless motor, (on the right) Brushed motor

2.1.2.4 Control strategy

As anticipated in 2.1.12.1.1 the Hannes DC motor speed is controlled through the SCMM board.

The EMG-Master output is a reference position, taken as input by the Logic board and converted into a modulated signal (PWM) which drives the motor as explained in 2.1.2.3. The control is implemented by exploiting two PIDs.

A Proportional – Integrative – Derivative controller (PID) is a closed-loop control strategy based on feedback, widely adopted for industrial applications.

The three parts fulfil the requirements for most of the control systems: The Proportional term is fast in responding to current errors but does not allow to reach the desired setpoint accuracy.

The Integral term filters high frequency noise and yields zero steady-state error in tracking a constant setpoint. However, the integral component is slow in responding to immediate changes of the system.

Lastly, the Derivative term bases the control on a prediction of future error but amplifies high frequency noises [51].

These three parameters need to be properly tuned to achieve the optimal control strategy: The output of a PID is expressed as:

$$PID = K_p * e(t) + K_i \int_0^t e(t')dt' + K_d \frac{\partial e(t)}{\partial t}$$

Where K_p , K_i , K_d represent the three constants to be determined. These parameters must be tuned for each control application because they depend on the response of the system the PID is employed in.

In Hannes system the feedback loop is implemented by means of an Encoder.

The core of an encoder is its engine. The engine takes the positional and speed information and then supplies the signal that is sent to the application.

The sensor employed in Hannes is a magnetic encoder which identifies changes in magnetic fields from a rotating magnetized wheel or ring.

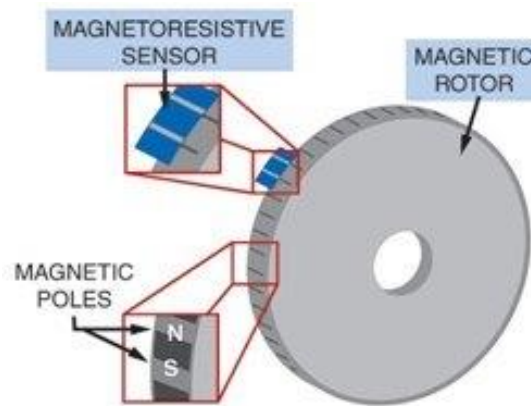


Figure 31 Magnetic encoder scheme

The main components of a magnetic encoder are the sensing circuit, a rotating wheel and a series of magnetic poles around the circumference of the wheel.

Based on the strength of the magnetic field, the poles cause a predictable response in the sensor as the wheel spins past the magnetic sensor. The output is a digital signal proportional to the magnetic field detected.

The resolution of this kind of encoder is determined by the number of magnetized pole pairs on the wheel pole and the number of sensors.

The main advantage of employing magnetic encoders is the reliability in contaminated environments: unlike other type of encoders, indeed, this kind of encoder doesn't suffer from dirt or moisture and work well in broad temperature specifications [52, 53].

Hannes encoder is a custom made sensor provided with a gearbox of 3.8 ratio.

The encoder sends as output the actual angular position of the motor (θ_{misurata}). The input of the first P-controller is the difference between the reference position (θ_{ref}) and the actual one and gives as output a reference current (i_{ref}) in order to modulate the motor speed according to the need.

The error between the reference current (i_{ref}) and the actual motor current (i_{out}) sensed by the system enters the second PI controller resulting in a PWM modulation.

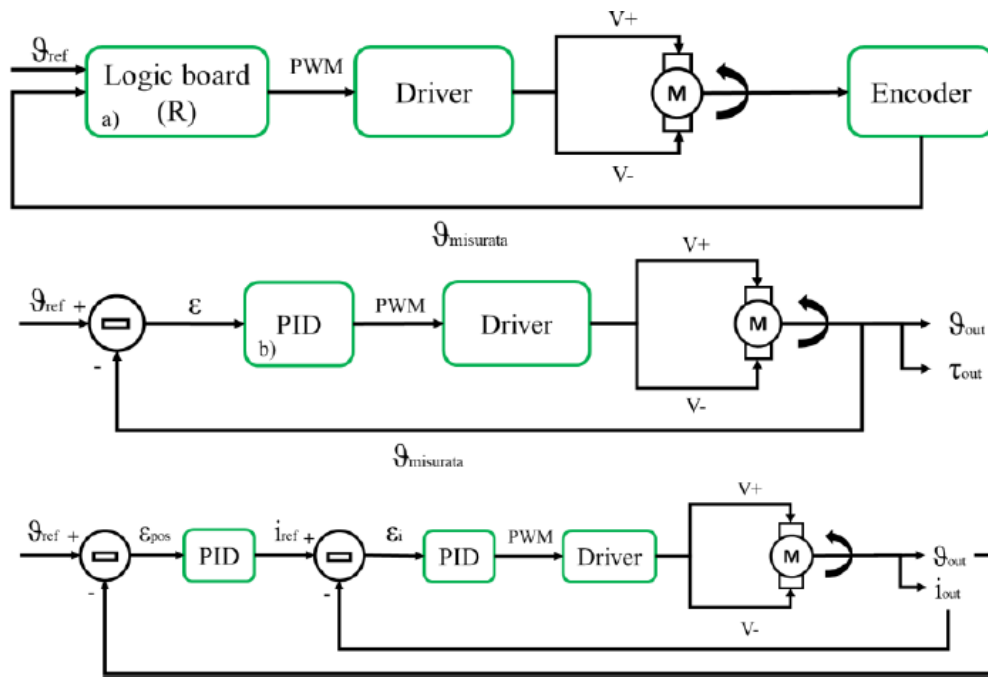


Figure 32 Hannes Control and Driver scheme:

From a general (top scheme) to a more detailed schematic (bottom scheme)

When interacting with an object with an invariant reference angular profile, the error between the reference and measured angle ϵ_{pos} increases, leading to a rise of the motor current. Limiting the current flowing in the motor helps in both preventing any damages to the electrical circuits and restrict the absolute value of the torque generated by the fingers on the object too.

Moreover, in order to save battery, an additional feature was implemented. When performing a closure, if the current flowing inside the batter is higher than a threshold for a specific time interval, then the K_p of the P-controller is set to a value close to zero. As a result, both the battery and motor current drop to zero immediately.

Theoretically, this would cause for Hannes to open but it was observed how the hand still keeps its grip until an open command is given by the Logic board.

The hypothesis is that the overall static friction contributions of the belts and pulleys system is higher than the tendency of the fingers to open, given by the force exerted by the decompression of the springs of the return-wire mechanisms.

2.1.3 Multi-body model implementation

Considering the complexity of Hannes' structure, characterized by the presence of both passive and active components, the best method to model its behaviour is by creating a multi-body model. Among the existing multi-body modelers, it has been decided to use Simscape Multi-body because of the user-friendly interface and the direct connection with MATLAB for the post processing of the data.

Simscape Multi-body is an extension of MATLAB which provides a simulation environment for 3D mechatronic systems. It was realized as a support in the development and testing of control systems, since it allows to parametrize the model using MATLAB variables and to import Simulink control systems to be integrated with the multi-body model. Moreover, it is a valuable resource to verify the design choices before assembling the physical device, thus leading to a reduction in the overall costs and the manufacturing of less mechanical prototypes.

Simscape allows to define rigid and flexible bodies by using 3D parametrized blocks or by creating personalized parts directly on MATLAB. However, it is also possible to import complete CAD assemblies, including masses, inertias and centre of mass positions which are all automatically assigned to the simulated model. The CAD assembly constraints are converted into joint blocks. Inside these blocks there are sensors that allows to monitor the measured variables trends as a function of time. It is also possible to assign stiffness and damping and to actuate the joint in position or in force/torque.

A model realized in Simscape needs three fundamental blocks to work:

- **Solver configuration;**
- **World frame;**
- **Mechanism configuration.**

Each physical network, represented in Simscape, requires information related to the solver settings for the simulation. The solver configuration block (Figure 33) specifies the solver parameters necessary to the model before the beginning of the simulation.

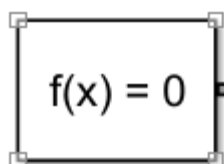


Figure 33 Solver configuration block

The world frame block (Figure 34) represents the global reference in a model. Its axis are orthogonal and built according to the right-hand rule. All the other reference frames are referred to the world frame directly or indirectly.

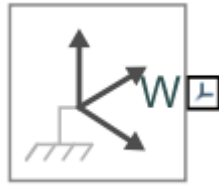


Figure 34 World frame block

The last block, the mechanism block (Figure 35), provides the mechanical and simulation parameters to a mechanism. Particularly, these parameters include the gravity and the linearizing delta for the computation of the partial derivatives. The predefined gravity vector is:

$$G = [0 \ 0 \ -9.80665] \text{ m/s}^2$$



Figure 35 Mechanism block

Apart from these principal blocks, which need to be always present, there are other important blocks such as the rigid transform block (Figure 36) that allows to orient parts.

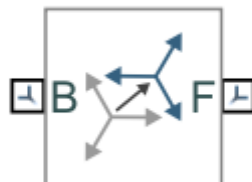


Figure 36 Rigid transform block

This block is characterized by an input and an output port. The input port B (base) specifies the frame to which the rigid transform is referred to, while the output port F (follower) represents the part to which the transform is applied.

In order to insert tri-dimensional physical components it is necessary to use the solid blocks defining geometric characteristics and local reference frames to be used for the coupling with other objects. These couplings are obtained by means of mechanical joints.

The most frequent joint in the Hannes hand is the revolute joint (Figure 37). It allows the rotation around a specific axis.



Figure 37 Revolute joint block

As mentioned before, the joint actuation options, regarding motion and torque, can be set in this block.

Concerning the torque there are three possible alternatives:

- None: no actuation torque;
- Provided by the input: actuation torque given from a physical signal;
- Automatically computed: Simscape computes and applies the torque based on the model dynamics.

On the other hand, the options for the motion are:

- Provided by input: the motion is provided by a physical signal which describes the desired trajectories;
- Automatically computed: the motion is automatically computed and applied by Simscape based on the model dynamics [54].

The cooperation between all these blocks, resulting in a visual 3D simulation of the complete system, is possible thanks to the solver which elaborate and solves the resulting differential equations of motion.

In Simscape different types of solvers are available and the choice of the most appropriate depends on the application.

A Simscape solver employs numerical methods to solve a set of ODEs representing the model. It computes the time of the next simulation step and it tries to fulfil the accuracy requirements set by the user. The simulation step duration is called step size and it is the interval in which the computation happens. The choice of the right solver depends on the following characteristics [55]:

- **System dynamics;**
- **Computation speed;**
- **Solution stability;**
- **Solver robustness.**

Each solver makes a different trade-off between these characteristics. Specifically, MATLAB solvers can be classified by three properties:

1. Computation step size;
2. Model states;
3. Integration scheme.

Considering the computation step size, there are two types of solvers:

- **Fixed step solvers:** solve the model by using the same step size from the beginning until the end of the simulation. It is possible to set the step size before starting the simulation or to let the solver choose it automatically. Generally increasing the step size makes the model less accurate but it increases the computation speed. It is necessary to choose this type of solver for real time application.
- **Variable step solvers:** vary the step size during the simulation. In this way these solvers can decrease the step size to increase the accuracy in correspondence of specific events (e.g., zero-crossing events, rapid state changes, etc.). While they can increase the step size when there are slow state changes. As a result, the total number of steps is reduced leading to faster simulation with good level of accuracy during specific events. However, these solvers cannot be used for real time applications.

Concerning the model states, the solvers can be distinguished in:

- **Continuous solvers:** employ numerical integration to compute continuous states of the model at the current time step based on the states at the previous step and on the state derivatives.
- **Discrete solvers:** they are used to solve discrete models. They only compute the next simulation time step for the model relying on the fact that each block in the model updates its individual discrete state.

Finally, considering the integration scheme, solvers are divided into:

- **Implicit solvers:** compute each step self-consistently. They are designed to solve stiff problems, thus systems which have both slowly and quickly varying continuous dynamics. They provide great stability for oscillatory dynamics, but they are also computationally more expensive.
- **Explicit solvers:** use past information in the equation to solve the next step. This causes instability since the system of equations is not solved completely at any time. Explicit solvers are used for non-stiff problems.

The characteristics just mentioned can be recognized from the solver name. The computation step size is indicated by the number of digits. A single digit number represents a fixed step solver. The information related to the integration scheme are given by the letters. If there are only numbers in the solver name, it is explicit [56].

Considering all this information, we chose the solver *ode23t* for our application. It computes the model's state at the next time step using an implementation of the trapezoidal rule with a free interpolant. It is an implicit solver since our problem is stiff and it is variable step because it reduces the computation time.

The solver performance can be evaluated thanks to a valuable resource offered by Simscape, the solver profiler.

The solver profiler analyses the model highlighting patterns which affect the simulation speed. It provides graphical and statistical information about the simulation, solver settings, events and errors as shown in Figure 38. These data are useful to understand the points where the solver has difficulty in finding the solution to the equations. Particularly, the solver profiler reports the most important events that occur when simulating a model such as:

- Zero-crossing events;
- Solver exception events;
- Solver reset events;
- Jacobian computation events.

These events are necessary for an accurate simulation. However they involve a certain computational cost, thus if they occur frequently they can slow down or even stall the simulation [57].

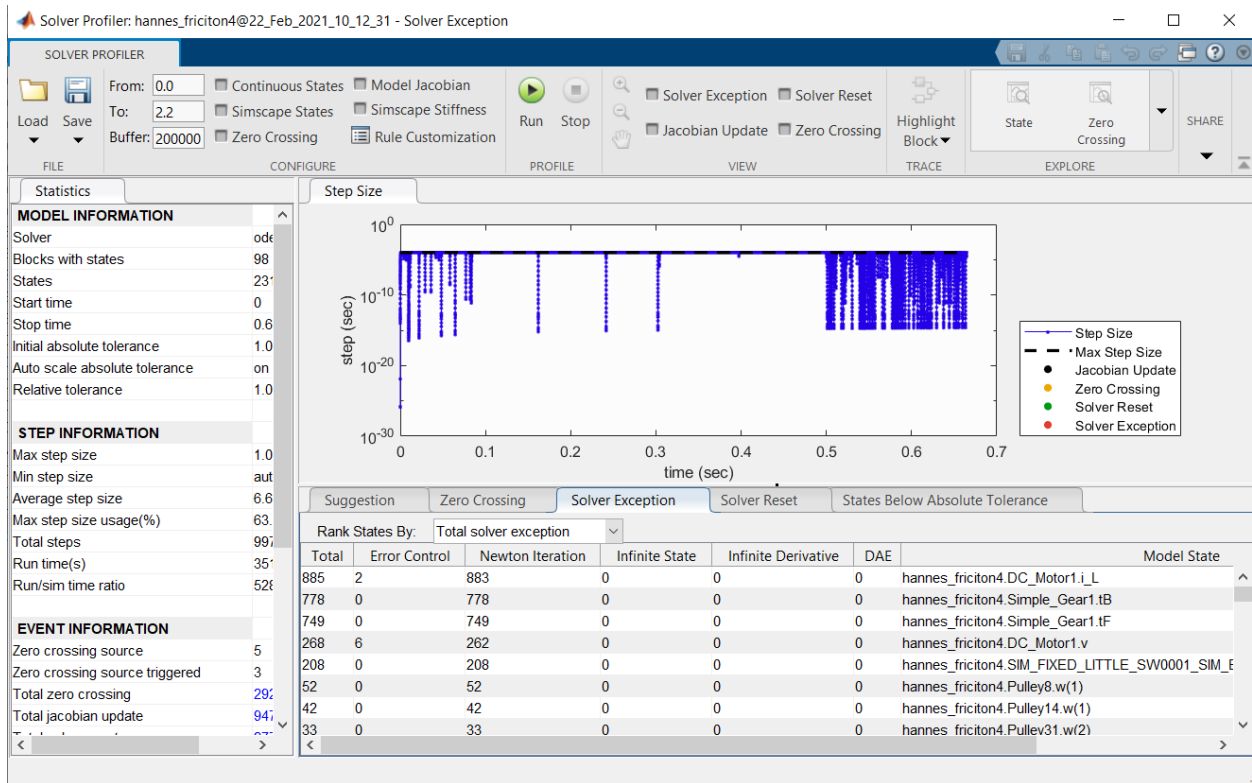


Figure 38 Solver profiler interface

The solver profiler was an essential tool in our thesis since it helped us to understand how to increase the simulation speed.

2.1.3.1 From CAD to Simscape

The thesis work started from the CAD model of the Hannes in Creo. As stated, there are four versions of Hannes whose differences depend on the size, small or big, and on the hand, left or right. The CAD model used was of a small right hand which included all the components.

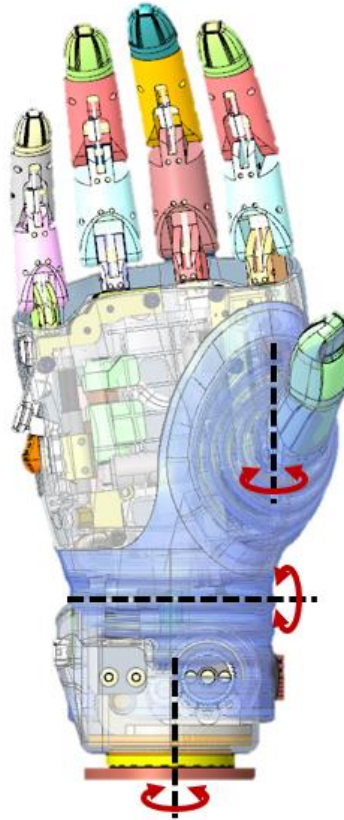


Figure 39 Complete CAD model of Hannes:

This model includes all the mechatronic components

However, the complete CAD model (Figure 39) was too complex to be exported in the Simscape environment, so it was necessary to select only the most important components. For this reason, we chose to export the metallic support and the five fingers case, thus excluding all the mechanical pins, springs and wires present inside.

The exportation was possible thanks to the SimMechanics link. This is a MATLAB plug-in compatible with PTC Creo which needs to be installed on the CAD application to export CAD assembly models. The plug-in takes as input an assembly and provides as outputs an XML model description file and .stl mesh files.

First of all, it was necessary to create the assembly including the hand principal components following a guide explaining how to obtain an URDF (Unified Robotics Description Format) file from a CAD model [58].

The first step consisted in grouping together parts that move together generating the link geometries.

Links are rigid bodies characterized by three elements:

- Visual element specifies the shape and the material of the link.
- Inertial element specifies the inertial properties such as mass and inertial matrix.
- Collisional element specifies the collisional properties.

Links are constrained together through joints which describe the relative motion between them.

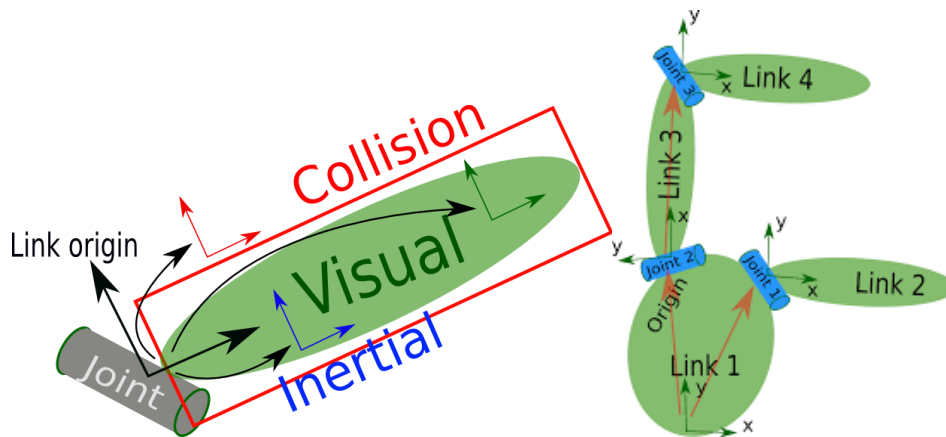


Figure 40 Links and joints

(On the left) Links and joints assembly. (On the right) Characteristics of a link

Each generated link geometry was saved as a *shrinkwrap* and connected to the others in a Creo Assembly. Once the Assembly was complete, the SimMechanics link was used to export the assembly as an XML file.

An XML file is a plain text file that uses custom tags to describe the structure and other features of the document in a way that both humans and robots can read [59].

Afterwards the XML file was converted into an URDF file by means of a python script [60].

An URDF is an XML specification used to model multi-body systems: It is composed by various XML elements, such as `<robot>`, `<link>`, `<joint>`, nested in hierarchical structures known as XML trees. It provides information about the type of joints, inertia properties of the links and so on.

Finally, the generated URDF file was imported in Simscape thanks to the `smimport` function.

2.1.3.2 Characterization of Simscape blocks

Once the URDF model was imported, Simscape generated a multi-body model of the hand, whose visualization is reported in Figure 41.

As shown in Figure 42 the white block on the left represents the metallic support to which the five fingers are attached. Each finger, apart from the thumb, is composed of four links which we referred to as:

- **A fixed link** rigidly connected to the metallic support through a weld joint.
- **A base link** connected to the precedent one through a revolute joint allowing the movement of add/abduction of the finger.
- **A phalanx link** connected through a revolute joint for the flexion/extension of the proximal phalanx.
- **A middle link** connected through a revolute joint for the flexion/extension of the middle phalanx.

As introduces in 2.1.2.1, the four fingers in Hannes have one DoF less with respect to the human fingers since the middle phalanx and distal phalanx are rigidly connected resulting in one link.

The same goes for the thumb whose rotation, in addition, is not active; consequently the user needs to passively rotate it to perform different type of grasps. It can have three stable positions which are equally spaced along a range of motion (ROM) of 60°.

The three links composing the thumb are:

- **A fixed link** rigidly connected to the metallic support through a weld joint.
- **A base link** connected to the precedent one through a weld joint, whose angles can be changed to obtain the three positions previously mentioned.
- **A thumb link** connected through a revolute joint for the abduction of the thumb.

Each link composing the fingers was characterized in terms of inertial properties which were derived from Creo CAD model after correctly characterizing each part with the corresponding material.

Specifically, the inertial properties set were the mass, the coordinates of the centre of mass and the products and moments of inertia computed with respect to the centre of mass. This step was necessary since the exportation from CAD model resulted in a wrong inertial characterization.

In order to verify if the inertia properties were set correctly, we performed a dynamic analysis exploiting the inverted double pendulum model since it has the same structure of the fingers (two links and two hinges).

The purpose was to compare the drop time of the links of each finger, subjected to the gravity force, with the one resulting from the analytical calculations. For this reason, we built four models of inverted double pendulum, each specific for each finger, by assigning the same length, mass and moment of inertia of the phalanx and middle links to the two parts composing the pendulum. We defined the angles θ_1 and θ_2 to reproduce the initial condition.

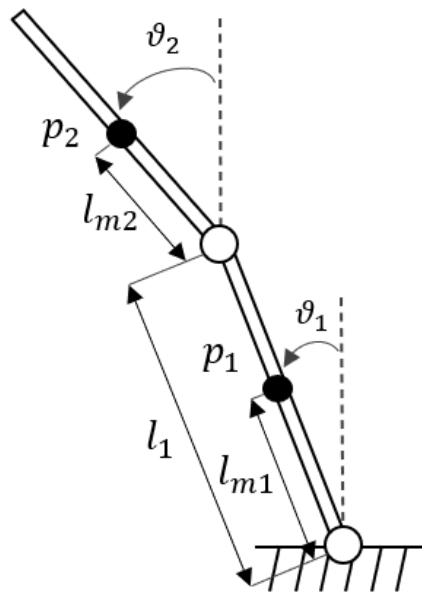


Figure 43 Double-inverted pendulum:

Schematic employed for the Euler-Lagrange computation

The drop time was computed analytically starting from the Euler-Lagrange equation:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = u_i$$

$$L(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - U(\theta)$$

Where:

- L = Lagrangian equation;
- $\theta, \dot{\theta}$ = angular displacement and velocity;
- u = generalized forces performing work on θ ;
- T = kinetic energy;
- U= potential energy.

The complete set of equations is reported in Appendix I.

Once obtained this time, we simulated the inverted double pendulum behaviour on the fingers in Simscape, by eliminating the mechanical limitation at the level of the joints and we registered the simulated drop time. We compared the time necessary to reach an excursion of 82° for the phalanx link and of 50° for the middle one. The results are reported in section 3.1.1.

Concerning the revolute joints, it was necessary to impose an upper and lower angular limit to define the ROM of each of them.

Table 1 reports the angular boundaries imposed.

	Index	Middle	Ring	Little
ABD (up)	5°	5°	5°	5°
ABD (lw)	-5°	-5°	-5°	-5°
MCP (up)	82°	82°	82°	82°
MCP (lw)	0°	0°	0°	0°
PIP (up)	0°	0°	0°	0°
PIP (lw)	-50°	-50°	-50°	-60°

Thumb	ROT	ABD
up	0°	60°
lw	-60°	0°

Table 1 Angular limits for each joint of each finger

2.1.3.3 Fingers return mechanism

As previously mentioned in 2.1.2.1, the return mechanism is responsible for the reopening of the fingers and it is composed of a compression spring, the return wire and the mechanical pins to guide the wire.

The compression spring was modelled through the block reported in Figure 44 which can be found in the Simscape library “Forces and Torques”.



Figure 44 Compression spring block of the “Forces and Torque” library on Simscape

This block allows to characterize the spring stiffness and its natural length whose value was provided in the datasheets of the springs used in Hannes fingers.

The compression springs are pretensioned of a certain amount to keep the fingers open and to obtain a refined grasp.

By using a callipers, we measured empirically the springs shortenings and set them in Simscape by defining the position of the two extremities of the spring through rigid transforms connected to the two ports of the block.

The coupling between wire and mechanical pins was realized using the Simscape library “Belt and Cables” containing the blocks reported in Figure 45.

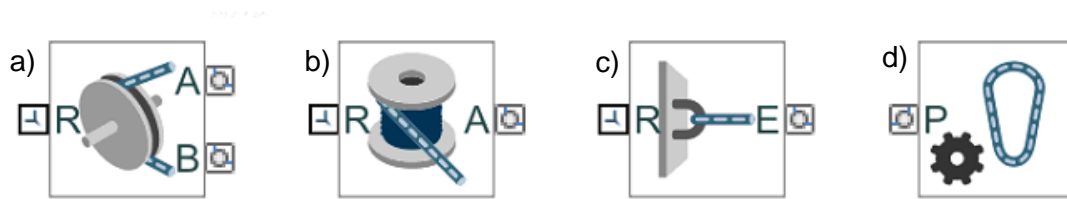


Figure 45 “Belt and Cables” Simscape library

a) Pulley block, b) Spool block, c) Cable-end block, d) Wire properties block

The Figure 45 a) represents a pulley of a belt-cable circuit. The pulley rotates about the reference frame z-axis; the belt-cable wraps from port A to port B in the standard counter-clockwise direction. Instead, port R needs to be connected to a rigid block representing the pulley where its mass and dimensions are assigned.

We modelled the pulley as a cylindrical rigid body of given dimensions and positioned it through a rigid transformation previously derived from Creo, in order to place each mechanical pin in the same position as in the CAD model.

The Figure 45 b) represents a spool of belt-cable with unconstrained capacity to spool and unspool, the Figure 45 c) is the end point of a belt-cable and finally the Figure 45 d) models the cable itself.

In the implementation of the return wire mechanism the contribution of the aesthetic glove wasn't considered (see 2.1.2.1).

In this case another dynamic analysis was performed to assess if the return mechanism was working correctly. For this purpose, we computed the equilibrium torques with respect to the joint between the phalanx and the middle link for each finger. This was done to find analytically the minimal force needed to move the middle link.

$$\Sigma M = 0$$

$$W \cdot l_w + F_{ext} \cdot l_{ext} - F_s \cdot l_s = 0$$

$$F_{ext} = \frac{F_s \cdot l_s - W \cdot l_w}{l_{ext}}$$

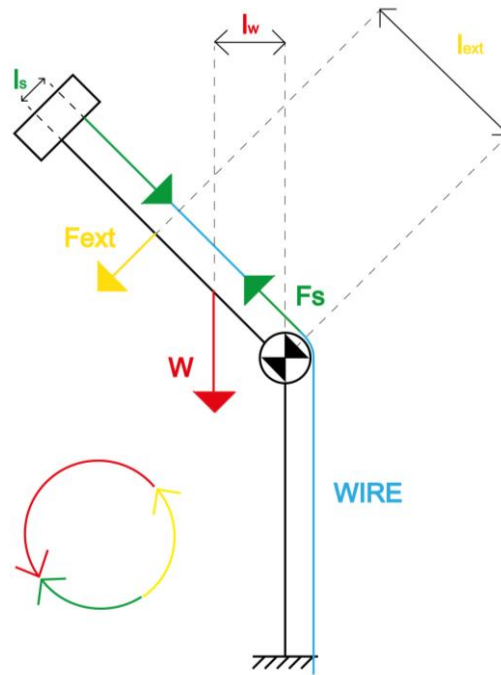


Figure 46 Forces and moments acting on the middle part of a finger

Considering each finger in Simscape, the phalanx joint was locked, and a force was applied to the middle part (see the profile of the force applied in Figure 47). Consequently, the minimum force needed to move the middle link was measured. In section 3.1.2, the comparison between the analytical and the simulated minimal force is reported.

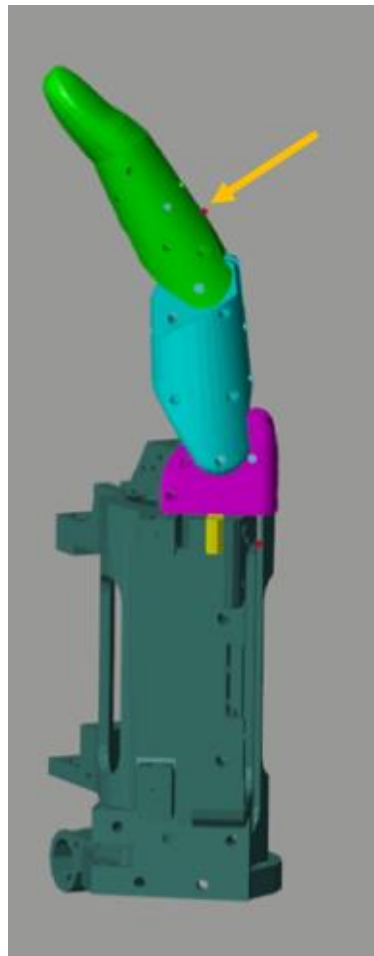
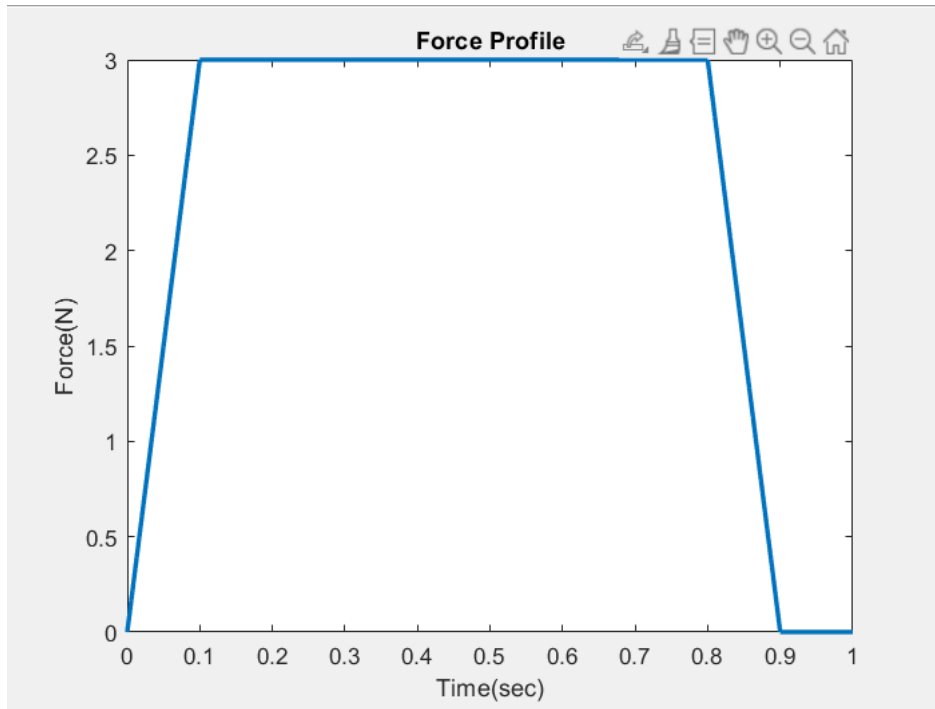


Figure 47 Return-wire simulation

(Up) Profile of applied force, (Down) location of point where the force was applied.

Unfortunately the MATLAB Multi-body library didn't allow to define non-rotating mechanical pins for the wire routing (as actually implemented in Hannes), thus we had to implement rotating pulleys to route the wires. This limitation caused a remarkable increase in the compilation time, more than 12 hours, since the solver was asked to solve the dynamics of each rotating pulley.

In order to reduce the compilation cost, we decided to implement an alternative solution which consisted in the introduction of torsional springs (Figure 48) in place of the original return mechanism.



Figure 48 Torsional spring

The torsional springs were implemented through the revolute joint block as indicated in Figure 49.

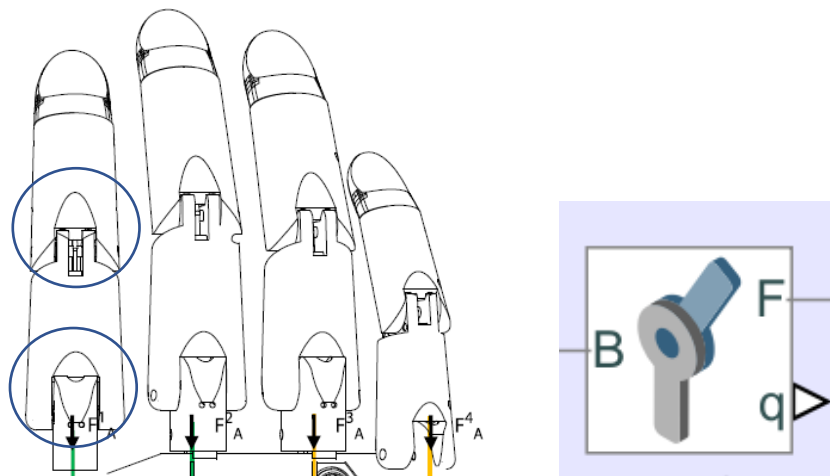


Figure 49 Torsional spring in simulation

(on the left) Location of implemented torsional springs. (On the right) Revolute joint block

The pre-load and the elastic constant of the spring were set as the equilibrium position and spring stiffness respectively. The procedure of finding the values of these variables consisted of different steps.

First an external force was applied to the middle link of each finger (Profile of the force in Figure 50).

The force caused the closure of the fingers, thus the joint angular excursions and the spring forces were recorded.

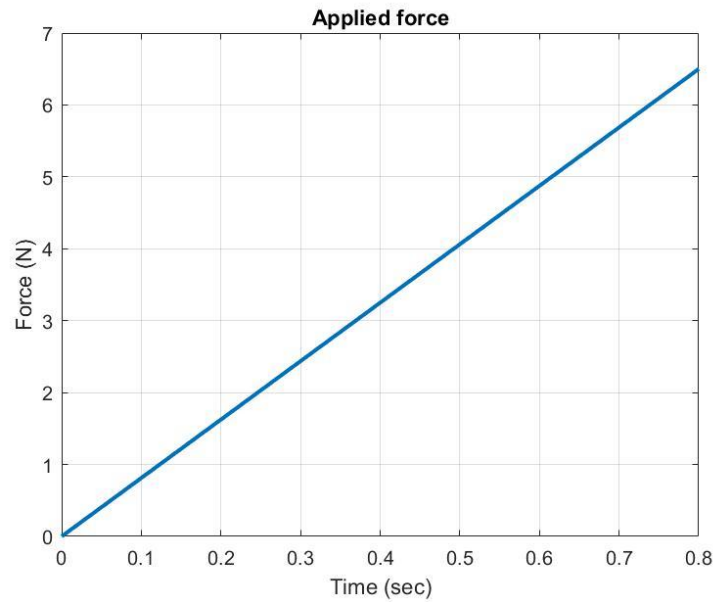


Figure 50 Applied force profile

The values of the moments at the two joints were computed through the following formula:

$$M = F \cdot b$$

Where:

M = moment

F = spring force registered from the Simscape simulation

b = spring force arm

The aim was to find the correct values for the torsional springs pre-load and stiffness so that they would apply the same moment to the joints as the original return mechanism. The moment given by the torsional springs is described by the formula:

$$M = K \cdot \theta \pm M_0$$

M = moment

K = spring elastic constant

θ = angles registered from the Simscape simulation

M_0 = moment at equilibrium

For each trial, a considerable number N of values of moments and deflexion angles were recoded. These variables constitute a system of N equations where the only unknown variable is the stiffness K.

Since it was an overdetermined system, the K value was identified by exploiting the least square method.

Once the torsional springs were implemented, it was necessary to verify if the new solution was equivalent to the precedent one. Consequently, we applied the force profile reported in Figure 50 to the distal phalanx of each finger and we recorded the angular excursion of the two joints, both in the model with torsional springs and in the one with the traditional return mechanism.

The results of the comparison are reported in 3.2.1.

2.1.3.4 Master-slave system

Concerning the hand closure mechanism, it is possible thanks to the system of pulleys previously described.

The leader wire is wrapped around the motor pulley sited at the base of the hand. Then it wraps around thirteen pulleys and finally it is fixed to the thumb. This system of rotating pulleys was modelled with the “Belt and Cables” library used before (Figure 45).

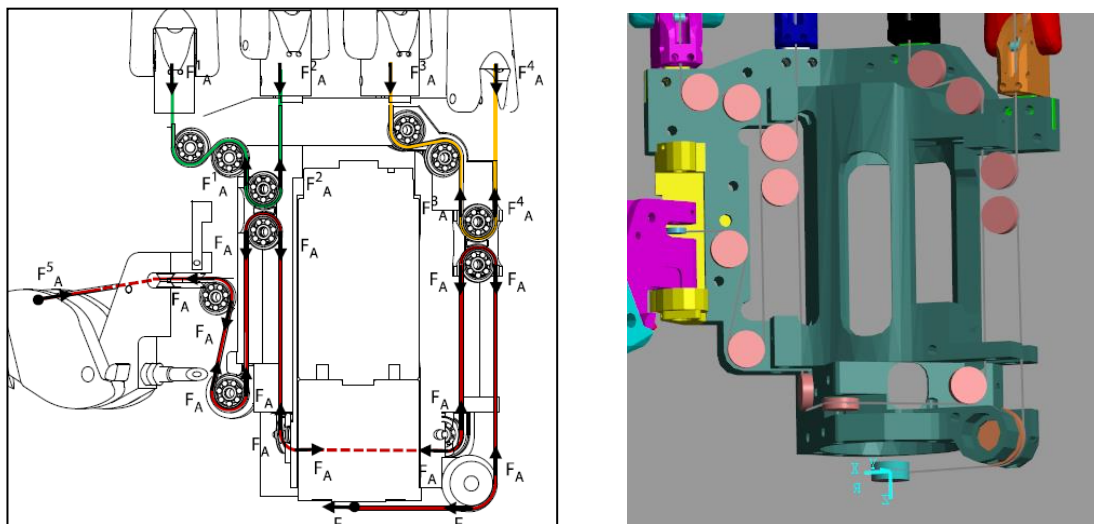


Figure 51 master-slave system in simulation

(On the left) Master wire scheme. (On the right) Master wire Simscape implementation

Two differential elements are part of this system and they allow to activate the four fingers. They are made of two carts moving vertically, which contain two pulleys each. The leader routes around one pulley and the slave wire routes around the other.

The carts were modelled by adding a prismatic joint (Figure 52) to the two pulleys previously described.



Figure 52 Prismatic joint block

As introduced in 2.1.2.2, two compression springs are inserted at the bottom of the two carts. These elements were modelled as linear springs by exploiting the prismatic joint block in Simscape.

There are two slave wires: one is connected to both index and middle, while the other to ring and little. These two wires pass through a system of mechanical pins, similar to those of the return mechanism, which were modelled as rotating pulleys as before.

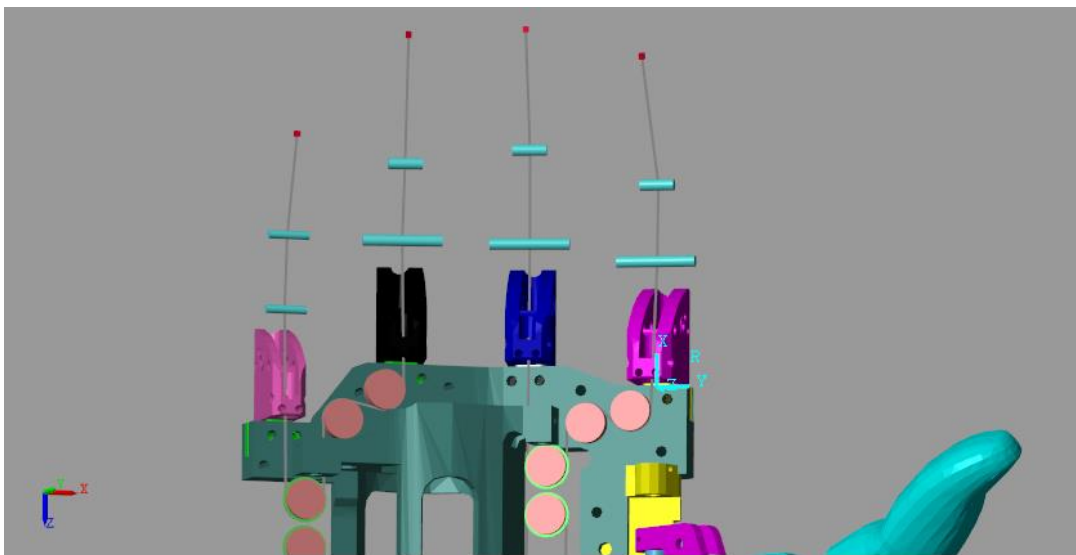


Figure 53 Slave-wire Simscape implementation

Lastly, we implemented the adduction/abduction modulation of the fingers. We modelled the hinged constrains as a torsional spring inserted at the level of the revolute joint, at the base of each finger. The stiffness of such springs was found empirically.

The use of rotating pulleys in place of the mechanical pins caused the same problems previously mentioned. In this case the solution we adopted consisted in avoiding the “Belt and Cables” library by exploiting the physics behind a pulley system: each pulley introduces two forces along the wire characterized by same module, wire direction and opposite versus.

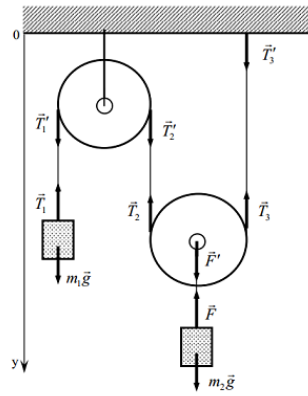


Figure 54 Example of a pulley system:

Representation of the involved forces

The same goes for each pin present in the slave mechanism. We implemented one force for the middle link and two forces for the phalanx link as shown in Figure 55.

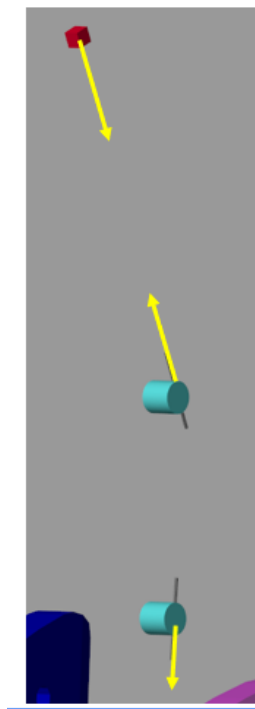


Figure 55 Scheme of Forces implementation:

In yellow the forces replacing the wire

We removed all the thirteen pulleys and the leader wire, we modelled all the mechanical pins inside the fingers as rigid bodies and we inserted external forces in place of the slave wire.

The force value was computed from the torque applied by the motor pulley.

Once this solution was implemented, we verified the amount of error it introduced, in terms of fingers range of motion, with respect to the model with the original pulley system. In this way we were able to exploit this simplified model to perform quicker tests since its compilation time reduced to 1 minute.

The comparison between the pulleys implementation and external forces implementation can be found in paragraph 3.2.2.

2.1.3.5 DC motor

The brushed motor described in 2.1.2.3 was modelled through the DC motor block reported in Figure 56, present in the brushed motor section of the “Electromechanical” library.

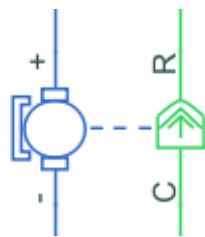


Figure 56 DC motor model

This block represents the electrical and torque characteristics of a DC motor. When a positive current flows from the electrical + to - ports, a positive torque acts from the mechanical C to R ports. It was necessary to set both the electrical and mechanical parameters, taken from the Faulhaber motor datasheet.

The +/- ports are connected to a controlled voltage source which takes as input the PWM (pulse with modulation) given by the PID controller. On the other hand, the R port is connected to a simple gear block (Figure 57) modelling the cycloidal gearbox present in Hannes. Inside this block the reduction ratio of 19:1 was set.

After the first reduction stage we connected an ideal rotational motion sensor, measuring the slow shaft rotation angle in radians, which is used to control the motor pulley in position. It was necessary to add another reduction stage with a reduction ratio of 3.8:1 to model the gearbox present inside the encoder. This last one was modelled through an ideal rotational motion sensor.

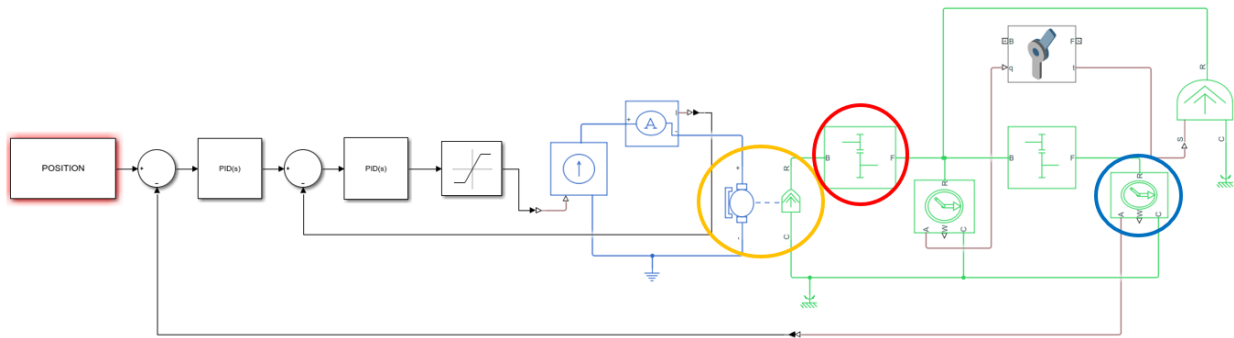


Figure 57 Realistic control scheme

Model of DC motor (in yellow), gearbox (in red) and encoder system (in blue) connected to a double PID controller.

This implementation is the most accurate with respect to the one present in Hannes; however, it caused a considerable increase in computational costs, thus leading us find an alternative solution.

For this reason, we decided to control the motor pulley in torque instead of position.

The torque actuating the motor pulley is measured from the ideal torque sensor positioned after the first reduction stage, in place of the ideal rotational motion sensor.

The second reduction stage, along with the rotational motion sensor modelling the encoder, were omitted as well, since the motor pulley rotational angle is measured directly from the revolute joint position sensor.

This angle is then divided for the reduction factor before being subtracted to the reference angle, generating the error in position.

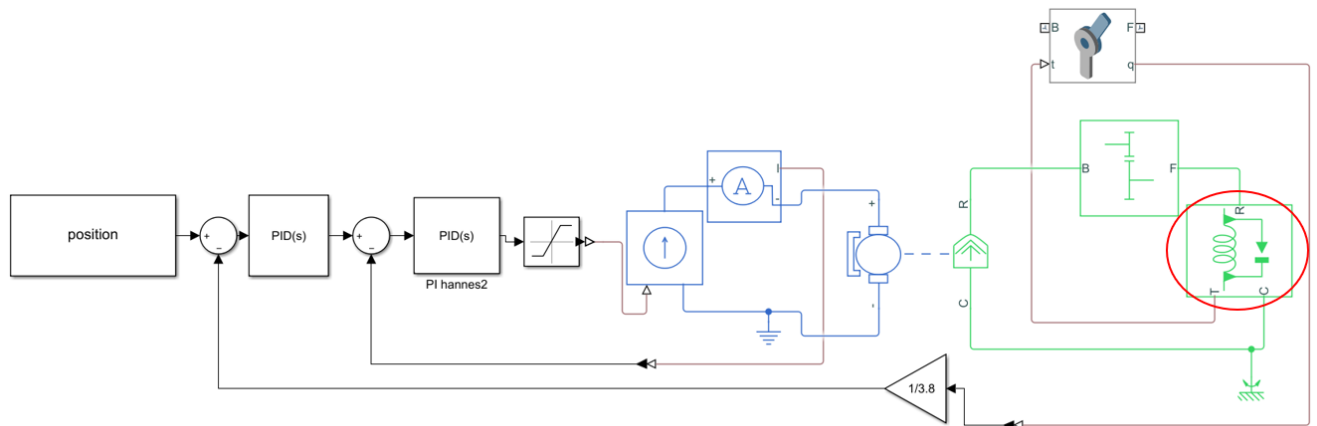


Figure 58 Simplified model of motor and PID controller system.

(In red) Ideal torque sensor.

2.1.3.6 Double PID control

The control is realized through two PID's (Figure 58). As stated in 2.1.2.4, the first one takes as input the error between the reference position and the measured encoder position and gives as output a current. This current is compared to the one measured through the current sensor placed after the controlled current source, generating an error which is the input for the second PID.

After the second PID there is a saturation block to limit the input current to the upper and lower saturation values of 4.6 A thus avoiding the generation of enormous torque when the fingers hit objects.

The proportional, integral and derivative parameters of two PID were set according to the constants chosen for the real hand Hannes.

2.1.3.7 Frictions estimation

Mechanical systems are subjected to nonideality. In the actual hand Hannes there are frictions at the level of each joint, rotating pulley and mechanical pin but also along the guides for the two carts of the differential mechanism.

In order to reproduce a realistic model of the hand, it was necessary to model the nonideality as well.

This was a challenging step due to the impossibility to measure the actual frictional components from Hannes. For this reason, we designed an experimental protocol to estimate those values.

The idea was to use previous acquisitions taken from Hannes: The REHAB team provided us with information about the motor current and the relative angular excursion of each finger

during closure. The current profile and an example of angular excursion are reported in Figure 59.

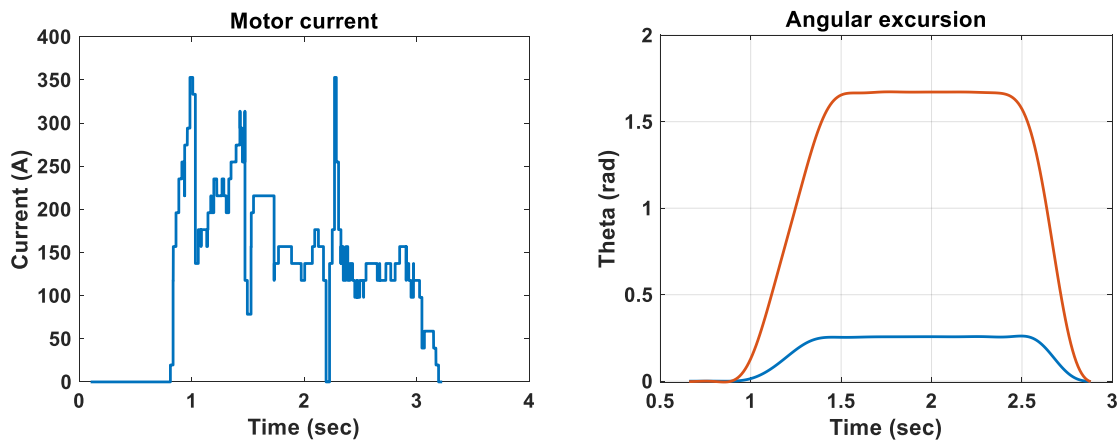


Figure 59 Example of motor current

(On the left) Motor current profile. (On the right) Middle finger phalanx (in red) and middle (in blue) parts angular excursion associated to the motor current.

Hannes closure time (t_H) in response to a given torque profile was derived from this data. Consequently, we set the friction coefficients at the level of each joint and carts' guides so that t_M (closure time of the model) would match t_H .

Due to the limitations of the “Belt and Cables” library it was not possible to characterize the mechanical pins and wires in terms of frictions.

The idea was to distribute the total frictional contributions between the different fingers joints and the cart guides.

The frictions in Simscape were modelled by using the blocks reported in Figure 60 respectively for the revolute and the prismatic joints, found in the “Multi-body Multiphysics” library.



Figure 60 Blocks from Multi-body Multiphysics library:

(On the left) Rotational friction, (On the right) Translational friction

The coefficients to set inside these blocks represent [61]:

- The “Breakaway Friction Torque”: the sum of the Coulomb and the static frictions.
- The “Breakaway Friction Velocity”: the velocity at which the Stribeck friction is at its peak. At this point, the sum of the Stribeck and Coulomb friction is the Breakaway friction force.
- The “Coulomb Friction Torque”: the friction that opposes rotation with a constant torque at any velocity.
- The “Viscous friction coefficient”: Proportionality coefficient between the friction torque and the relative angular velocity.

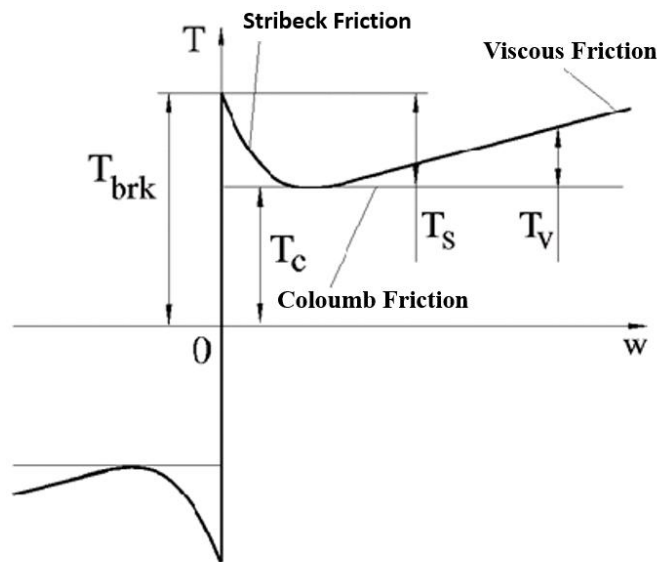


Figure 61 Variation of the friction torque wrt angular velocity:

Definition of Stribeck, Viscous and Coloumb frictions

2.1.3.8 Object interaction

An interesting feature of Simscape is the possibility to model the contact forces acting on solid blocks, without which the two objects would co-penetrate if moving toward each other.

According to how the contact forces are modelled, the behaviour of the two solids changes consequently. In particular, the normal f_n and friction f_r are forces which the two bodies exert on each other when in contact.

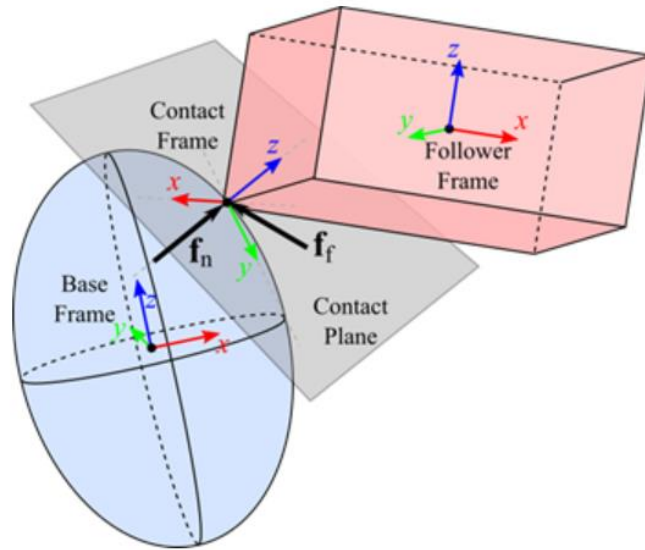


Figure 62 Interaction between Base Frame and Follower Frame

The **Solid Contact Force** is the block which models the forces acting between the Base Frame and the Follower Frame using the penalty method.

The penalty method allows the bodies to penetrate a small amount to compute the forces.



Figure 63 Solid Contact Force block

The Normal Force f_n is computed using the spring-damper system and is applied based on the penetration depth (**Stiffness**) and the penetration velocity (**Damping**), meanwhile the **Transition Region Width** specifies the transitional region to the force equations, at which end the full stiffness and damping are applied.

The Friction Force f_f lies in the contact plane and opposes the relative tangential velocities between the two bodies near the area of penetration.

In Solid Contact Force block it's possible to model the friction force as Smooth Stick-Slip which produces more realistic contact dynamics. If this feature is selected, the friction force can be characterized in terms of coefficient of Static friction (ratio between friction force and normal force) and coefficient of Dynamic friction.

Through this block it's possible to measure the **separation distance** between the two solid bodies, the **Normal force** and the **Friction force magnitude**.

In our work we exploited the Solid Contact Force block to model the interaction between the phalanges of the fingers and objects of different properties.

By changing the Stiffness parameter, it was possible to model rigid and soft objects. Particularly, we decided to use as reference a hand dynamometer and a ball for the rigid and soft object respectively (Figure 64).



Figure 64 (On the left) ball, (on the right) hand dynamometer

Not knowing a priori the stiffness of these objects, we characterized the Solid Contact Force block by trying to match model parameters and the same ones measured on Hannes (see 3.1.4)

This was the first step to implement the object stiffness recognition further discussed in the next paragraph.

2.1.4 New implementation: Object stiffness recognition

As mentioned in 1.1, nowadays the prosthetic hand Hannes does not provide any haptic feedback to the user.

Due to project specifications, INAIL requested for the prosthetic system to be as affordable as possible. Therefore, the implementation of sensory feedbacks would lead to an increase in both weight and costs of Hannes.

However, the lack of tactile sensation represented an important setback during Cybathlon 2020 competition at which Hannes participated, where we attended as part of Rehab team.

Cybathlon is a worldwide competition organized by ETH where different countries competed against each other, completing everyday tasks with the help of State-of-Art assistance systems.

The race for prosthetic hands was composed by 6 tasks in total and the pilot was asked to finish all of them as soon as possible.



Figure 65 Hannes pilot at Cybathlon 2020

The Rehab team performed very well at 5 out of 6 tasks, highlighting the versatility and precision of Hannes in performing difficult tasks such as hammering, cutting bread, and hanging clothes.

The most challenging task was to recognise the shape and stiffness of several objects without the support of visual feedbacks: the pilot had to insert the prosthetic hand into different black boxes containing the different objects and pair them to reference ones.

This kind of ability can be improved with lots of exercise but unfortunately due to COVID-19 situation it was not possible to train the pilot properly, leading to the failure of this task and the loss of points.

From this experience we decided to implement an object stiffness recognition to help the user differentiate rigid and soft objects grasped. In addition, as mentioned in 1.1, prostheses

that provide haptic feedback in closed-loop to their users, help subjects to better feel the connection with the non-physiological hand and promote the dexterity during grasping tasks.

Our challenge was to implement an object-stiffness recognition system without the use of any force sensor but only exploiting the already existing sensors on the hand: the current sensor and the encoder.

It is important to stress how the development of the Multi-body model of Hannes gave us the opportunity to thoroughly study and understand the mechatronic working principles of the hand, making the model a powerful tool to inspire new ideas.

2.1.4.1 Exploited sensors

The choice of the current was given by the following consideration: As previously introduced in 2.1.2.4, while performing a closure, the interaction between the fingers and an object causes for the motor current to increase proportionally to the stiffness of the object. This means that a correlation exists between the amplitude of the current and the stiffness of the interface.

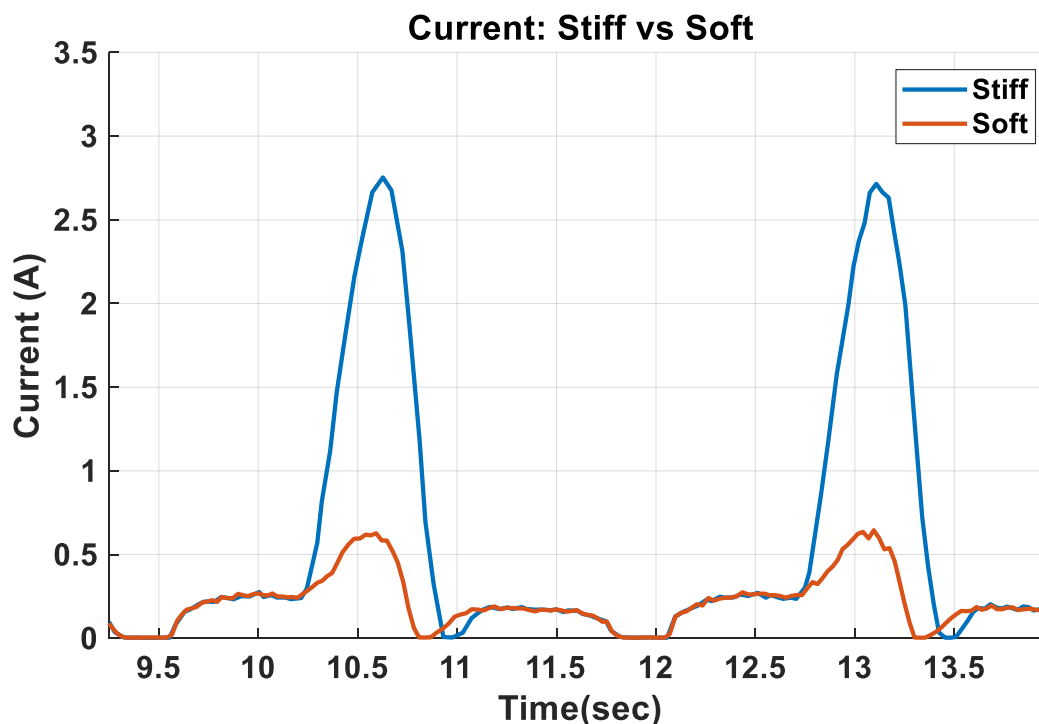


Figure 66 Hannes motor current: Soft objects (blue line) vs Rigid object (red line)

The previous graph (Figure 66) represents the behaviour of the motor current of Hannes when interacting with an object of low stiffness (blue line) and high stiffness (red line). In the first case a soft ball was used, meanwhile in the second case a hand dynamometer was used.

The chosen opening-closing command was a sinusoidal profile reference position. Currents were measured as explained in 2.2.

As we can notice, the curves are characterized by three phases:

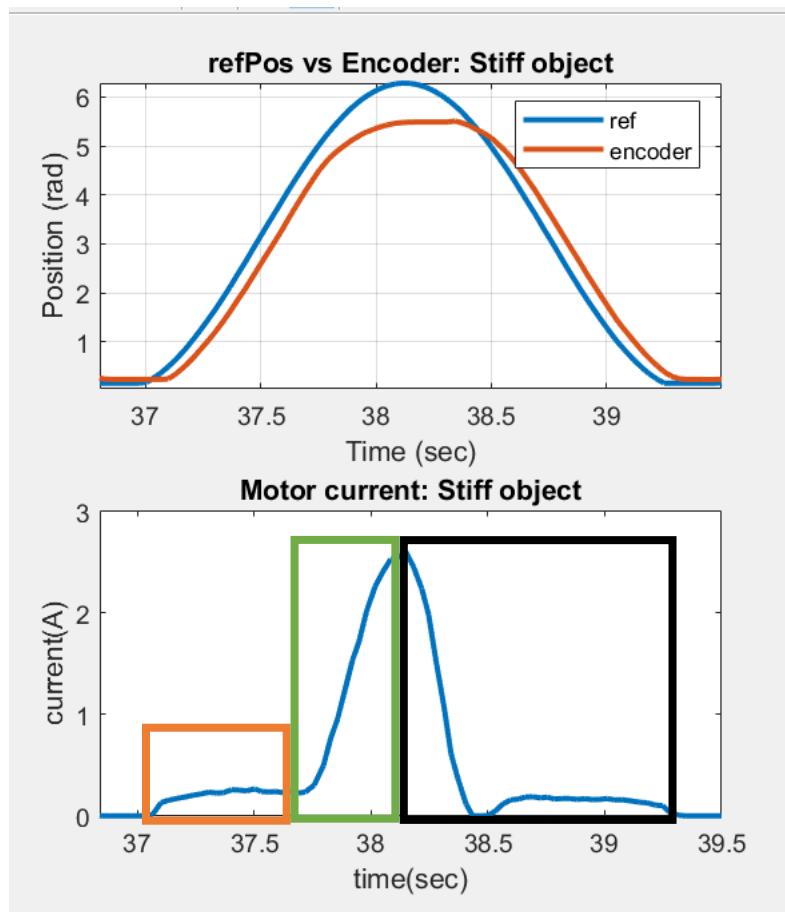


Figure 67 (Up) Graph of RefPos vs Encoder output, (Down) Hannes motor current:

Void current (in orange), Grasping (in green), Opening (in black)

1. Void (in orange). It represents the free bending of fingers. In this phase the error between the reference position and actual position of the motor pulley is very small.
2. Grasping (in green). In this phase the fingers hit the object and current starts rising considerably up to a maximum value. The error between reference position and the output of the encoder rises too.
3. Opening (in black). The reference position commands for the hand to re-open. Current gradually decreases to zero. An additional positive curve is registered which represents the motor current flowing on the opposite direction, needed to perform the

re-opening of the hand. The positive sign is due to the measurement of the signal after a rectification stage.

The main differences between the two curves in Figure 66 are the maximum amplitude of the current and the steepness of its rising edge, both higher when an interaction with a rigid object occurs.

The same behaviour was registered also in the Hannes model where we modelled the ball and the hand dynamometer.

The comparison between the real Hannes motor current and the current generated by the model is further discussed in 3.1.4.

Apart from the current, we decided to exploit the information given by the reference position provided to the control system, and the output position measured by the encoder too. We wanted to enforce the difference in behaviour of the Hannes hand when interacting with objects of different stiffness. Particularly, when the hand interacts with a rigid object the error between the RefPos and the output of the encoder is higher than the error obtained while grasping a soft object.

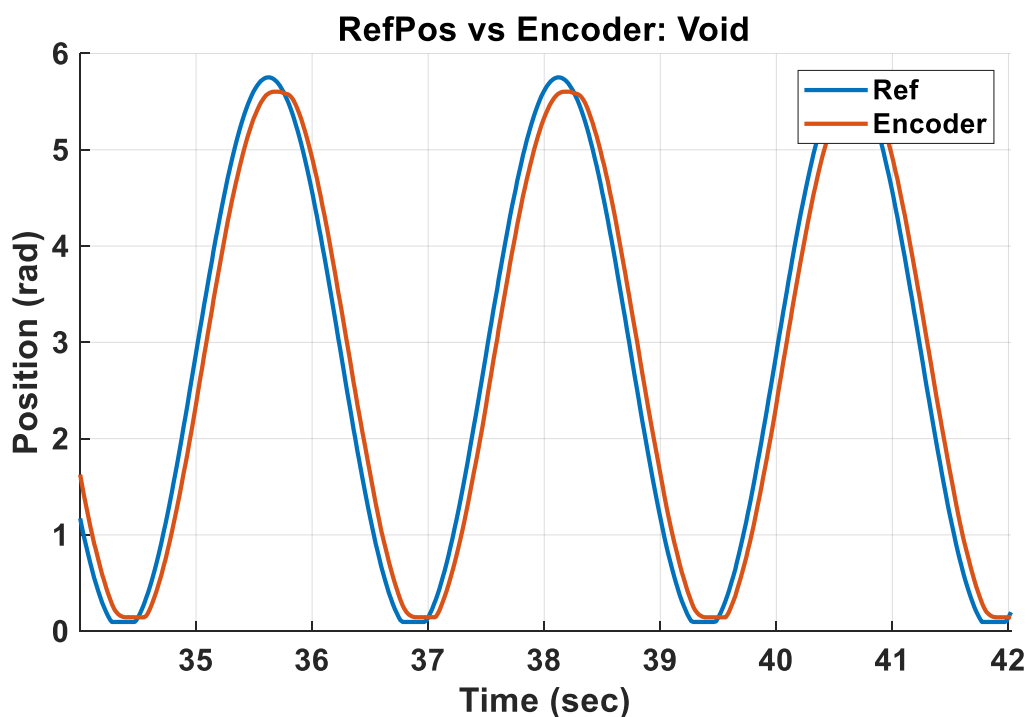


Figure 68 RefPos vs Encoder output: Void closure

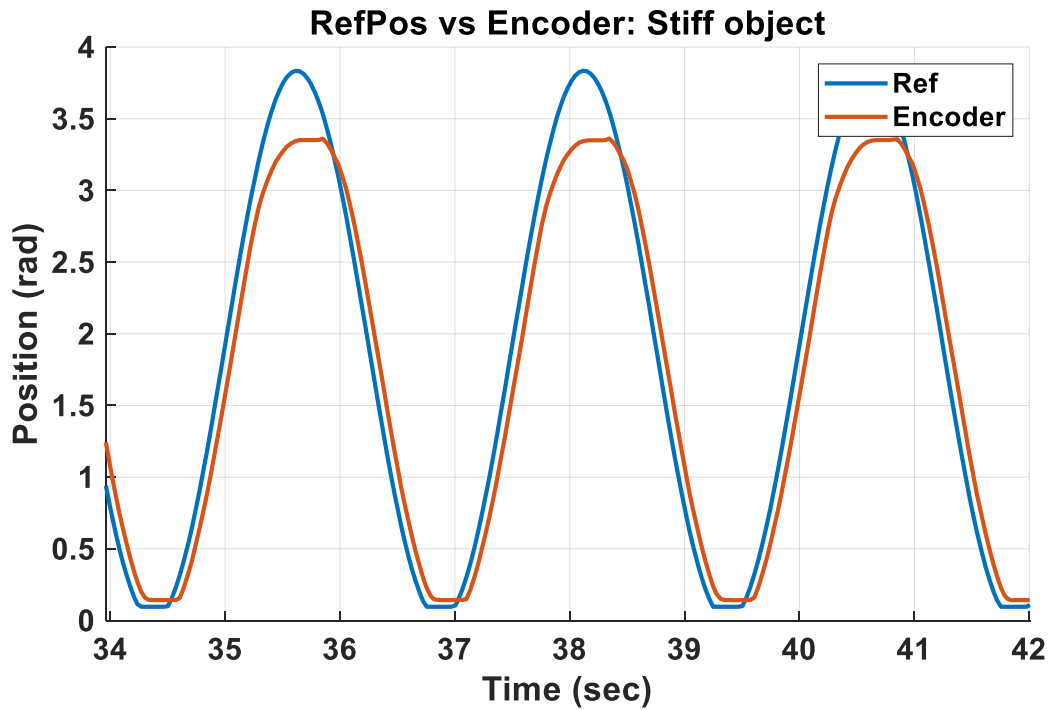


Figure 69 RefPos vs Encoder output: Stiff object

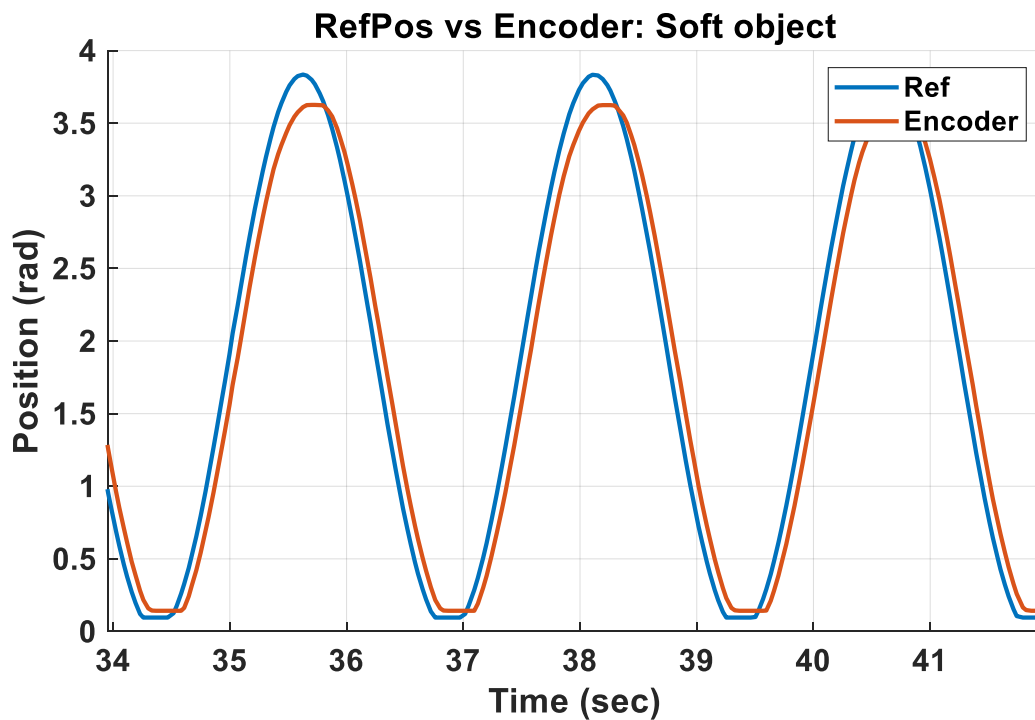


Figure 70 RefPos vs Encoder output: Soft object

The idea was to exploit the Multi-body Model of Hannes to extract valuable information and use them as inputs for machine learning tools for the implementation of stiffness recognition: We created a matrix with four columns (training dataset), where the first three contain the data measured from several model acquisitions (motor current, reference position,

measured position) and the last one includes the labels. Particularly these data are the result of different kind of interactions: different objects shapes (cylindrical, prismatic and spherical) and different objects positions. The stiffness assigned to these objects were the ones used to model the hand dynamometer and the ball.

<u>current</u>	<u>refPos</u>	<u>encoder</u>	<u>class</u>
297.8250	2.5285e+04	2.6130e+04	0
298.1810	2.5288e+04	2.6135e+04	0
298.5396	2.5290e+04	2.6140e+04	0
298.8995	2.5293e+04	2.6145e+04	0
299.2604	2.5295e+04	2.6149e+04	0
299.6218	2.5298e+04	2.6154e+04	0
299.9835	2.5300e+04	2.6159e+04	0
300.3454	2.5303e+04	2.6164e+04	1
300.7073	2.5305e+04	2.6168e+04	1
301.0692	2.5308e+04	2.6173e+04	1
301.4308	2.5311e+04	2.6178e+04	1
301.7922	2.5313e+04	2.6183e+04	1
302.1531	2.5316e+04	2.6187e+04	1

Figure 71 Training Dataset example:

The first three columns are the input data, the last one the associated classes

We appointed three classes: **0**, identifying the void closure, **1**, identifying the interaction with a rigid object, and **2**, the interaction with a soft object.

In particular, we refer to void closure as the data obtained from a void acquisition and the parts corresponding to void and opening during a grasp as shown in Figure 67. In order to identify these parts, we defined a current threshold of 0.3A below which every point was classified as 0. Instead, all the points above that threshold were classified as 1 or 2, according to the interaction type.

2.1.4.2 Tool MATLAB: classification learner

Machine learning is based on the idea of learning from experience. It employs computational methods to learn from the available data, consequently bigger datasets improve the algorithms performance.

In machine learning there are two types of technique: supervised and unsupervised learning.

Supervised learning algorithms require a known set of input data and known responses to those data to train a model, capable of generating reasonable predictions in response to new data. Supervised learning uses classification and regression techniques to generate predictive models.

Classification techniques generally predict categorical responses (e.g., if a tumour is benign or not), while regression techniques are used for continuous responses (e.g., changes in temperature).

On the other hand, unsupervised learning takes as input a dataset without labelled responses and finds hidden patterns or intrinsic structures in it. The most used unsupervised learning technique is clustering which allows to find hidden patterns or groupings in data [62].

For our application we chose to use supervised learning and in particular the classification technique since we needed to obtain categorical responses.

We employed the MATLAB app, Classification Learner, for this purpose.

The Classification Learner app, receiving in input the training dataset described before, automatically trains a set of models, thus giving the possibility to choose the best one. This is a great advantage since the choice of the right model depends on different factors. For example, highly flexible models frequently overfit the data and they are more subjected to noise, while simpler models might have lower accuracy.

Once inserted the input matrix, it is possible to choose among three validation methods to examine the accuracy of the models [63].

1. Cross-validation: it partitions the dataset in a number of folds that it is possible to choose. For each fold it trains the model using the out-of-fold data and it assesses the model performance with the in-fold observations.
2. Holdout validation: it allows to specify a percentage of the dataset to use as test set. Consequently, the model is trained on the training set and it is evaluated on the test set. This method is suitable only in case of large datasets.
3. No validation: it does not assure protection against overfitting.

We selected the cross-validation method and we chose the default number of folds (five).

Considering that we did not know which model type could be the best for our application, we trained all the available ones in the app such as decision trees, discriminant analysis, support vector machines, logistic regression, nearest neighbours, naive Bayes, and ensemble classification.

The model Ensemble Bagged Trees resulted as the best in terms of training accuracy (100%).

Ensemble is a supervised learning algorithm which trains a set of classifiers in parallel. Each classifier makes a prediction about the new data which is finally labelled with the prediction with the highest number of votes. This method is more accurate, but it requires a higher computational cost.

There are different types of ensemble algorithms. The one used for our classification is the bagged trees.

Bagged is the abbreviation of bootstrap aggregating and it involves that each classifier's vote has the same weight. In particular bagged trees employ the random forest algorithm which implies that the classifiers are decision trees. The decision trees are generated from a random selection of attributes at each node to determine the split.

After training the ensemble classifier, we exported and test it first on new data acquired from the model.

Then we tested the classifier on data acquired directly from Hannes, as explained in 2.2.

2.2 Experimental set-up and validation

In order to control Hannes hand, a specific set-up and interface was used.

First of all, we decided to connect the Hand directly to the main power supply, which guarantees a constant voltage supply of 12V to the electronics. Then Hannes was connected to a PC (HP Pavilion Power Laptop 15 i7 8Gb Ram) through an USB cable.

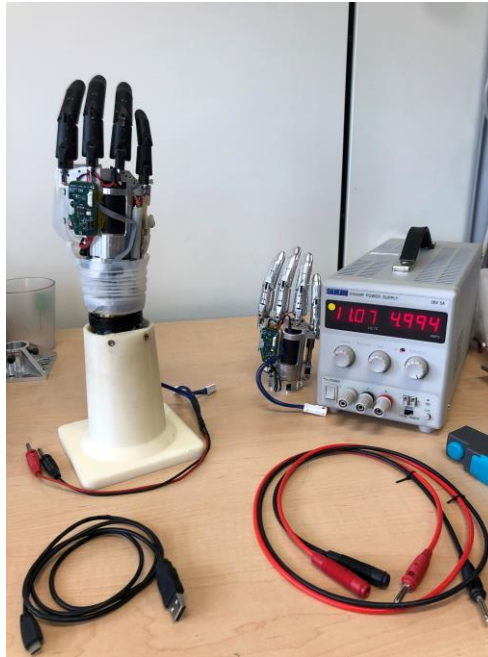


Figure 72 Experimental Set-up:

Hannes hand, constant voltage power supply and USB cable

The Hannes control GUI consists of different features thanks which it's possible to control Hannes opening/closure (either manually or setting a given waveform as reference position), and record different parameters such as motor current, battery current, encoder output and reference position.

For the acquisitions with the hand dynamometer, an already existing testbench was used.

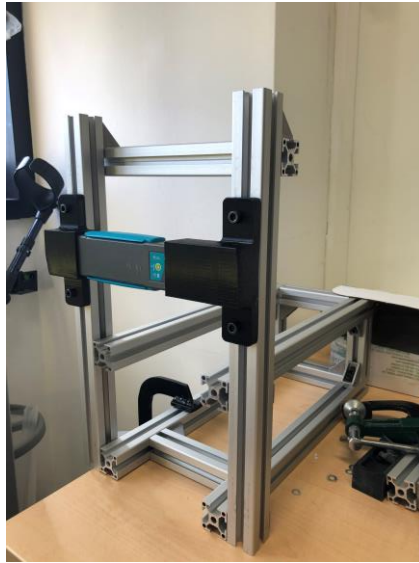


Figure 73 Testbench:

Metallic support for the hand dynamometer

The structure is a custom-made system designed to support the Vernier hand dynamometer. It is composed by movable parallel arms and a two-cuffs system acting as holder.

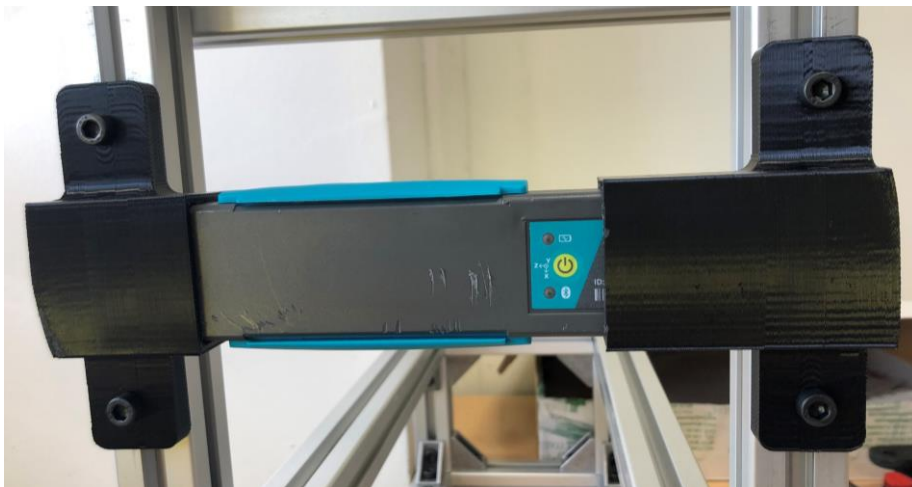


Figure 74 Two-cuffs system

Vernier hand dynamometer is a tool which allows to measure grip and pinch strength and to perform muscle fatigue studies.

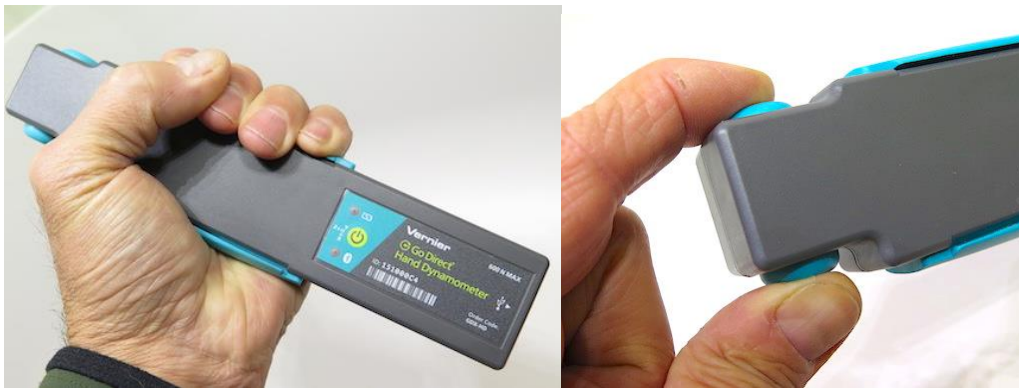


Figure 75 Vernier hand dynamometer

When applying a pressure on the sensor, the upper light blue plate deforms with respect to the lower one. By means of a dedicated GUI connected by Bluetooth, it's possible to register the force applied on the sensor over time.

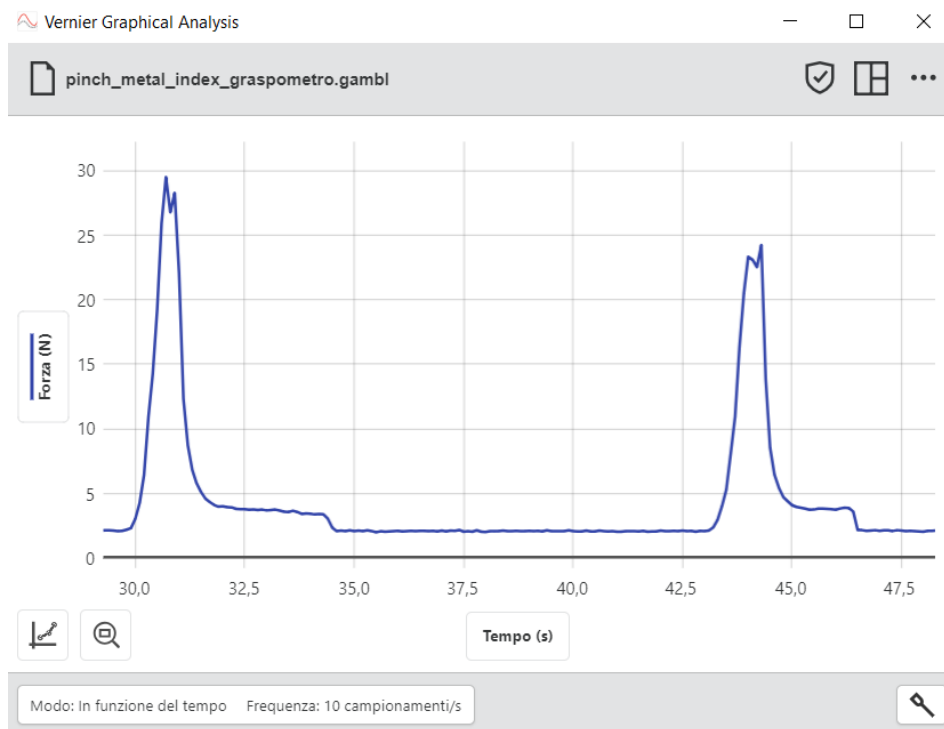


Figure 76 GUI Vernier Hand Dynamometer

Once the most feasible position of Hannes was chosen, the dynamometer was fixed to the support so that while performing a closure only the distal phalanges of the four fingers would impact on the upper plate of the sensor. The Thumb was stuck under the sensor, in contact with the reference plate.

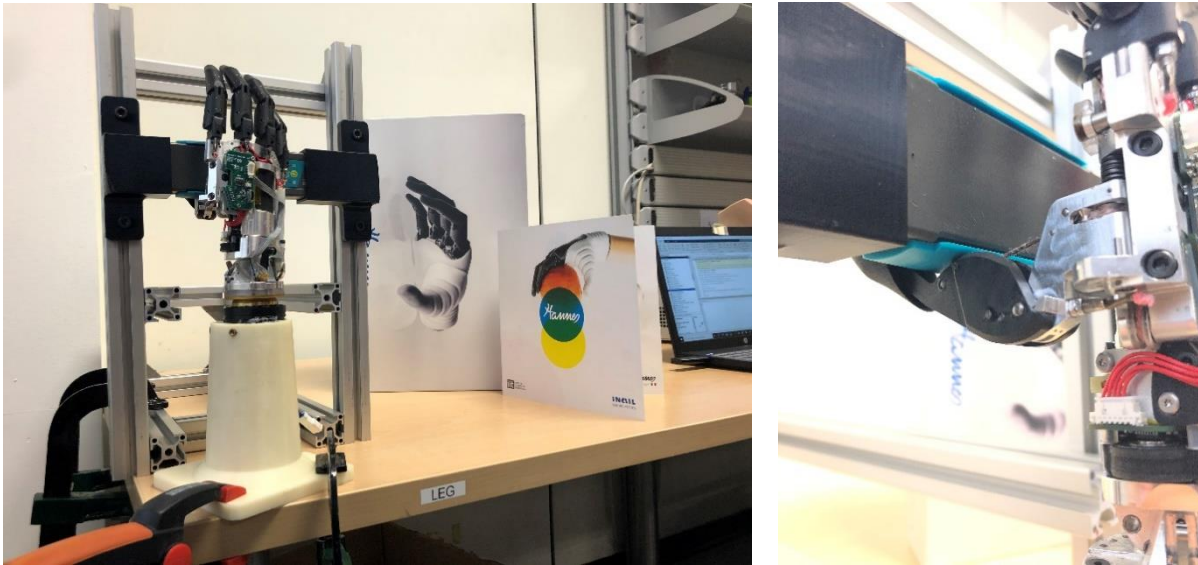


Figure 77(On the left) Hannes and testbench, (on the right) Position of Hannes thumb

We performed thirty-nine closures of the hand on the dynamometer and we measured, with a frequency of 30 Hz, the output of the encoder, the motor current and the battery current generated by Hannes system. At the same time we acquired the forces acting on the sensor with a frequency of 100 Hz.

The input in position given to Hannes through the GUI was a sine wave with a frequency of 4 Hz and an amplitude of 40000, corresponding to a partial closure.

We reproduced these same conditions on the multi-body model in Simscape and we compared the output measured from Hannes to the same ones generated by our model (3.1.4).

For the validation of the classifier (2.2.1) we acquired Hannes interactions with ten different objects reported in Figure 78. Four of them are considered rigid and the others soft. In particular, the distinction between soft and rigid objects stands in the amount of the object deformation: an object subjected to a visible deformation, which results to be greater than 5 mm, is considered soft, otherwise it is rigid.



Figure 78 Object employed to test the classifier

In this case the reference position, given as input to Hannes, consisted of thirty-nine sine waves with a frequency of 4 Hz whose amplitude was chosen, for each object, in a way to avoid damages to the prosthesis.

In the end we obtained thirty-nine acquisition for each object, which were used to test the classifier.

2.2.1 Validation metrics

In order to validate the model (3.1), we compared the results of the analytical calculations and of the acquisition obtained from Hannes experimental set ups with those generated by the multi-body model. We characterized the differences in terms of relative error % by using the following formula:

$$\varepsilon\% = \frac{V_d - V_m}{V_d} \cdot 100$$

Where V_d corresponds to the desired value, which is either the analytical value or Hannes experimental value, while V_m represents the measured value correspondent to the multi-body model output.

Particularly, concerning the comparison between Hannes and the model motor currents, we computed the mean signal out of thirty-nine motor current signals acquired from Hannes. In this way we could compare this mean signal with the one measured from the model.

Instead for the validation of the optimized solutions (3.2), we employed the root mean square error (RMSE) as evaluation metrics:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}}$$

\hat{y} = estimated value (new implementation results);

y =original value (original implementation results);

N = number of samples.

The classifier performances were evaluated on ten different objects as explained before. The acquisition output data were used to create a matrix (test dataset) similar to the one in Figure 71. However, this matrix was composed of three columns, thus excluding the last one containing the labels. Then this matrix was provided to the classifier which assigned a class for each sample of the data. The resulting classes were compared to the correct ones and the number of right classifications was computed.

The metrics used to evaluate quantitatively the classifier performance was F1Score which can be expressed as follow:

$$\begin{cases} PR = \frac{nP}{nP + nFN} \\ RE = \frac{nP}{nP + nFP} \\ F1Score = 2 * \frac{PR * RE}{PR + RE} * 100 \end{cases}$$

where nP is the number of true positive, nN the number of true negative, nFP the number of false positive and nFN the number of false negative, PR is the *Precision* and RE the *Recall*.

The choice of F1Score instead of accuracy was due to the lower robustness of the latter for classes that do not present a perfect symmetrical cardinality [64].

To compute the F1Score, the array of predictions and the true classes are windowed with a floating window of 200ms and an overlap of 50ms. For each window the resulting class is selected executing the mode.

CHAPTER 3 Results

We divided this section in three main chapters: In 3.1 we discussed the results inherent to the validation trials we performed to assess the accuracy of the model.

Due to heavy computational costs some optimizations needed to be performed, we presented the results in the section in 3.2.

Lastly, in 3.3 we showed the accuracy of the proposed classifier.

3.1 Validation

3.1.1 Drop time

As previously described in 2.1.3.2, we verified the correct modelling of inertial characteristics of the four fingers (index, middle, ring and little) by computing a comparison between the drop time of each phalanx versus the drop time of a double inverted pendulum.

Keeping in mind that the maximum angular excursion for the proximal phalanx (we referred to it as *phalanx*) is 82° and for the distal phalanx (here referred as *middle*) is 50° , in the following table we reported the time instant at which each phalanx and middle reaches the upper limit:

	T_s (SEC)		T_A (SEC)		E_{REL} %	
	Phalanx	Middle	Phalanx	Middle	Phalanx	Middle
INDEX	0.159	0.077	0.163	0.078	+2.5%	+1.3%
MIDDLE	0.157	0.077	0.161	0.077	+2.5%	0%
RING	0.159	0.08	0.163	0.078	+2.5%	-2.56%
LITTLE	0.164	0.064	0.161	0.068	-1.9%	+5.8%

Table 2 drop time and relative error % results:

Comparison between simulation (T_s) and analytical (T_A) drop times of the phalanx (in blue) and the middle (in green) for each finger. The maximum relative error percentage was found for the middle part of the little finger (+5.8%) meanwhile the lower relative error percentage was found for the middle part of the middle finger (0%).

The higher relative error was found for the little finger *middle* (+5.8%) and the lower one for the middle finger *middle* (0%).

3.1.2 Return-wire mechanism

Once the correct inertial characterization of fingers was assessed, the next step was the implementation and validation of the return wire.

As already explained in 2.1.3.3 we verified whether the minimum external force required to set the distal phalanx of each finger in movement was the same as the one computed analytically.

The following table (Table 3) shows the results of the dynamic analysis performed:

FINGER	F _{EXT} (S)	F _{EXT} (A)	E _{REL} %
THUMB	3.51 N	3.66 N	+4.1 %
INDEX	1.65 N	1.66 N	+0.6 %
MIDDLE	2.46 N	2.42 N	-1.7 %
RING	1.83 N	1.83 N	0 %
LITTLE	0.99 N	0.99 N	0 %

Table 3 External forces and relative error % results:

Comparison between simulation (S) and analytical (A) forces for each finger. The higher relative error % was found for the Thumb and the lower one for both ring and little finger.

The maximum relative error found was of 4.1% for the thumb meanwhile the smaller errors (0% circa) were found for the index, ring and little fingers.

3.1.3 Closure time: Hannes vs Model

Once the master-slave and return-wire mechanism were implemented we noticed how the closure time of the model was very short with respect to the actual closure time of Hannes (T_{model} was about a fraction of second meanwhile the T_{Hannes} around 1 second).

This behaviour could be easily explained by the absence of any frictional component in the Hannes model system.

Once the nonideality was modelled in Simscape (see 2.1.3.7), we inquired the difference in closure time by comparing the encoder output of Hannes with respect to the encoder output in the model given the same reference position.

In Figure 79 we see in green the reference position in input to the system, in blue the motor pulley position measured by the Hannes encoder and in red the motor pulley position measured in the Simscape model. We can notice how the red curve and the blue ones are comparable in terms of amplitude and rising/falling slope: the closure time in the real hand is 1.189 sec and in our model is 1.180 sec with a relative error % of +0.79%.

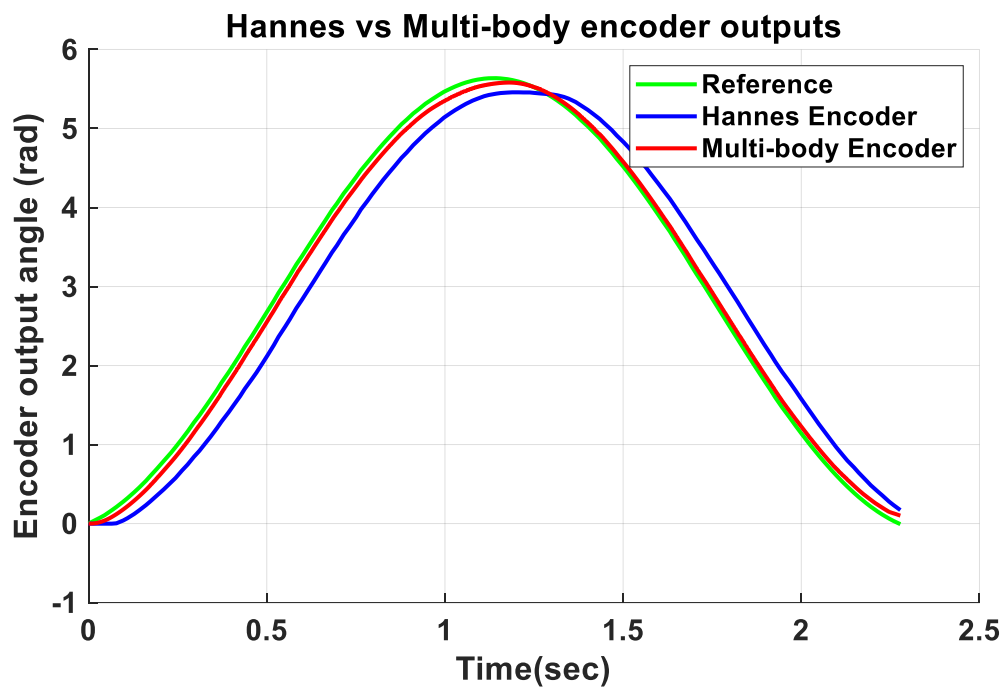


Figure 79 Comparison between reference input, Hannes and Model encoder outputs

the relative error percentage between the closure time of Hannes and of the Multi-Body model is +0.79%.

3.1.4 Motor current: Hannes vs Model

In this section we compared the motor currents generated by our model in Simscape and the real current present in Hannes.

By performing tests in different prosthetic hands, we noticed how the behaviour of each of them was unique with respect the others. This is clearly visible in Figure 80 where in blue we can see the motor currents generated by different Hannes hands during a void closure, meanwhile in red there is the motor current generated by our model. After a transient, the curve reaches a plateau of 0.25 A, inside the blue area.

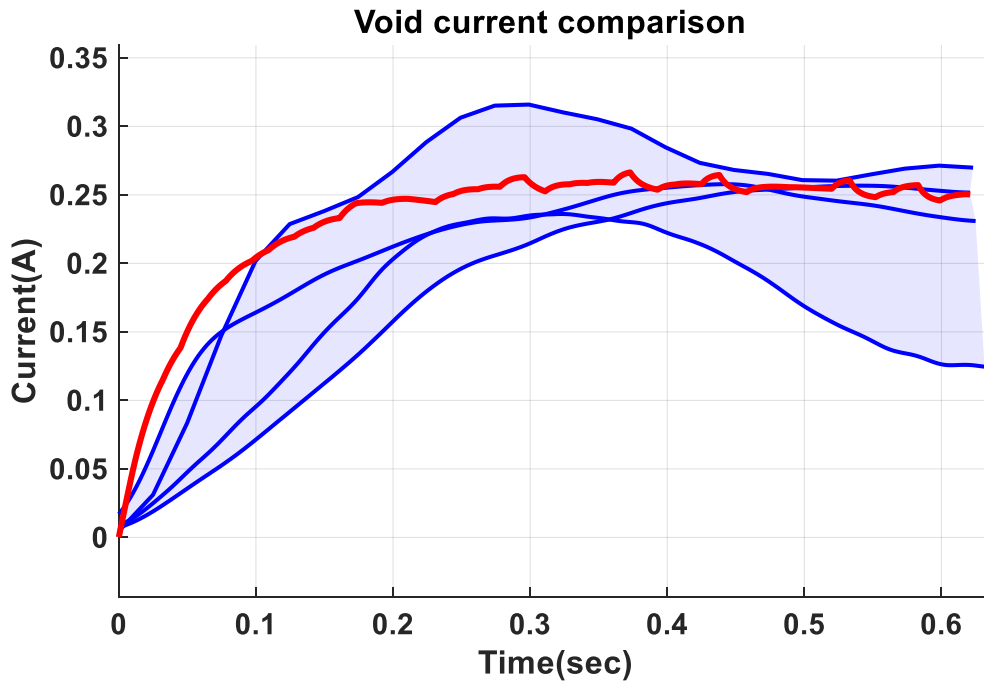


Figure 80 Void currents:

Comparison between the void motor currents (in blue) of different Hannes hands and the model motor current (in red). This graph shows how each Hannes hand behaves differently from another, meanwhile the Multi-body model shows an average behaviour between them.

Then we compared the motor current generated by Hannes and the model during different grasping phases. Particularly, as anticipated in 2.1.3.8, we tried to model the chosen rigid and soft objects in the Simscape environment by changing the stiffness parameter in the Solid Contact Force block (Figure 63). For the considerations previously presented in 2.1.4, the motor current behaviour changes according to the characteristics of the grasped object. For this reason, we selected the stiffness of the two objects which could make the motor current of Hannes and of the model as similar as possible.

In Table 4 we reported the comparison between different parameters where c stands for closure and o stands for opening.

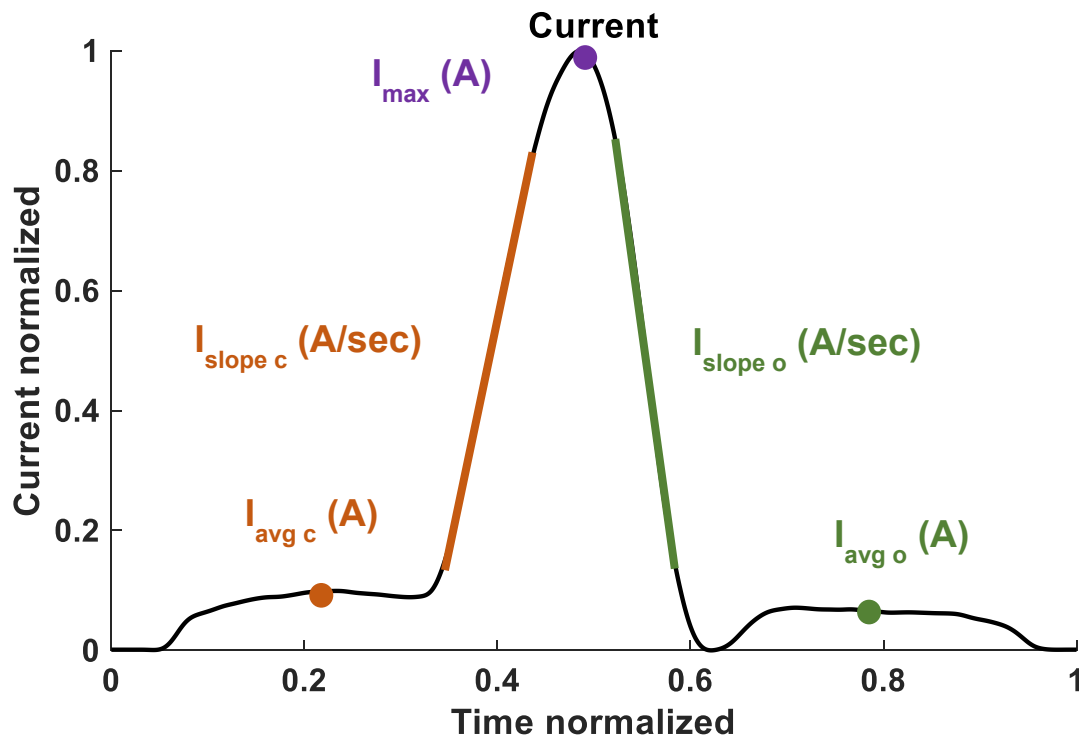


Figure 81 Normalized current scheme

The chosen parameters are: the average void currents ($I_{avg\ c}$, $I_{avg\ o}$) and the slopes of the current ($I_{slope\ c}$, $I_{slope\ o}$) during closure and opening of the hand, and the maximum current I_{max} .

TEST	$I_{AVG\ C}$ (A)		$I_{SLOPE\ C}$ (A/SEC)		I_{MAX} (A)		$I_{AVG\ O}$ (A)		$I_{SLOPE\ O}$ (A/SEC)	
	Rigid	Soft	Rigid	Soft	Rigid	Soft	Rigid	Soft	Rigid	Soft
HANNES	0.236	0.254	7.17	1.04	2.59	0.577	0.167	0.169	-11.1	-2.11
MULTI-BODY MODEL	0.241	0.236	6.56	1.14	2.50	0.668	0.158	0.158	-6.9	-1.84
RELATIVE ERROR %	+2.1%	+7%	+8.5%	-9.6%	+3.5%	-16%	+5.4%	+6.5%	+38%	+13%

Table 4 Current parameters results

The maximum relative error percentage was found for the $I_{slope\ o}$ for the rigid body interaction (+38%), meanwhile the lowest one for the $I_{avg\ c}$ during Rigid interaction (+2.1%).

The maximum relative error % was found for the $I_{slope\ o}$ of the rigid object (+38%), meanwhile the smaller one for $I_{avg\ c}$ of the rigid object (+2.1%).

Then we compared the sum of the forces exerting by Hannes fingers on the hand dynamometer with respect to the ones of the model (Figure 82).

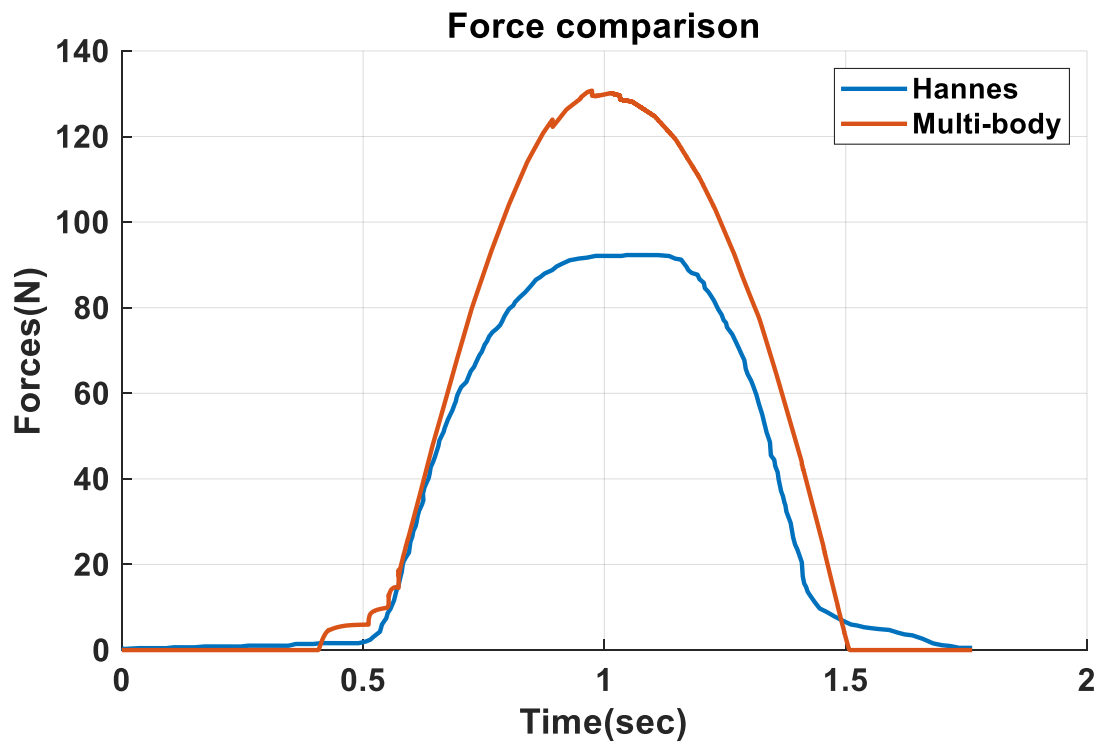


Figure 82 Forces exerted by fingers

The force exerted by Hannes fingers is in blue, meanwhile the force exerted by the Multi-body model fingers is in red.

The maximum value of the force exerted by Hannes is 92.3 N meanwhile the maximum value of the force exerted by the multi-body model is 130.7 N, with a relative error % of -16.7%.

3.2 Optimization

3.2.1 Torsional springs

As previously introduced in 2.1.3.3, the return wire mechanism implementation resulted in very high computational costs (Time of simulation about 12 hours). For these reasons, we looked for an alternative solution to speed up the simulation. By implementing pre-loaded torsional springs in place of the compression spring and wire system, the computational cost reduced significantly (Time of simulation about 28 minutes).

We wanted check whether with the new implementation the kinematic of the fingers was comparable to the model presenting the return wire mechanism.

The following graph (Figure 83) shows the comparison between torsional spring (TS, dashed line) implementation versus return-wire mechanism one (RW, stream line), where the angular excursion of the phalanx is reported in blue and the angular excursion of the middle in red.

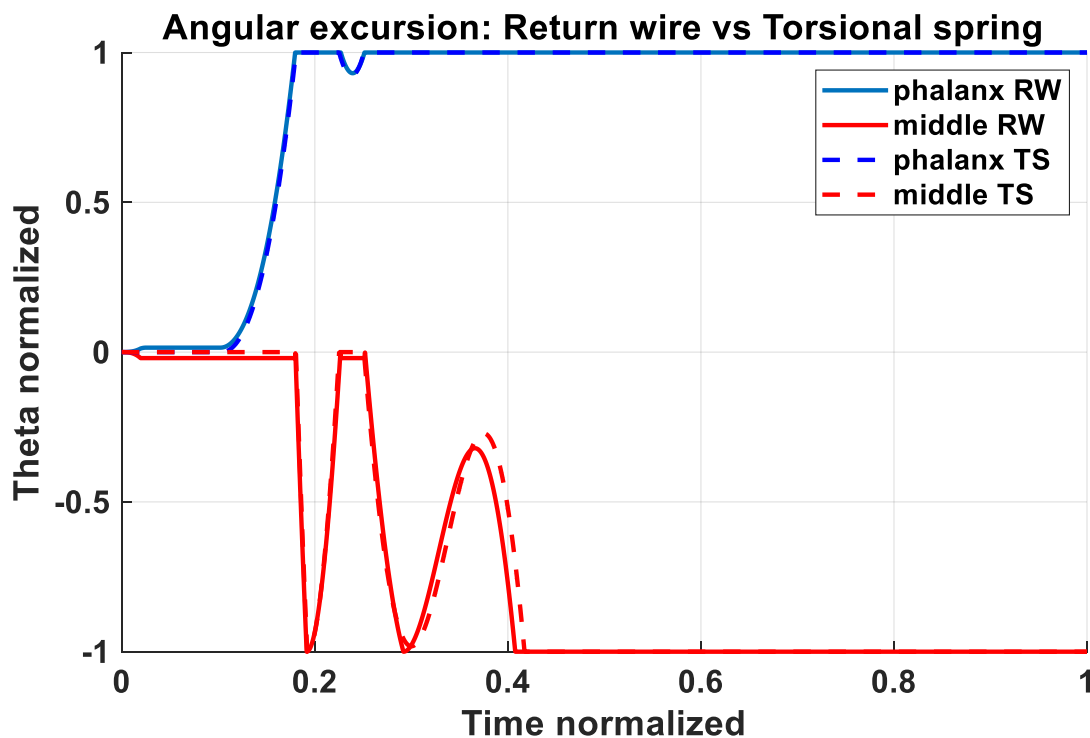


Figure 83 Normalized angular excursion RW vs TS scheme

The stream lines (in blue for the phalanx and in red for the middle) represent the return-wire implementation meanwhile the dashed lines (in blue for the phalanx and in red for the middle) represent the torsional spring implementation.

The angular excursion of the middle part is negative due to the selected reference system with respect to the digit rotates from 0° to -50° .

To assess the difference between the two curves we computed the RMSE for each phalanx of each finger.

FINGER	RMSE_{PHALANX} (RAD)	RMSE_{MIDDLE} (RAD)
THUMB		0.0493
INDEX	0.0112	0.0399
MIDDLE	0.0107	0.0399
RING	0.0099	0.0964
LITTLE	0.0242	0.0615

Table 5 RMSE RW vs TS results

All the RMSE in the table are lower than 0.10 (rad).

3.2.2 Force implementation

Another proposed implementation was the substitution of the pulleys-cable system with external forces directly acting on the proximal and distal phalanges. This solution made it possible for the Time of simulation to ulteriorly reduce at 1 minute.

The following graph (Figure 84) shows the kinematic behavior comparison of phalanx and middle parts between the two models (stream line = External force implementation, dashed line = pulley-cable implementation):

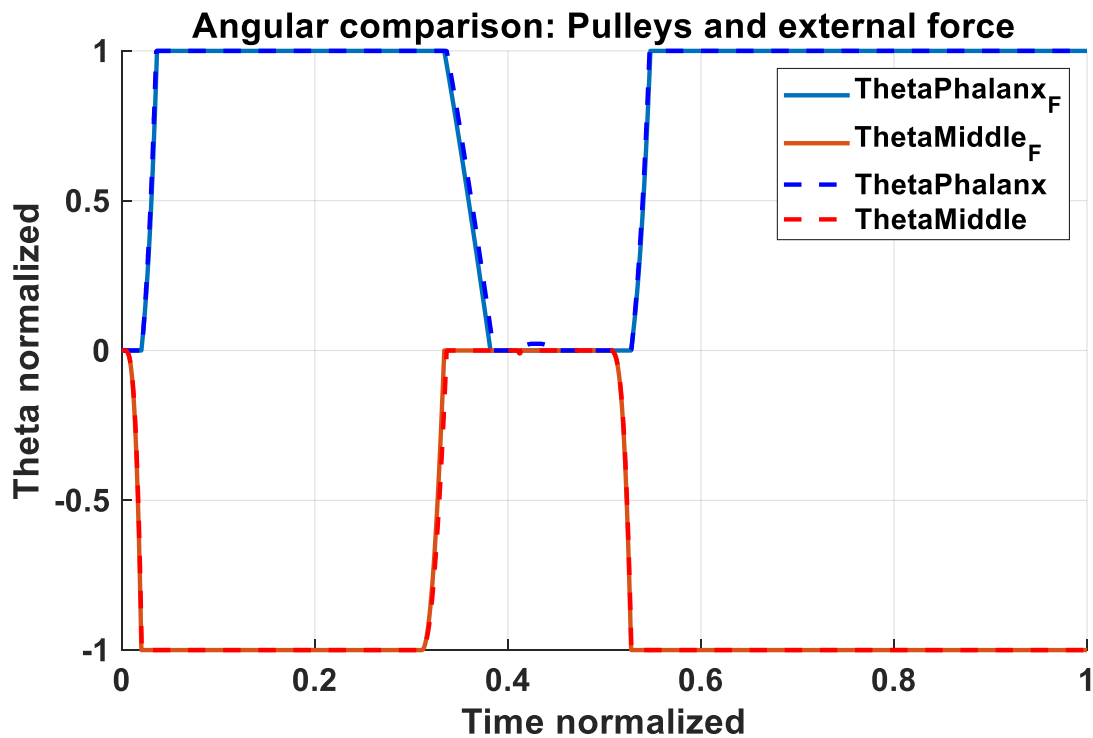


Figure 84 Normalized Angular excursion pulleys vs force implementation scheme.

The stream lines (in blue for the phalanx and in red for the middle) represent the external forces implementation meanwhile the dashed lines (in blue for the phalanx and in red for the middle) represent the pulley-cable implementation.

We computed the RMSE error between the curves:

FINGER	RMSE _{PHALANX} (RAD)	RMSE _{MIDDLE} (RAD)
INDEX	0.0256	0.0108
MIDDLE	0.0625	0.0478
RING	0.0751	0.0683
LITTLE	0.0595	0.0556

Table 6 RMSE pulleys vs forces results

All the RMSE in the table are lower than 0.10 (rad).

3.3 Application

3.3.1 Classifier

Once selected the classifier with the best result in terms of training accuracy, it was tested on new data acquired from the model by making the multi-body model interact with a rigid pentagon and a soft pentagon object.

In both cases the F1Score resulted of 100%.

Considering these results, we decided to test this classifier on the real data acquired from Hannes.

We tested it on four rigid and six soft objects and we evaluated the total performance of the classifier through the F1Score, which resulted of 87.84%.

We also computed the F1Score for each object, as reported in Table 7 and Table 8.





RIGID OBJECT	F1SCORE
HAND DYNAMOMETER 	97.3%
METAL BLOCK 	89.3%
BLACK SUPPORT 	92.3%
CRUTCH 	94.9%

Table 7 F1Score of rigid object

(On the left) Objects used for the rigid object interaction. (On the right) F1score computation for each object. The maximum F1score was found during the interaction with the hand dynamometer (97.3%), meanwhile the lowest F1score for the metal block (89.3%)

SOFT OBJECT	F1SCORE
BLUE BALL 	100%
GREEN BALL 	99.3%
BANANA PLUSH 	88.6%
SPONGE 	99%
ORANGE PLUSH 	69.5%
GRAPE PLUSH 	80.53%

Table 8 F1Score of soft objects

(On the left) Objects used for the soft object interaction. (On the right) F1score computation for each object. The maximum F1score was found during the interaction with the blue ball (100%), meanwhile the lowest F1score for the orange plush (69.5%).

CHAPTER 4 Discussion

The present study aimed to demonstrate how a multi-body model of a prosthetic hand is able to optimize the real system and to propose new implementations.

Once created as described in 2.1.3, we compared the kinematic and dynamic behaviour of our model with respect to actual Hannes.

For what concerned the evaluation of the inertial characteristics of the fingers, the results in 3.1.1 showed low relative errors percentage for each phalanx (<10%) with respect to the analytical computations. This suggests that the masses and inertial components of each finger were modelled correctly.

The return wire mechanism implementation was considered successfully validated obtaining a relative error percentage lower than 10% for each finger (see 3.1.2).

Comparing the closure time between the multi-body model and Hannes, we can conclude that there was no difference (relative error percentage of +0.79%). This result reflects overall a correct estimation of the friction components in the model. The two curves in Figure 79 present comparable amplitude and slope, the Hannes curve, though, results slightly shifted on the right along the x-axis. This is mostly due to the breakaway friction components of the pulleys system not perfectly modelled and the absence of gear-box inertia in our model.

For what concerned the evaluation of the motor currents generated by our model with respect to the real ones in Hannes, in Figure 80 we can observe how the slope of the red curve (Motor current of the model) during the transient phase is slightly higher than the slope of the other blue curves (Motor current of Hannes). This is due to the mechanical characteristics of Hannes: the master-slave system is characterized by pulleys and wires. During opening-closures of the prosthetic hand, the wire tends to lose as shown in Figure 85.

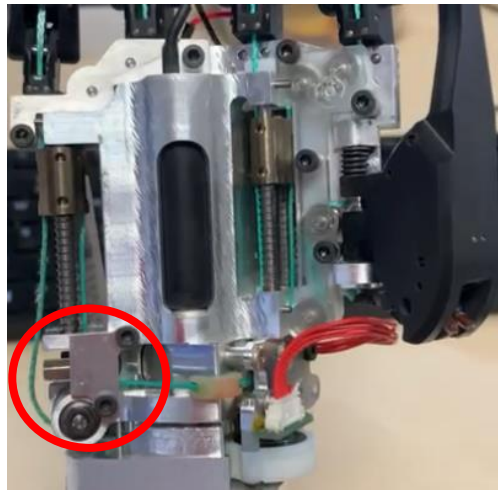


Figure 85 Loose wire

This photo represents an example of wire loosening (highlighted in red) during opening and closure of Hannes.

This means that when the system receives a closure command, the motor spends the first spins to re-wind the wire on the motor pulley. As a consequence, the control system isn't affected immediately by the inertial components characterizing the master-slave system thus the motor current rises slower.

This is not present in the multi-body model where the wire is ideal and inextensible. From this assumption it's easy to conclude why the motor current slope generated in simulation is higher.

In the Table 4 we reported the comparison between the motor current curves generated by Hannes with respect to our model during different grasping phases of a rigid object and a soft object. The currents associated to the rigid object show a comparable behaviour in terms of maximum amplitude, average void current during closure and average void current during opening (<10%). The error between the rising slopes of the two curves is also <10% meanwhile we see a larger error between the slopes during the re-opening of the hands (>10%). This is due to the difficulty of modelling an object of unknown characteristics and recreating the same experimental conditions in the Simscape environment.

Using the hand-dynamometer we were able to compare the total force exerted by Hannes fingers on the object with respect the total forces exerted by our model (see Figure 82). The results reported an error between the maximum forces exerted around 10%. This is due to the unmodeled losses, e.g. additional frictions, of the prosthesis that were not identified in the multi-body model. As a consequence, a higher percentage of the motor current is converted into mechanical torque in the model with respect to the actual device.

Regarding the currents generated during the grasping of a soft object, the error related to the average current during void opening and closure is lower than 10% and the same goes for the rising slope of the curve. The closure slope is around 10% probably due to the

reasons already cited. The main difference between the curves is noticeable for the maximum values of current, this is due to the difficulty of correctly estimating the real object stiffness.

In order to speed up the simulation time, we introduced some optimizations of the model.

In 3.2.1 we computed the RMSE of the trajectory computed by the fingers during closure between a model presenting the return wire mechanism and another model optimized with the use of torsional springs. The results report how the two implementations are equivalent (RMSE < 0.1 rad). This optimization can also be introduced in future versions of the Hannes hand, leading to a lighter and more compact solution.

In 3.2.2, instead, we compared the pulleys-wire system implementation with respect to external forces implementation. Also in this case, the RMSE of the trajectory computed by the fingers during closure between the two models are very low (RMSE < 0.07 rad). This secondary optimization cannot be used for mechanical optimization of the device but represents a useful tool to speed up simulation times.

Due to good results obtained in the validation, we decided to exploit the model to implement a new feature not currently present in Hannes: the stiffness recognition of grasped objects. Particularly, we trained a classifier, using data extracted from the interaction of the model with different objects in terms shapes, orientations, and stiffness (we pointed two values for the stiffness: one for the soft object and the other one for the rigid object).

As we can see from 3.3.1 the total classifier performance was very high (>85%), meaning that the overall interaction with the objects was correctly classified. Particularly, the classifier resulted to be more accurate in the classification of soft objects since in some of the rigid interactions, during the first contact phase, the objects were recognized as soft (See Figure 98). Moreover, it seems that the classification of narrow objects (see the metal block in Table 7) and of objects of small dimensions (see the plushies in Table 8) is more challenging for the classifier. This is probably related to the type of objects employed for the training. A possible solution could be introducing in the model the interaction with objects with greater variability to design and train a more performant classifier.

Nevertheless, these preliminary results demonstrate the possibility to recognise different object stiffness exploiting the motor current, reference position and measured position information. This demonstrates the possibility to maintain the Hannes hand a light and compact device while introducing new feature for a better human-machine interaction and utilization.

CHAPTER 5 Conclusion

In this study we demonstrated the great utility of creating a model for a complex system such as a prosthetic hand.

By exploiting a multi-body approach, we were able to reproduce the complete mechatronic structure of the poly-articulated prosthesis Hannes, composed by the mechanical system based on pulleys and cables, the DC motor, and the control system.

To have a realistic model we estimated the non-ideality affecting the system. This was a challenging step due to the impossibility to measure the actual frictional components from Hannes. The results reported how the multi-body model behavior is comparable to the average behavior of real device in terms of closure time and motor current generation.

Once created and validated, the model represents a powerful tool to easily investigate kinematic and dynamic variables of the real system such as joint angular excursion, joint torques and grasping forces exerted by the fingers. This is a great advantage with respect to traditional experimental set-ups which infer time consuming procedures and the use of expensive instrumentation.

Moreover, the model development allowed us to study and understand each aspect of Hannes leading to the proposal of a solution to optimize the mechanical components. In our study we investigated an alternative implementation for the return-wire mechanism, which consists in the introduction of two torsional springs per finger in place of the compression spring-wire system currently present.

The last step was the modelling of the interaction between the simulated hand and objects of different characteristics. This allowed us to identify the correlation between the stiffness of the chosen items and the motor current behaviour, leading to the proposal of a new feature not currently present in Hannes: the introduction of a variable haptic feedback according to the grasped object deformability. The idea was to extract valuable information from the model (motor current, RefPos, and output of the Encoder) and train a classifier to implement stiffness recognition.

The classifier showed very good results in distinguishing the interaction of the model with a rigid object from the interaction with a soft object.

Because of this, we decided to test the classifier on the real data acquired from Hannes. The results suggested that, even though the classifier was trained on data acquired from the model, it is capable overall to generalize on real data measured from the actual prosthesis Hannes.

However, its limitation was in the classification of objects of small dimensions, suggesting the need to create a wider dataset including items of variable sizes and stiffness or to train the classifier directly on data acquired from Hannes.

In conclusion, the multi-body approach demonstrated to be feasible in analysing and optimizing the prosthetic hand Hannes. Nevertheless, some aspects of the model may need to be improved such as the possibility to employ the simulation in online applications, not currently possible according to the chosen software configuration, the onerous simulation time, and a more accurate estimation of the frictional components present in Hannes.

Future applications of this work may focus on the implementation of torsional springs on the real prosthetic hand. This solution would lead to the elimination of the current bulky components, providing space inside the fingers for possible introduction of additional sensors.

Also, it would be interesting to implement an actual haptic feedback for amputees by exploiting the information of the classifier and testing if this solution can actually improve the embodiment of the prosthesis.

Bibliography

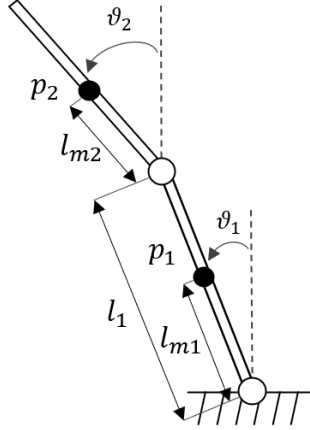
- [1] J. T. Belter, J. L. Segil, and B. S. Sm, "Mechanical design and performance specifications of anthropomorphic prosthetic hands: a review," *Journal of rehabilitation research and development*, vol. 50, no. 5, pp. 599-611, 2013.
- [2] R. Robert, "Model-based development of neuroprostheses for paraplegic patients," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 354, no. 1385, pp. 877-894, 1999.
- [3] M. A. Evans, "Model or prototype which, when and why?," 1992.
- [4] M. Laffranchi *et al.*, "The Hannes hand prosthesis replicates the key biological properties of the human hand," *Science Robotics*, vol. 5, no. 46, 2020.
- [5] P. Farlex, "Medical Dictionary," *Saunders comprehensive veterinary Dictionary*, 2012.
- [6] J. Rumbaugh, I. Jacobson, and G. Booch, "The unified modeling language," *Reference manual*, 1999.
- [7] J. Ludewig, "Models in software engineering—an introduction," *Software and Systems Modeling*, vol. 2, no. 1, pp. 5-14, 2003.
- [8] C.-H. Chien, C.-H. Chen, and T.-S. Jeng, "An interactive augmented reality system for learning anatomy structure," in *proceedings of the international multiconference of engineers and computer scientists*, 2010, vol. 1: International Association of Engineers Hong Kong, China, pp. 17-19.
- [9] V. Socha *et al.*, "Training of pilots using flight simulator and its impact on piloting precision," *Transport Means. Juodkrante: Kansas University of Technology*, pp. 374-379, 2016.
- [10] G. I. Zahalak, "An overview of muscle modeling," *Neural Prostheses: replacing motor function after disease or disability*, 1992.
- [11] R. Riener and T. Fuhr, "Patient-driven control of FES-supported standing up: a simulation study," *IEEE Transactions on rehabilitation engineering*, vol. 6, no. 2, pp. 113-124, 1998.
- [12] M. Muthugala, M. Munasinghe, M. Lakshan, L. Madurangi, and A. Jayasekara, "Design of an interactive robotic head with human-like movements," in *2013 IEEE 8th International Conference on Industrial and Information Systems*, 2013: IEEE, pp. 355-360.
- [13] D. De Lorenzo *et al.*, "Intraoperative forces and moments analysis on patient head clamp during awake brain surgery," *Medical & biological engineering & computing*, vol. 51, no. 3, pp. 331-341, 2013.
- [14] T. Varady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models—an introduction," *Computer-aided design*, vol. 29, no. 4, pp. 255-268, 1997.
- [15] H. Dallali *et al.*, "Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach," in *2013 IEEE International Conference on Mechatronics (ICM)*, 2013: IEEE, pp. 598-603.
- [16] P. Flores, *Concepts and formulations for spatial multibody dynamics*. Springer, 2015.
- [17] J. G. De Jalon and E. Bayo, *Kinematic and dynamic simulation of multibody systems: the real-time challenge*. Springer Science & Business Media, 2012.
- [18] R. Ryan, "ADAMS—Multibody system analysis software," *Multibody systems handbook*, pp. 361-402, 1990.
- [19] W. Rulka, "SIMPACT—A computer program for simulation of large-motion multibody systems," in *Multibody systems handbook*: Springer, 1990, pp. 265-284.
- [20] R. Smith and E. Haug, "DADS—Dynamic analysis and design system," in *Multibody systems handbook*: Springer, 1990, pp. 161-179.
- [21] P. Flores, J. Ambrósio, J. P. Claro, and H. M. Lankarani, "Multibody systems formulation," in *Kinematics and Dynamics of Multibody Systems with Imperfect Joints*: Springer, 2008, pp. 23-45.
- [22] S. E. Laboratory. "Multibody dynamics analysis basics." <https://www.sky-engin.jp/en/MBDynTutorial/chap02/chap02.html> (accessed).

- [23] Y. He and J. McPhee, "Application of optimisation algorithms and multibody dynamics to ground vehicle suspension design," *International Journal of Heavy Vehicle Systems*, vol. 14, no. 2, pp. 158-192, 2007.
- [24] K. O'riordain, P. Thomas, J. Phillips, and M. Gilchrist, "Reconstruction of real world head injury accidents resulting from falls using multibody dynamics," *Clinical Biomechanics*, vol. 18, no. 7, pp. 590-600, 2003.
- [25] A. M. Anamateros, "Modellazione multicorpo per la valutazione della severità delle lesioni da impatto," Politecnico di Torino, 2020.
- [26] K. Otsuka, Y. Wang, K. Fujita, H. Nagai, and K. Makihara, "Multifidelity modeling of deployable wings: Multibody dynamic simulation and wind tunnel experiment," *AIAA journal*, vol. 57, no. 10, pp. 4300-4311, 2019.
- [27] L. Ángel, M. Pérez, C. Díaz-Quintero, and C. Mendoza, "ADAMS/MATLAB Co-Simulation: dynamic systems analysis and control tool," in *Applied Mechanics and Materials*, 2012, vol. 232: Trans Tech Publ, pp. 527-531.
- [28] P. Fisette and J.-C. Samin, "Robotran: Symbolic generation of multi-body system dynamic equations," in *Advanced Multibody System Dynamics*: Springer, 1993, pp. 373-378.
- [29] G. Medrano-Cerda, H. Dallali, M. Brown, N. Tsagarakis, and D. Caldwell, "Modelling and simulation of the locomotion of humanoid robots," 2010.
- [30] M. Schlotter, "Multibody system simulation with simmechanics," *University of Canterbury*, pp. 1-23, 2003.
- [31] E. Neha, M. Suhaib, S. Asthana, and S. Mukherjee, "Grasp analysis of a four-fingered robotic hand based on matlab simmechanics," *Journal of Computational & Applied Research in Mechanical Engineering (JCARME)*, vol. 9, no. 2, pp. 169-182, 2020.
- [32] 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, 2007.
- [33] P. Geethanjali, "Myoelectric control of prosthetic hands: state-of-the-art review," *Medical Devices (Auckland, NZ)*, vol. 9, p. 247, 2016.
- [34] A. Marinelli, "Evaluation and Optimization of Prosthesis Control Based on Regression Machine Learning Algorithms," Università degli studi di Genova, 2018.
- [35] Ottobock. "Sensor Hand Speed." <https://shop.ottobock.us/Prosthetics/Upper-Limb-Prosthetics/Myo-Hands-and-Components/Myo-Terminal-Devices/SensorHand-Speed/p/8E39~58> (accessed).
- [36] O. Van Der Niet and C. K. van der Sluis, "Functionality of i-LIMB and i-LIMB Pulse hands: Case report," *Journal of rehabilitation research and development*, vol. 50, no. 8, p. 1123, 2013.
- [37] Ottobock, "Mano bebionic." [Online]. Available: <https://www.ottobock.it/protesizzazioni/prodotti-dalla-a-alla-z/mano-bebionic/>.
- [38] T. Laliberté, M. Baril, F. Guay, and C. Gosselin, "Towards the design of a prosthetic underactuated hand," *Mechanical Sciences*, vol. 1, no. 1, pp. 19-26, 2010.
- [39] B. Massa, S. Roccella, M. C. Carrozza, and P. Dario, "Design and development of an underactuated prosthetic hand," in *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, 2002, vol. 4: IEEE, pp. 3374-3379.
- [40] M. G. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the Pisa/IIT SoftHand," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 768-782, 2014.
- [41] S. J. Lederman and R. L. Klatzky, "Hand movements: A window into haptic object recognition," *Cognitive psychology*, vol. 19, no. 3, pp. 342-368, 1987.
- [42] J. W. Sensinger and S. Dosen, "A review of sensory feedback in upper-limb prostheses from the perspective of human motor control," *Frontiers in Neuroscience*, vol. 14, 2020.
- [43] A. Chatterjee, P. Chaubey, J. Martin, and N. Thakor, "Testing a prosthetic haptic feedback simulator with an interactive force matching task," *JPO: Journal of Prosthetics and Orthotics*, vol. 20, no. 2, pp. 27-34, 2008.
- [44] C. Pylatiuk, A. Kargov, and S. Schulz, "Design and evaluation of a low-cost force feedback system for myoelectric prosthetic hands," *JPO: Journal of Prosthetics and Orthotics*, vol. 18, no. 2, pp. 57-61, 2006.

- [45] P. E. Patterson and J. A. Katz, "Design and evaluation of a sensory feedback system that provides grasping pressure in a myoelectric hand," *J Rehabil Res Dev*, vol. 29, no. 1, pp. 1-8, 1992.
- [46] R. Clement, K. E. Bugler, and C. W. Oliver, "Bionic prosthetic hands: A review of present technology and future aspirations," *The surgeon*, vol. 9, no. 6, pp. 336-340, 2011.
- [47] T. Rosenbaum-Chou, W. Daly, R. Austin, P. Chaubey, and D. A. Boone, "Development and real world use of a vibratory haptic feedback system for upper-limb prosthetic users," *JPO: Journal of Prosthetics and Orthotics*, vol. 28, no. 4, pp. 136-144 %@ 1040-8800, 2016.
- [48] V. Gurevich, *Electric relays: Principles and applications*. CRC Press, 2018.
- [49] B. Joshi, R. Shrestha, and R. Chaudhar, "Modeling, Simulation and Implementation of Brushed DC Motor Speed Control Using Optical Incremental Encoder Feedback," in *Proceedings of IOE Graduate Conference*, 2014.
- [50] R. Condit, "Brushed DC Motor Fundamentals," *Microchip Application Note AN905*, Microchip Technology Inc, 2004.
- [51] C. Knospe, "PID control," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 30-31, 2006.
- [52] S. Wang, Z. Wu, D. Peng, S. Chen, Z. Zhang, and S. Liu, "Sensing mechanism of a rotary magnetic encoder based on time grating," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3677-3683, 2018.
- [53] S. Hao, Y. Liu, and M. Hao, "Study on a novel absolute magnetic encoder," in *2008 IEEE International Conference on Robotics and Biomimetics*, 2009: IEEE, pp. 1773-1776.
- [54] P. Campisi, "Progettazione e controllo di un esoscheletro industriale azionato con motori brushless," Politecnico di Torino, 2019.
- [55] MathWorks. "Choose a solver." <https://it.mathworks.com/help/simulink/ug/choose-a-solver.html> (accessed).
- [56] C. Hahn. "Solver Choice for Simulink and Simscape." <https://blogs.mathworks.com/racing-lounge/2017/12/08/solver-choice/> (accessed).
- [57] MathWorks. "Examine Model Dynamics Using Solver Profiler." <https://it.mathworks.com/help/simulink/ug/examine-solver-behavior-using-solver-profiler.html> (accessed).
- [58] L. Florio. "Creo mechanism to URDF." <https://github.com/icub-tech-iit/cad-mechanics-public/wiki/Creo-mechanism-to-URDF> (accessed).
- [59] B. Gavin. "What is an XML file." <https://www.howtogeek.com/357092/what-is-an-xml-file-and-how-do-i-open-one/> (accessed).
- [60] S. Traversaro. "simmechanics-to-urdf." <https://github.com/robotology/simmechanics-to-urdf> (accessed).
- [61] MathWorks. "Machine Learning in Matlab." <https://it.mathworks.com/help/stats/machine-learning-in-matlab.html> (accessed).
- [62] MathWorks, "Select Data and Validation for Classification Problem." [Online]. Available: <https://it.mathworks.com/help/stats/select-data-and-validation-for-classification-problem.html>.
- [63] A. D. Bellingegni *et al.*, "NLR, MLP, SVM, and LDA: a comparative analysis on EMG data from people with trans-radial amputation," *Journal of neuroengineering and rehabilitation*, vol. 14, no. 1, pp. 1-16 %@ 1743-0003, 2017.

Appendix I

The drop time for each finger was computed starting from the Euler-Lagrange equations of the inverted double pendulum model, whose complete calculations are reported below.



$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = u_i$$

$$L(\theta, \dot{\theta}) = T(\theta, \dot{\theta}) - U(\theta)$$

The center of mass and its derivative of the two links are defined as:

$$p_1 = \begin{bmatrix} l_{m1} \cdot \sin\theta_1 \\ l_{m1} \cdot \cos\theta_1 \end{bmatrix};$$

$$p_2 = \begin{bmatrix} l_1 \cdot \sin\theta_1 + l_{m2} \cdot \sin\theta_2 \\ l_1 \cdot \cos\theta_1 + l_{m2} \cos\theta_2 \end{bmatrix};$$

$$\dot{p}_1 = \begin{bmatrix} l_{m1} \cdot \dot{\theta}_1 \cdot \cos\theta_1 \\ -l_{m1} \cdot \dot{\theta}_1 \cdot \sin\theta_1 \end{bmatrix};$$

$$\dot{p}_2 = \begin{bmatrix} l_1 \cdot \dot{\theta}_1 \cdot \cos\theta_1 + l_{m2} \cdot \dot{\theta}_2 \cdot \cos\theta_2 \\ -l_1 \cdot \dot{\theta}_1 \cdot \sin\theta_1 - l_{m2} \cdot \dot{\theta}_2 \cdot \sin\theta_2 \end{bmatrix};$$

The kinetic and potential energies are defined as follow:

$$T_1 = \frac{1}{2} \cdot m_1 \cdot l_{m1}^2 \cdot \dot{\theta}_1^2 + \frac{1}{2} \cdot J_1 \cdot \dot{\theta}_1^2;$$

$$T_2 = \frac{1}{2} \cdot m_2 \cdot \left(l_1^2 \cdot \dot{\theta}_1^2 + l_{m2}^2 \cdot \dot{\theta}_2^2 + 2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \cdot \cos(\theta_1 - \theta_2) \right) + \frac{1}{2} \cdot J_2 \cdot \dot{\theta}_2^2;$$

$$U_1 = m_1 \cdot g \cdot l_{m1} \cdot \cos \theta_1;$$

$$U_2 = m_2 \cdot g \cdot (l_1 \cdot \cos \theta_1 + l_{m2} \cdot \cos \theta_2);$$

For the Lagrangian equation:

$$L = T_1 + T_2 - U_1 - U_2;$$

Starting from these equations:

$$\frac{\partial L}{\partial \dot{\theta}_1} = m_1 \cdot l_{m1}^2 \cdot \dot{\theta}_1 + J_1 \cdot \dot{\theta}_1 + m_2 \cdot l_1^2 \cdot \dot{\theta}_1 + m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_2 \cdot \cos(\theta_1 - \theta_2);$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) = \ddot{\theta}_1 \cdot (m_1 \cdot l_{m1}^2 + J_1 + m_2 \cdot l_1^2) + m_2 \cdot l_1 \cdot l_{m2} \cdot \ddot{\theta}_2 \cdot \cos(\theta_1 - \theta_2) - m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_2 \cdot \sin(\theta_1 - \theta_2) \cdot (\dot{\theta}_1 - \dot{\theta}_2);$$

$$\frac{\partial L}{\partial \theta_1} = -m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \cdot \sin(\theta_1 - \theta_2) + m_1 \cdot g \cdot l_{m1} \cdot \sin \theta_1 + m_2 \cdot g \cdot l_1 \cdot \sin \theta_1;$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = 0;$$

It's possible to derive the equation of motion of the first link:

$$\begin{aligned} \ddot{\theta}_1 \cdot (m_1 \cdot l_{m1}^2 + J_1 + m_2 \cdot l_1^2) + \ddot{\theta}_2 \cdot (m_2 \cdot l_1 \cdot l_{m2} \cdot \cos(\theta_1 - \theta_2)) + m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_2^2 \\ \cdot \sin(\theta_1 - \theta_2) - m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \cdot \sin(\theta_1 - \theta_2) - m_1 \cdot g \cdot l_{m1} \\ \cdot \sin \theta_1 - m_2 \cdot g \cdot l_1 \cdot \sin \theta_1 = 0; \end{aligned} \quad (1)$$

Then we computed the same calculations for the second link:

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 \cdot l_{m2}^2 \cdot \dot{\theta}_2 + m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \cos(\theta_1 - \theta_2) + J_2 \cdot \dot{\theta}_2;$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) = m_2 \cdot (l_{m2}^2 \cdot \ddot{\theta}_2 + l_1 \cdot l_{m2} \cdot \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \sin(\theta_1 - \theta_2) \cdot (\dot{\theta}_1 - \dot{\theta}_2)) \\ + J_2 \cdot \ddot{\theta}_2; \end{aligned}$$

$$\frac{\partial L}{\partial \theta_2} = m_2 \cdot l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \cdot \sin(\theta_1 - \theta_2) + m_2 \cdot g \cdot l_{m2} \cdot \sin \theta_2;$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = 0;$$

And we found the equation of motion of the second link:

$$\begin{aligned} \ddot{\theta}_2 \cdot (m_2 \cdot l_{m2}^2 + J_2) + m_2 \\ \cdot (l_1 \cdot l_{m2} \cdot \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - l_1 \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \sin(\theta_1 - \theta_2) \cdot (\dot{\theta}_1 - \dot{\theta}_2) - l_1 \\ \cdot l_{m2} \cdot \dot{\theta}_1 \cdot \dot{\theta}_2 \cdot \sin(\theta_1 - \theta_2) - g \cdot l_{m2} \cdot \sin \theta_2) = 0; \end{aligned} \quad (2)$$

The equations obtained are two 2nd order differential equations synthesized as:

$$\ddot{\theta}_1 = f(\ddot{\theta}_2, \dot{\theta}_1, \theta_2, \theta_1, \theta_2);$$

$$\ddot{\theta}_2 = f(\ddot{\theta}_1, \dot{\theta}_1, \theta_2, \theta_1, \theta_2);$$

In order to obtain four 1st order differential equations, we set up the following system:

$$\begin{cases} x_1 = \theta_1 \\ x_2 = \dot{\theta}_1 \\ x_3 = \theta_2 \\ x_4 = \dot{\theta}_2 \end{cases}$$

$$\begin{cases} \dot{x}_1 = x_2 = \dot{\theta}_1 \\ \dot{x}_2 = \ddot{\theta}_1 \\ \dot{x}_3 = x_4 = \dot{\theta}_2 \\ \dot{x}_4 = \ddot{\theta}_2 \end{cases}$$

This system was solved in MATLAB.

From these calculations it was possible to obtain the joint angular excursions reported in the graphs below. In particular the stream lines refer to the angular excursion resulting from the analytical calculations while the dashed lines represent the simulated angular excursions. Moreover, the red color stands for the phalanx link and the blue one for the middle link. The drop time was extracted from these graphs.

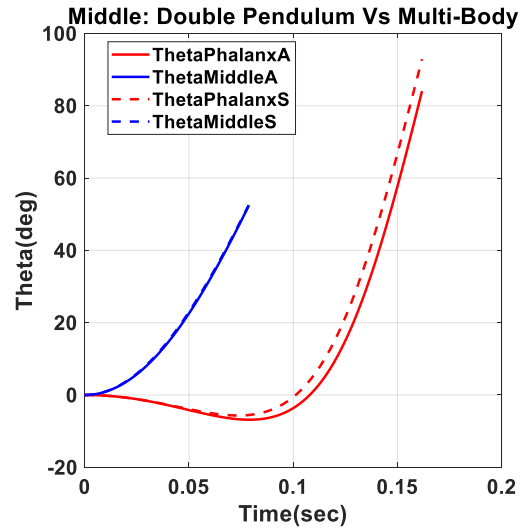
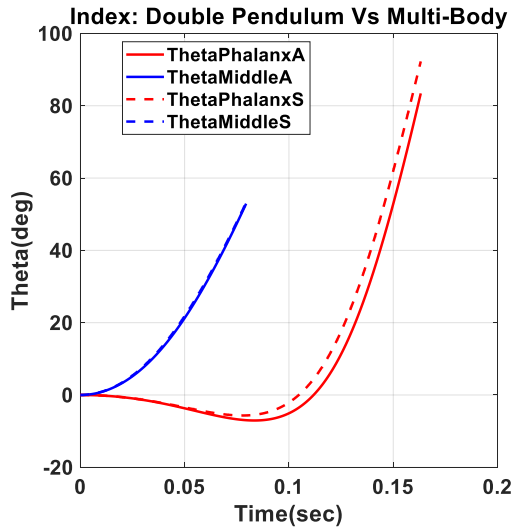


Figure 86 Index, middle Drop time

The stream line (in red for the phalanx and in blue for the middle parts) represents the analytical angular excursion of a double inverted pendulum model meanwhile the dashed line (in red for the phalanx and in blue for the middle parts) represents the angular excursion of each Multi-body model finger.

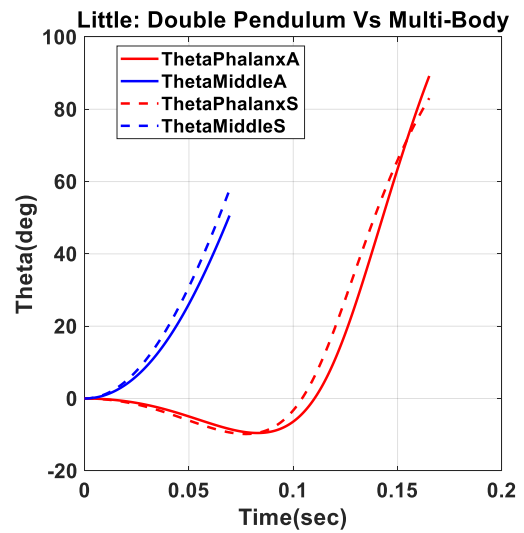
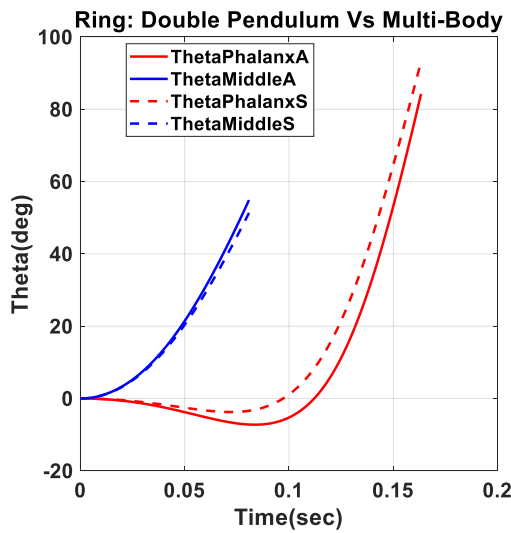


Figure 87 Ring, little Drop time

The stream line (in red for the phalanx and in blue for the middle parts) represents the analytical angular excursion of a double inverted pendulum model meanwhile the dashed line (in red for the phalanx and in blue for the middle parts) represents the angular excursion of each Multi-body model finger.

Appendix II

In this section we provided the comparison of the trajectory for each phalanx of each digit. The blue stream line represents the angular excursion of the proximal phalanx for the model presenting the return-wire implementation (RW), meanwhile the dashed blue line represents the angular excursion of the same phalanx but for the model presenting the torsional springs (TS) implementation.

The red stream line, instead, represents the angular excursion of the distal phalanx (Return-wire implementation) and the red dashed line the angular excursion of the same phalanx but for the model presenting the torsional spring implementation.

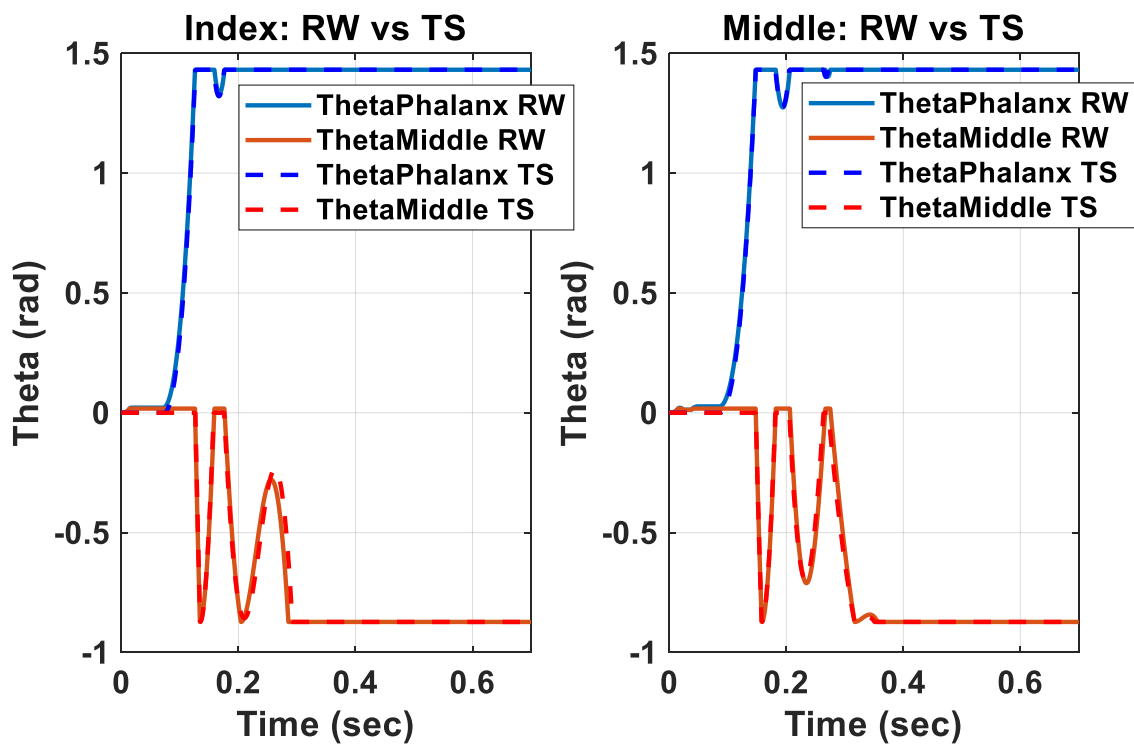


Figure 88 Index, middle RW vs TS

The stream line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the return wire (RW) implementation for each finger, meanwhile the dashed line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the torsional springs implementation (TS) for each finger.

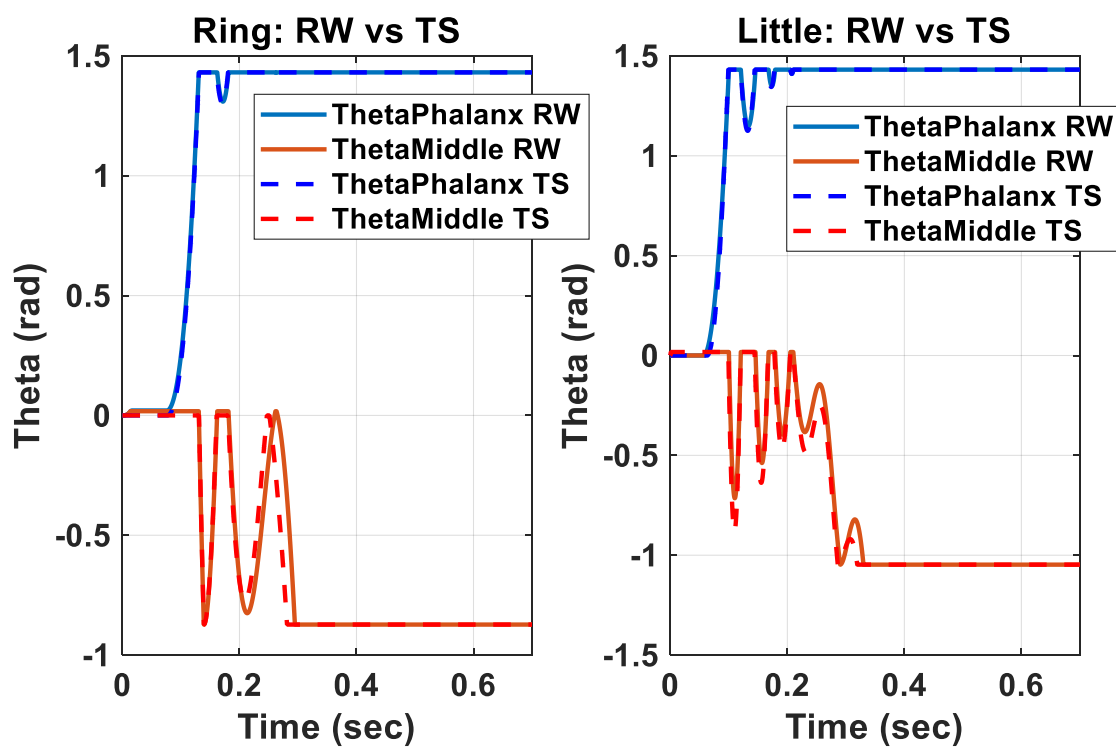


Figure 89 Ring, little RW vs TS

The stream line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the return wire (RW) implementation for each finger, meanwhile the dashed line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the torsional springs implementation (TS) for each finger.

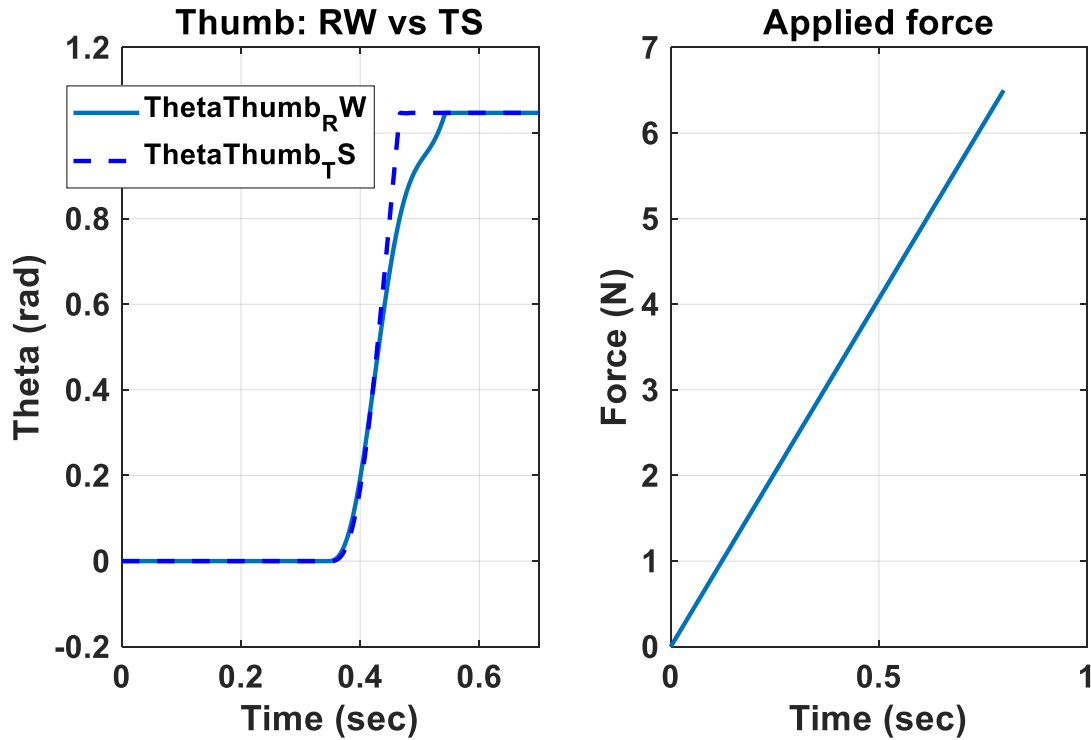


Figure 90 Thumb RW vs TS, applied force

(On the left) the stream line represents the angular excursion of the model presenting the return wire (RW) implementation for the thumb, meanwhile the dashed line represents the angular excursion of the model presenting the torsional spring implementation for the thumb. (On the right) The profile of the force applied to the fingers.

As we can see from the graphs, the angular excursion between the different phalanx of different fingers is comparable (please refer to 3.2.1 for the RMSE calculation for each curve). This suggests how the two implementations, return-wire and torsional springs, are equivalent in terms of kinematic of the model. However, thanks to the latter implementation we were able to lower the simulation time from 12 hours to 28 minutes.

Appendix III

In this section we provided the comparison of the trajectory for each phalanx of each digit. The blue dashed line represents the angular excursion of the proximal phalanx for the model presenting the pulley-wire system, meanwhile the stream blue line represents the angular excursion of the same phalanx but for the model presenting the external forces implementations.

The red dashed line, instead, represents the angular excursion of the distal phalanx (pulley-wire system) and the red stream line the angular excursion of the same phalanx but for the model presenting the external forces implementations.

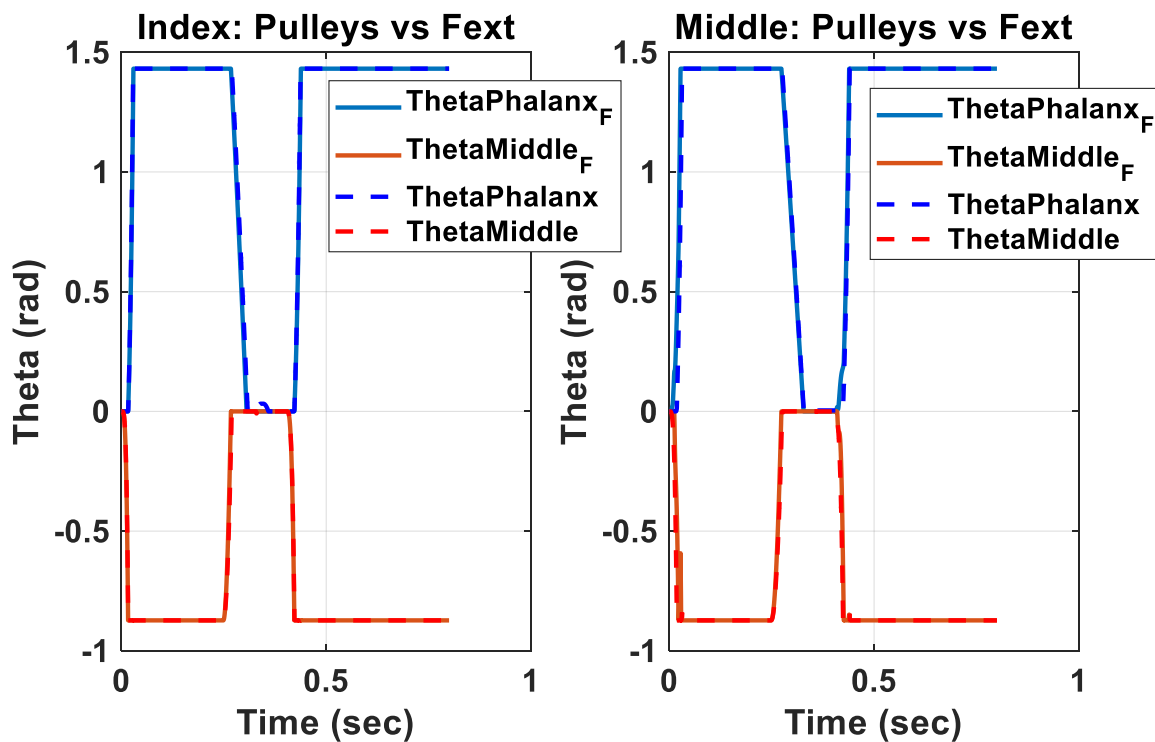


Figure 91 Index, middle pulleys vs Fext

The stream line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the external force implementation for each finger, meanwhile the dashed line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the pulley-cable implementation for each finger.

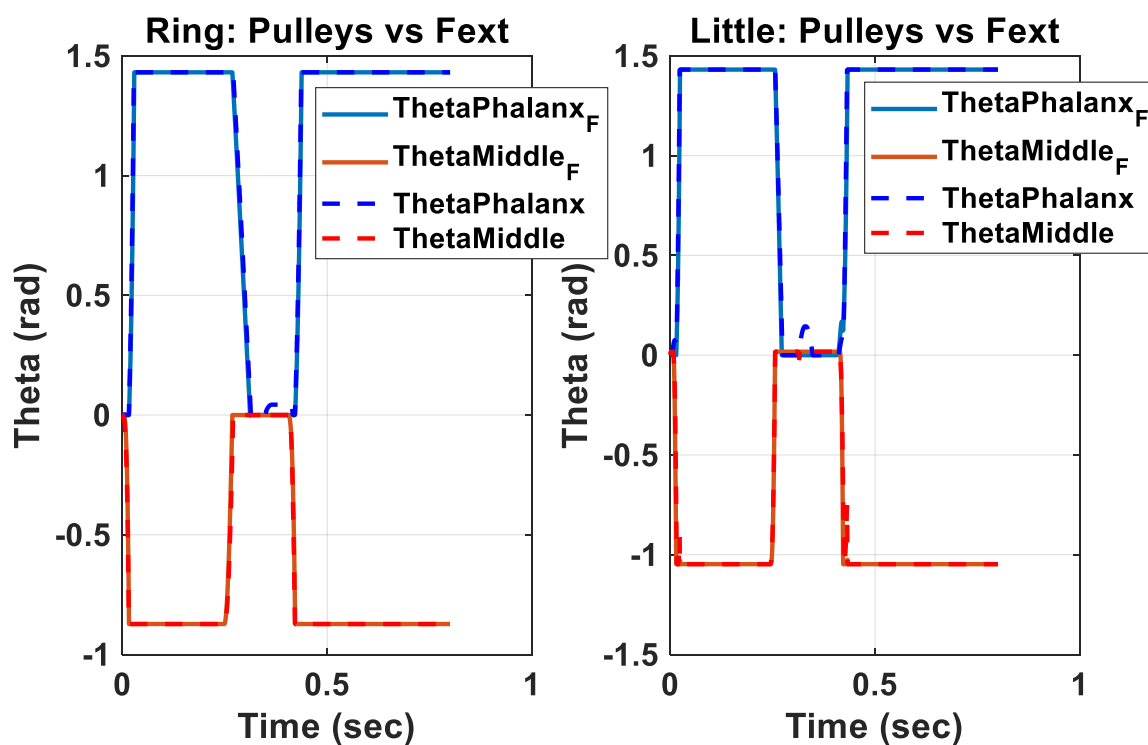


Figure 92 Ring, little pulleys vs Fext

The stream line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the external force implementation for each finger, meanwhile the dashed line (in blue for the phalanx and in red for the middle parts) represents the angular excursion of the model presenting the pulley-cable implementation for each finger.

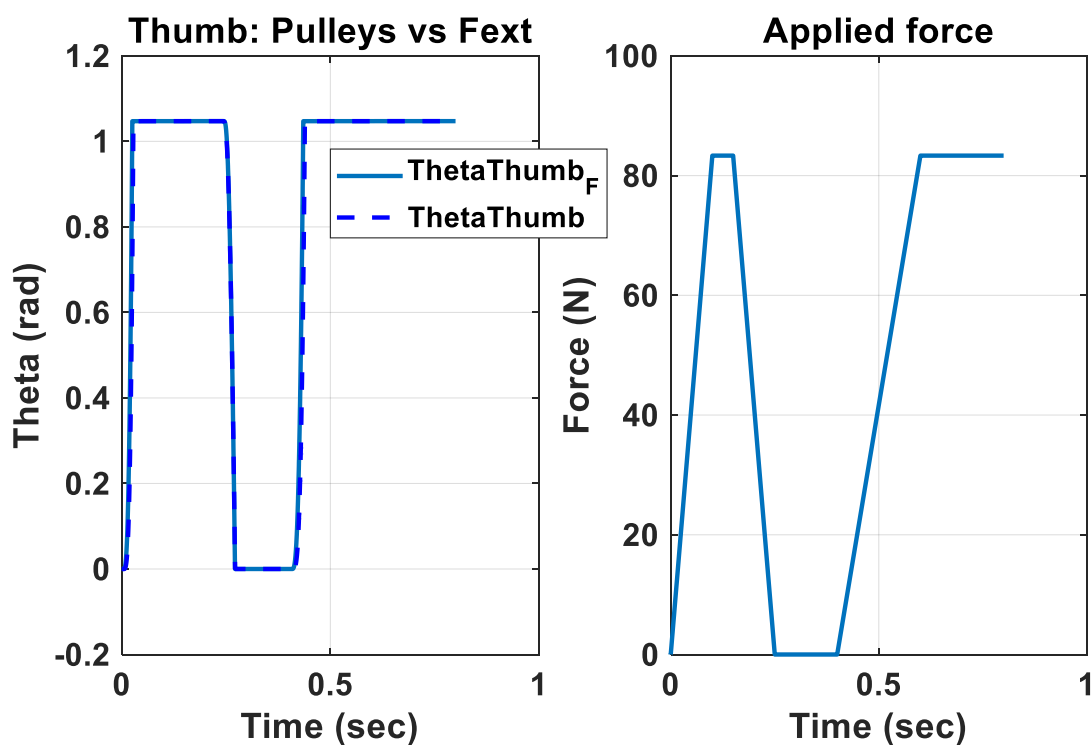


Figure 93 Thumb pulleys vs Fext, Force applied

(On the left) the stream line represents the angular excursion of the model presenting external force implementation for the thumb, meanwhile the dashed line represents the angular excursion of the model presenting the pulley-cable implementation for the thumb. (On the right) The profile of the force applied to the fingers.

As we can see from the graphs, the angular excursion between the different phalanx of different digits is comparable (please refer to 3.2.2 for the RMSE calculation for each curve). This suggests how the two implementations, pulley-wire system and external forces, are equivalent in terms of kinematic of the model. However, thanks to the latter implementation we were able to lower the simulation time from 28 minutes to 1 minute.

Appendix IV

This section presents the comparison between the motor current generated by the Multi-body model and the one generated by Hannes.

Particularly, we computed the relative error percentage between the two curves in terms of average current during the void closure and re-opening of the hand, the maximum current and the rising and falling slopes of the curves during the grasping phase (Please refer to 3.1.4 for the results)

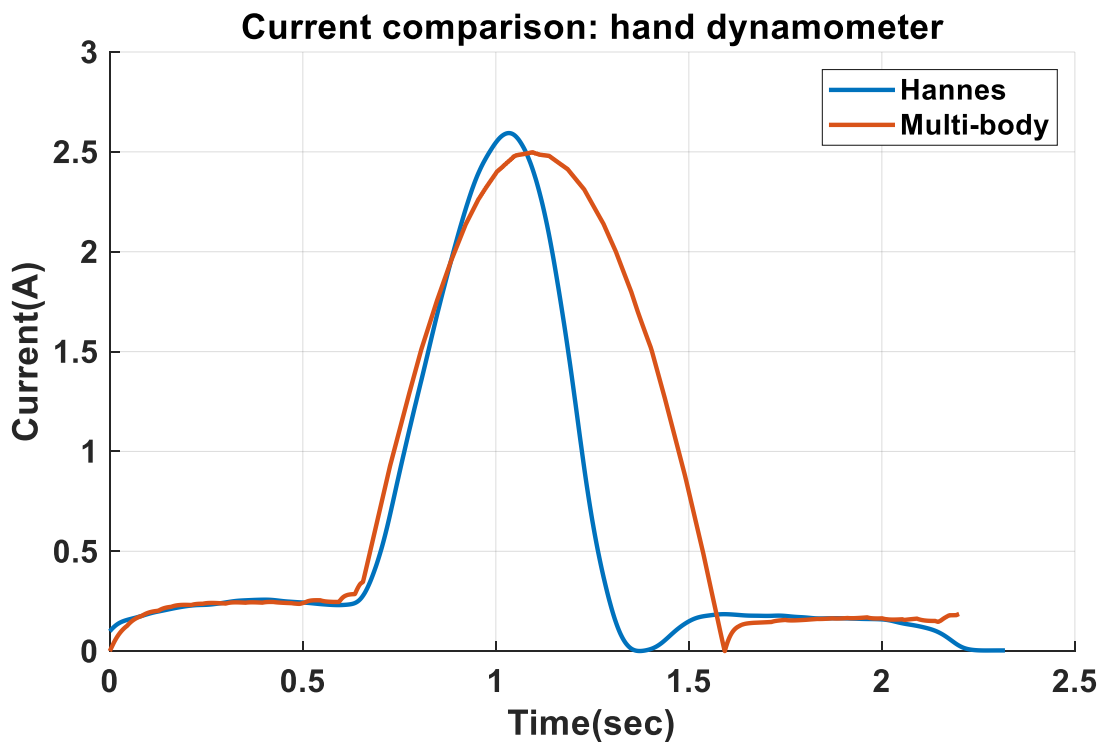


Figure 94 Current rigid object

In blue the motor current generated by Hannes and in red the motor current generated by the Multi-body model during rigid object interaction.

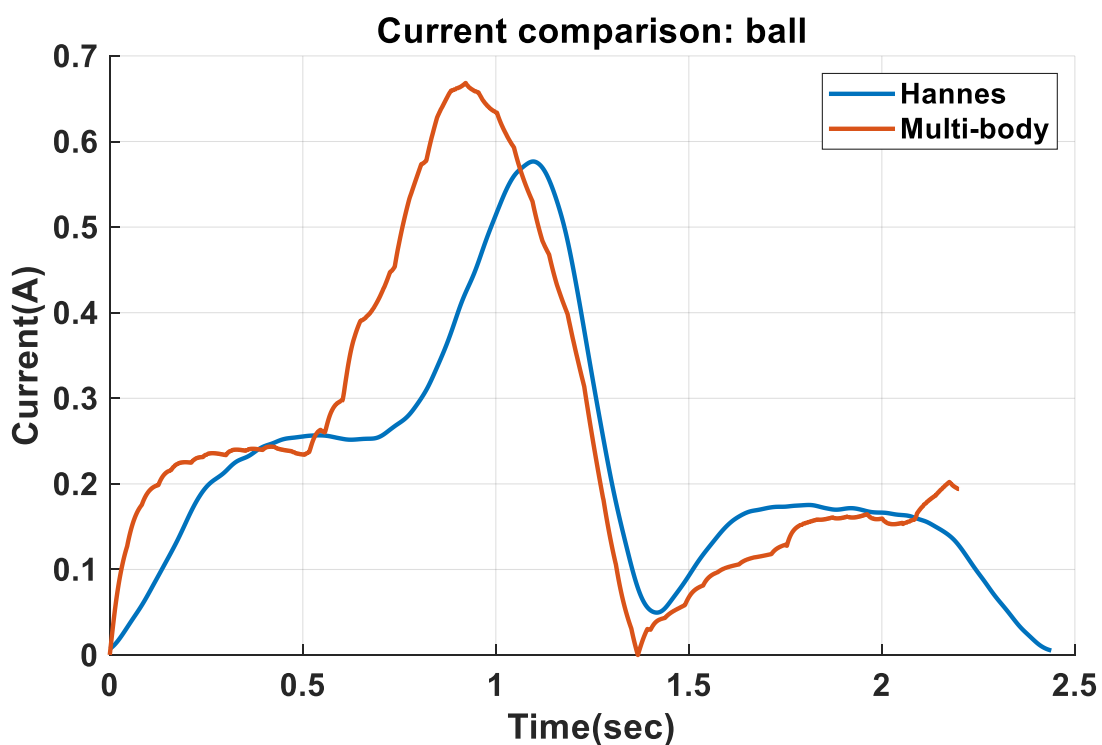


Figure 95 Current soft object

In blue the motor current generated by Hannes and in red the motor current generated by the Multi-body model during soft object interaction.

The Figure 95 reports the two motor current curves generated by the model (red line) and Hannes (blue line) during the interaction with a soft object. As we can see the transient phase during the void closure of the hand is different between the two. This is probably due to the frictional components as described in 2.1.3.7.

Appendix V

The following graphs presents the results of the objects classifications, whose F1Scores were reported in Table 7 and Table 8.

The graphs show the motor current trends as function of time during the different object interactions. For each current sample a class, indicating the type of interaction, is associated. In particular void is associated to class 0 (in green), rigid interaction to 1 (in red), soft interaction to 2 (in blue).

The ability of the classifier in identifying soft objects better than rigid ones is evident from the graphs. It is also noticeable the inaccuracy related to the classification of objects of small dimension.

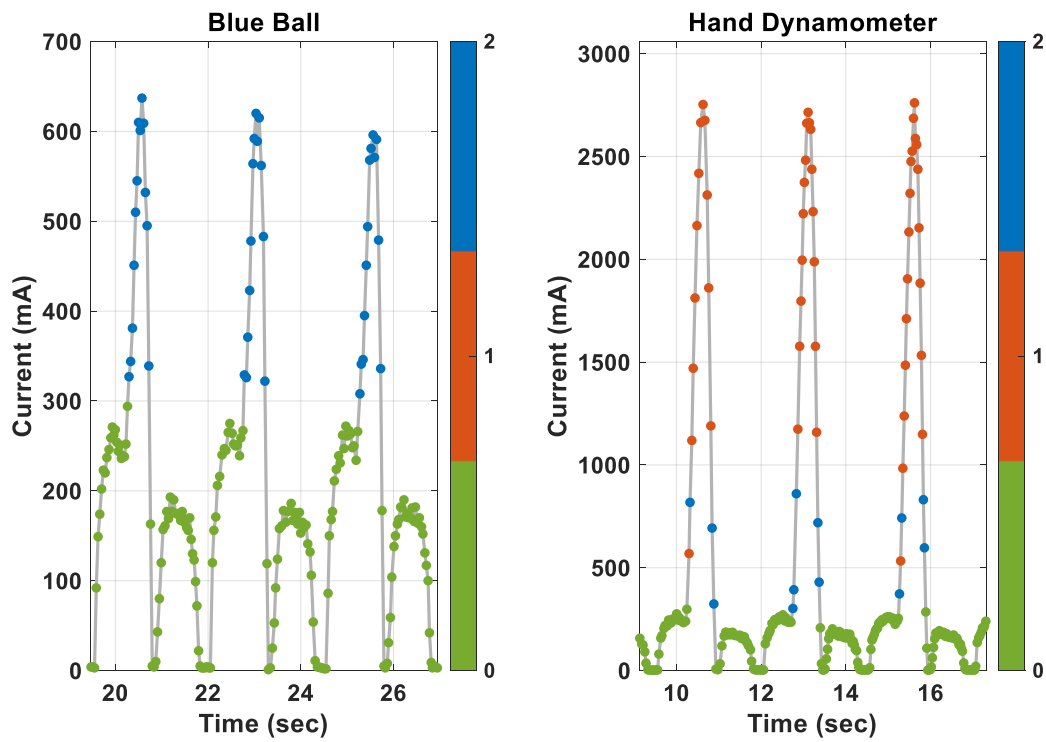


Figure 96 Classification (1)

These graphs represent the motor current generated by Hannes during different object interactions (soft on the left and rigid on the right). The blue points identify the soft object interaction, the red points the rigid object interaction, and the green points the void closure.

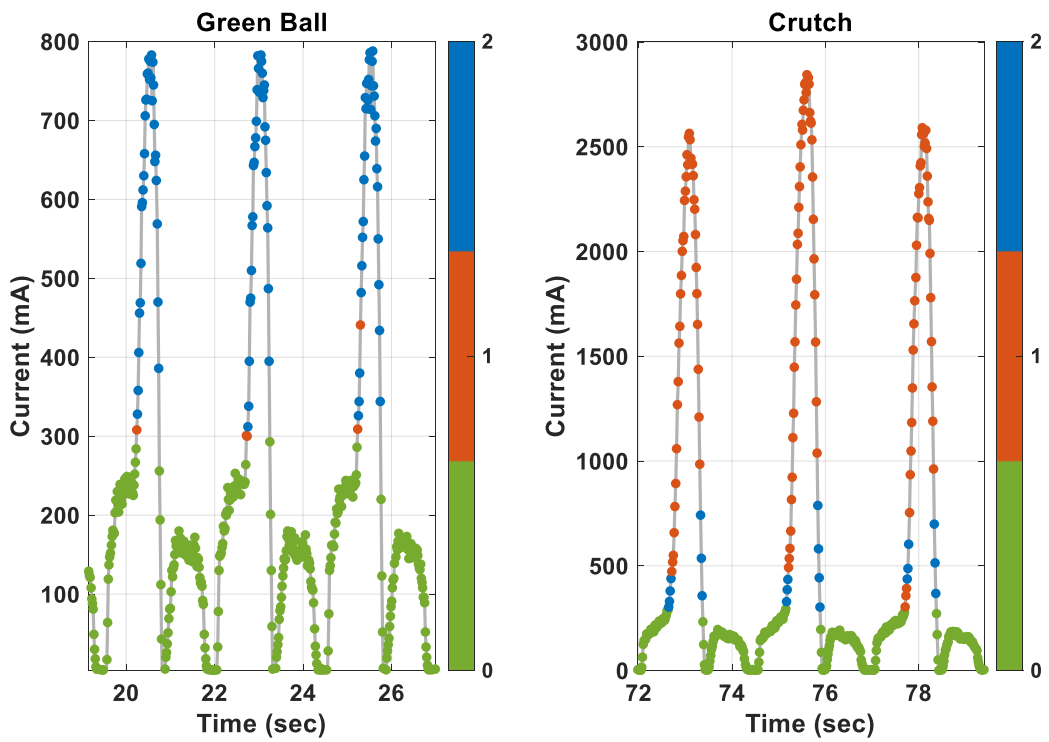


Figure 97 Classification (2)

These graphs represent the motor current generated by Hannes during different object interactions (soft on the left and rigid on the right). The blue points identify the soft object interaction, the red points the rigid object interaction, and the green points the void closure.

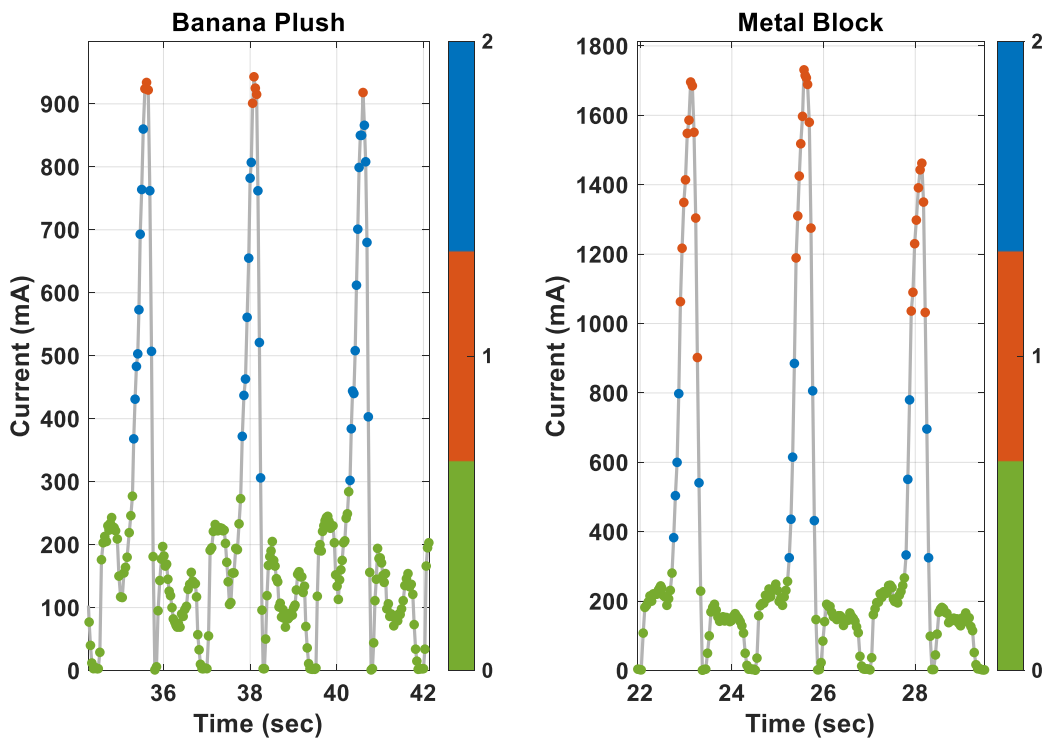


Figure 98 Classification (3)

These graphs represent the motor current generated by Hannes during different object interactions (soft on the left and rigid on the right). The blue points identify the soft object interaction, the red points the rigid object interaction, and the green points the void closure.

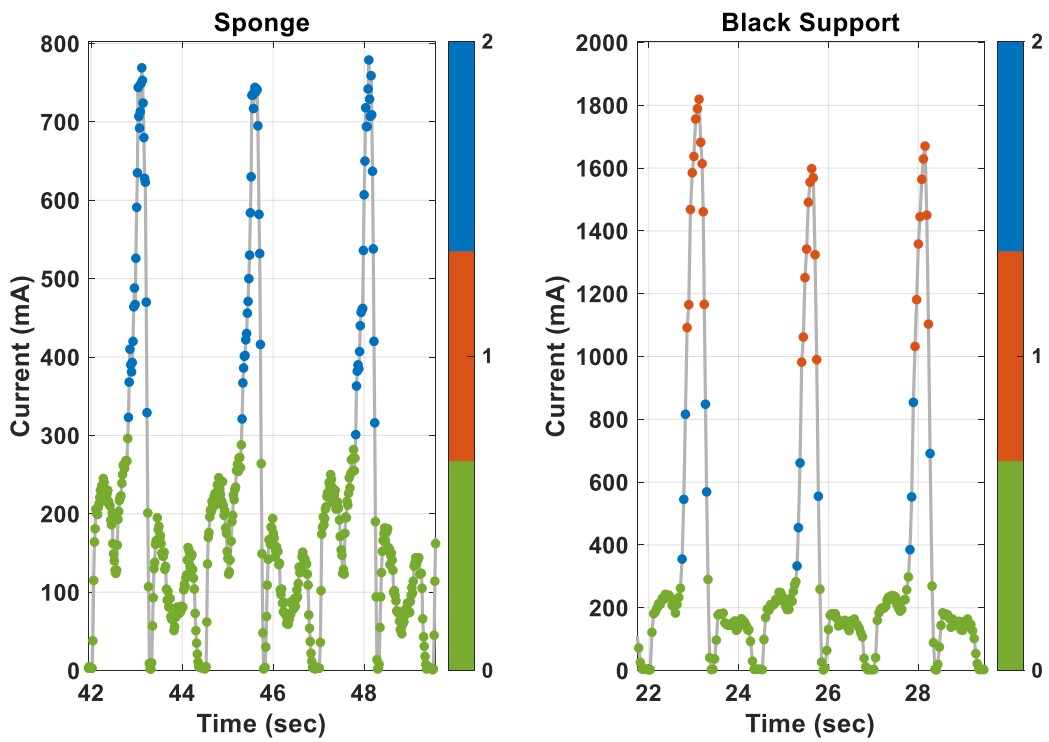


Figure 99 Classification (4)

These graphs represent the motor current generated by Hannes during different object interactions (soft on the left and rigid on the right). The blue points identify the soft object interaction, the red points the rigid object interaction, and the green points the void closure.

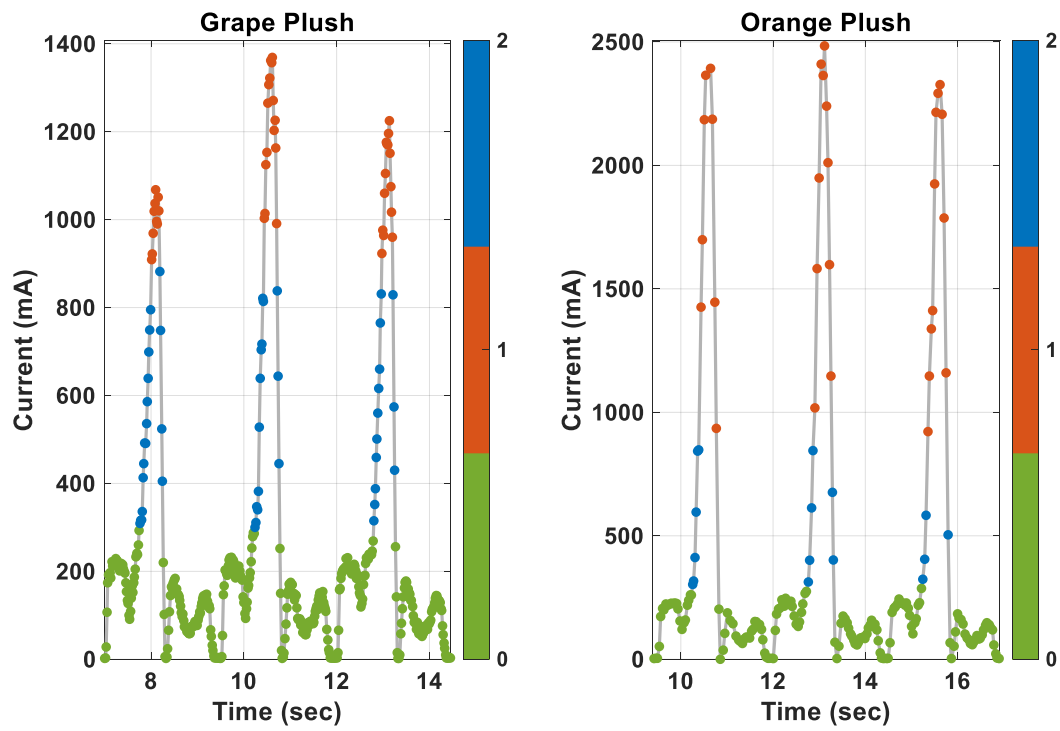


Figure 100 Classification (5)

These graphs represent the motor current generated by Hannes during two soft object interactions. The blue points identify the soft object interaction, the red points the rigid object interaction, and the green points the void closure.