

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Biomedica

Tesi di Laurea Magistrale
Simulation HMI with OpenSim



Relatori

prof. Laura Gastaldi

prof. Stefano Paolo Pastorelli

prof. Giovanni Gerardo Muscolo

Candidato

Matteo Musso

Anno Accademico 2019/2020

Table of contents

1	Introduction.....	8
2	Exoskeletons	10
2.1	Classification and types of the exoskeleton	11
2.2	Fields of use of exoskeletons	12
2.3	Limitations and problems.....	15
2.4	Simulation of exoskeleton.....	17
3	OpenSim	22
3.1	OpenSim Tools	23
3.1.1	Inverse Dynamics Tool	24
3.1.2	Computed Muscle Control (CMC) Tool	25
3.1.3	Forward Dynamics Tool.....	27
3.2	OpenSim API.....	29
3.3	OpenSim files	30
3.4	Available musculoskeletal models	32
3.5	Creation and importing of the exoskeleton geometry	37
4	Materials and methods	43
4.1	Elbow tests.....	43
4.2	Elbow tests results	45
4.3	Shoulder tests.....	48
4.4	Problems and errors in the MoBL model.....	49
4.5	Shoulder tests results.....	54

5 Exoskeleton development.....	57
5.1 Elbow tests and optimization of <i>BushingForce</i> parameters	60
5.2 Modelling of the exoskeleton shoulder.....	66
5.3 Test on the overall model	70
5.4 Analysis on the impact on muscle activation.....	72
5.4.1 Results	73
6 Comparison with real case and results	80
6.1 Real experiment	80
6.2 Setting up our simulations.....	81
6.3 Results	84
7 Conclusion	87
8 References.....	89
Acknowledgements.....	92

Table of figures

Figure 2.1 Hardiman, the first powered exoskeleton. Image taken from [1]	10
Figure 2.2 Raytheon XOS 2 Exoskeleton. Image taken from [3]	12
Figure 2.3 Exoskeleton in automotive industry. Image taken from [2]	13
Figure 2.4 The human shoulder. Image taken from [13].....	16
Figure 2.5 Exoskeleton used in the study [31].	20
Figure 3.1 OpenSim symbol.....	22
Figure 3.2 Schematic of the CMC algorithm applied to gait. Image taken from [34]	25
Figure 3.3 The three interface layers of OpenSim built on SimTK. Image taken from [37] ...	29
Figure 3.4 Structure of an OpenSim model file	30
Figure 3.5 Arm26 model	32
Figure 3.6 Wrist model.....	33
Figure 3.7 Stanford VA Upper Limb Model.....	34
Figure 3.8 MoBL-ARMS model	34
Figure 3.9 UBL_Projet model.....	37
Figure 3.10 MATLAB code wrote to insert the new body.	40
Figure 4.1 The linked model and the global reference system.....	43
Figure 4.2 Elbow torques necessary for a flexion from 0° to 140°.....	45
Figure 4.3 A) Elbow flexion angle for the two models. B) Elbow flexion speed for the two models.	46
Figure 4.4 Force generated by the two CLF of the elbow.....	47
Figure 4.5 The model with two links at the upper arm and at the forearm and the global reference system.....	48
Figure 4.6 DELT muscles length during the shoulder flexion, for the Schutte1993Muscle_Deprecated model.	49
Figure 4.7 DELT muscles length during the shoulder flexion, for the Thelen2003Muscle muscle model.....	50
Figure 4.8 Change in BIClong change of path.....	51
Figure 4.9 A) BIClong length during a shoulder flexion. B) Muscle during the simulation. ..	51
Figure 4.10 A) DELT1 length during a shoulder flexion with an elv_angle of 80°. B) Muscle during the simulation.....	52
Figure 4.11 Torque necessary for a shoulder flexion.....	53

Figure 4.12 Torque at the “shoulder_el” coordinates.....	54
Figure 4.13 Shoulder flexion angles.	55
Figure 5.1 Musculoskeletal model coupled with the exoskeletal one.....	57
Figure 5.2 Scheme of a bushing force. Image taken from [45].....	59
Figure 5.3 A) Elbow coordinates trend B) Speed.	60
Figure 5.4 Confront between the elbow torques required to complete an elbow flexion. for different models.....	61
Figure 5.5 Forces and torques generated by the BushingForce on the exoskeleton forearm body.....	63
Figure 5.6 Elbow coordinates trend for the model with the new parameters of the BushingForce.	64
Figure 5.7 Elbow torque for different models.....	65
Figure 5.8 Forces and torques generated by the BushingForce, with the new parameters, on the exoskeleton forearm body.	65
Figure 5.9 Shoulder coordinates trend.	67
Figure 5.10 Shoulder torques for the different models.	67
Figure 5.11 Forces and torques generated by the BushingForce, with the new parameters, on the exoskeleton upper arm body.....	68
Figure 5.12 Forces and torques generated by the BushingForce, with the initial value of the parameters, on the exoskeleton upper arm body.	69
Figure 5.13 Shoulder coordinates trend when torque was applied at the MoBL shoulder.	69
Figure 5.14 A) Shoulder coordinates trend during the movement B) Elbow trend.	71
Figure 5.15 Elbow coordinates trend for the simulation with the active exoskeleton.	73
Figure 5.16 A) Brachialis (BRA) force trends B) Brachioradialis (BRD).....	74
Figure 5.17 A) Biceps long head (BIClong) force trends. B) Short head (BICshort) force trends.	75
Figure 5.18 A) Joint reaction forces on X. B) On Y axis.....	77
Figure 5.19 Joint reaction forces on Z axis.	78
Figure 6.1 EksoVest Image taken from [47]	80
Figure 6.2 The model used for the simulation.	81
Figure 6.3 The .xml file for the external force.	82
Figure 6.4 The motion file of the external force.	83
Figure 6.5 Anterior deltoid (DEL1) force trend.....	84
Figure 6.6 Lateral deltoid (DEL2) force trend.....	85

1 Introduction

In the last decades, in many areas, exoskeletons have been developed to support man in carrying out different types of tasks.

Exoskeletons were created to help workers in fields such as construction or industry as automotive one. These devices were able to provide support to man during heavy and repetitive tasks burdensome for the human body and help to increase worker performances and productivity. Exoskeletons are also effective in increasing safety reducing accident on the workplace and chronic diseases.

Exoskeletons have also been developed in the medical or rehabilitation field. These devices treat or support patients in case of loss of a function. These technologies have been also investigated in the military and commercial fields.

In the last decades, with the introduction of more powerful computers easily available, simulation software has been developed. They are capable to reproduce how a device will behave before its construction. Thanks to them the design of this equipment has becoming easier, faster and less expensive.

Exoskeleton does not work alone; it operates in parallel with the man whom it provides support to. To evaluate the level of help that these devices can provide, the muscle force that the man must apply to perform a task with and without it can be take in consideration.

Musculoskeletal simulation software was created to study this parameter and, more generally, the human movement and the muscular forces that generate it.

One of this software is OpenSim created by the Stanford researchers, which today has reached its fourth version thanks to continuous development.

The purpose of this thesis is to use this software to interface a musculoskeletal model of upper limb, available in the OpenSim bibliography, with an exoskeletal model created by us.

After a short overview of the exoskeletons, the first part of this work focuses on the study of OpenSim software. The main features of the software were investigated.

Next step was to run some simulations of the chosen musculoskeletal model, to become familiar with the software. The kinematics and dynamics analysis were carried out on the free model and linked to weights. The scope of this analyses was to compare the results obtained with real data found in the bibliography and highlight problems.

Subsequently, simulations with the exoskeleton connected were performed. In these tests the connection method used between it and the musculoskeletal model was optimized, as it had a great influence on the kinematic and dynamic behavior obtained. Tests were performed initially without muscle activation and with a passive exoskeleton model and then simulating muscle activation during the performance of a task. These last tests were performed without the exoskeleton and with device able to provide support or not.

Finally, we compared the model obtained with a real case study taken from the bibliography. In the paper it was studied the support that different types of exoskeletons provide during the task performed. We then reproduced the movement performed by the subjects of the experiment and evaluated the muscular activation of our human model, with or without the exoskeleton, to compare our results with their.

2 Exoskeletons

Exoskeletons are cybernetic devices that work in parallel to the user, they are linked to the human body with belts, straps or in other way. They are used in different field to increase human capabilities and support movements. They have been able to improve human performance both in terms of strength and duration,

In 1890 Nicholas Yagin developed what is considered the first exoskeleton-like device. It was a system of compressed gas bags to assist the lower limbs. It stored and released energy to support walking, running and jumping.

The first powered exoskeleton was developed by General Electric and the US Armed Forces in 1960 and called Hardiman (figure 2.1). Thanks to hydraulic and electric actuators it was able to amplify the force generated by a man, but it was too big and heavy to have real applications. Exoskeleton research and development are greatly increased, given the numerous fields in which they can be used and the many benefits that they can provide



*Figure 2.1 Hardiman, the first powered exoskeleton.
Image taken from [1]*

2.1 Classification and types of the exoskeleton

According to exoskeletonreport.com [2] these devices can be classified in different way.

One of the most immediate ways is to classify them according to which part of human body they support or actuate. In commerce there are full body exoskeletons or devices relative only to the upper or lower extremities, that can act on one or more human joints.

If the device has actuators like electric, pneumatics or hydraulic motors it is defined powered or active. The class of powered exoskeleton can be divided in static and dynamic. In the first subclass the actuators always need to be turned on in order to maintain the shape of the device. For the second class this is not necessary, the actuators can be turned off without loss of shape, they are more energy efficient. The active exoskeleton need component to fuel the actuators, like batteries or connection cables to an external power source. Moreover, they need sensors and other devices to control the actuator, like encoders or inertial sensors. These components evaluate the position of the user and calculate the level of force and torque needed for the support of the movement.

The passive exoskeletons haven't electrical power source, the force is generated by systems of spring and dampers that initially store energy to supply it later. This leads to a lower complexity and size compared to the active type. The passive devices can be used in different way:

- to redistribute the weight of an object, so that it does not act on the person;
- to store energy in order to support subsequent phases of the movement;
- to act like a damper to adsorb shock or reduce vibration.

Another type of passive exoskeletons is used to maintain static position, they are unobtrusive and can be locked in a specific position.

Other two classes in which the exoskeletons can be divided are the pseudo-passive and hybrid. The first class have batteries that power electronics that are not used to provide actuation, the second use FES (functional electrical stimulation) of the muscles as actuators [2], each have controllers and sensors that govern the movement.

The exoskeleton can be fixed to a support that can be itself fix or movable. The device can be mobile if it can move freely with the operator.

Another possible classification that can be made is related to the materials that constitute them. Initially, rigid materials like metals or carbon fiber were used, now researches are focused on

flexible materials for all the parts (soft exoskeleton or exosuit) to reduce the weight of the device and to make them more comfortable and manageable, not to limit the user's movements.

2.2 Fields of use of exoskeletons

The development of exoskeletons is taking place mainly in four areas: medical rehabilitation, for work and industry, military field and in the commercial field, like hiking assistance, sports and camera gimbal support.

In the medical field the studies and the research go in two directions: rehabilitation and mobility (augmentative device). In the first area the exoskeletons are used to make patients perform a repetitive movement, the exoskeletons help the performance of the task with a level of support that decreases gradually. The patients being able to perform more exercise repetitions, and in a better and more complete way, with the support of the device. As a patient recover the initial functionality, the amount of assistance decreases. When it ideally it reaches zero the device it's no longer needed. The mobility exoskeletons are used when the patient can't get back to the initial condition or its residual capabilities are not enough to complete the normal activities. They are used to support the patient permanently, since it will never recover completely.

Military exoskeletons are being tested by a lot of state, and obviously a lot of projects remain secret. They are thought to support the soldier, therefore they have to be comfortable to wear, very resistant and integrable with the armament. There is a problem with the active exoskeletons due to their batteries that should be very resistant and do not risk exploding, even if hit.



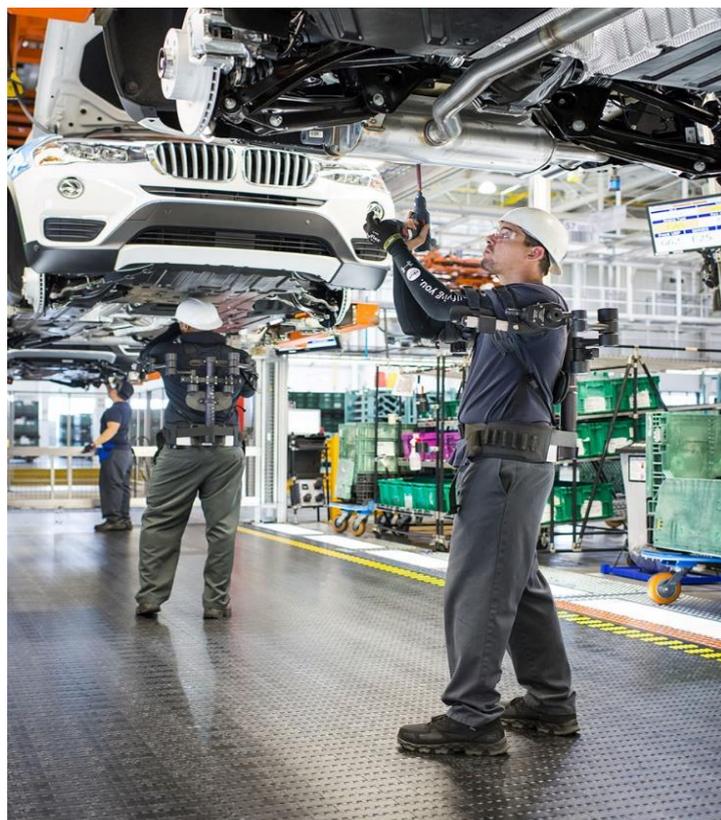
*Figure 2.2 Raytheon XOS 2 Exoskeleton.
Image taken from [3]*

In the industrial field the exoskeletons are thought to support the worker during the working activity.

A lot of research has been done especially the automotive industry and exoskeleton use has considerably increased. In this type of industries, the human contribution is still essential and hardly replaceable with a robot. The manual operations needed are difficult to be robotized. A considerable effort has been made to try to adapt the vehicle design to a completely mechanized production type. As reported by Voilqu'e et al. in [4], human operators are still better than robots and can overcome the limitations of these devices thanks to their intelligence, decision making capacity, dexterity (small and flexible parts) and ability to climb into the car or to move rapidly around the assembly [4].

These repetitive activities performed routinely lead to physical problems for the workers especially in the upper limb (at the shoulder) and at the spine. Exoskeletons are studied and used to reduce these musculoskeletal disorders (MSDs) and to improve human performances.

Voilqu'e et al. in their work [4] saw that in this field both passive and active exoskeletons are used; the market is basically divided equally between them.



*Figure 2.3 Exoskeleton in automotive industry.
Image taken from [2]*

Nowadays, passive exoskeletons are preferred by industrial companies because their mechanical actuation is reliable, easy to maintain, lighter, un-tethered and easier to use [4]. The active exoskeletons thanks to the sensors and controllers of the actuators are able to generate different profile of forces and torques compared to the spring and damper component of the passive exoskeletons. However, this is paid in terms of size, weight, complexity and obviously price. The active devices used, furthermore, usually support only a part of the user body and transfer the supported weight and force to another part of the body or directly to the ground. For these reasons industrial companies prefer the passive devices.

Exoskeletons should theoretically reduce the occurrence of MSDs, increase productivity, improve the quality of work and increase worker stamina. As reported in reference [5] these devices have some unexpected consequences as increasing muscle activity in other part of the body, specifically the legs, raising of the energy expenditure (metabolic cost), reducing the range of motion and therefore hindering the movement, enhancing the discomfort due to localized contact pressure.

Because of these aspects not always you can get an improvement in performance. The results are obviously influenced by the type of exoskeleton used and the task performed.

An example can be seen in the paper [6] that will be used to evaluate our exoskeleton model. Alabdulkarim et al. tested 3 type of upper limb devices. As can be seen in their paper [6] all the three exoskeletons tested increased the number of errors committed during the drilling task, especially when a greater precision is required. The two less anthropomorphic exoskeletons increased the overall average muscle activation compared to the free case.

The type of movement influences the performance too. In the paper [7] Kim et al. test an exoskeleton when the drill work is done overhead or at the shoulder height. This device acts directly on the shoulder of the participants. For the first case of study the number of errors respect the free condition significantly increases. In the second case, when the task was performed at the shoulder height, there is an improvement of the performance.

From this work we can understand how an improvement in working conditions is not so obvious. It is difficult to achieve a real decrease in total muscle activity required and obtain an increase in the worker's performance. Researches should focus on creating exoskeletons more specific to the type of movement performed.

2.3 Limitations and problems

Exoskeleton are still in an initial phase of development and implementation in the fields above listed. There are some problems to solve and aspects to improve, as reported in the reference [8]. The principal aspects to improve are six.

Power supply, especially for the exoskeletons to be used "in the field", i.e. outside a context in which this device can be plugged to a power source [8]. The batteries currently used have a limited duration and may explode.

Materials used to make the exoskeletons are in process of improvement, to obtain devices lighter and more manageable. Researcher are trying to reduce the weight of actuators but maintaining their power. To do so, elastic actuator and McKibben air muscle have been tried.

Another aspect that under development is the power control and modulation of the actuator. Algorithms used by the controller are in improvement phase to make the exoskeleton follow well the user during the movement and support him consistently during the task that must be performed.

Exoskeletons should also be adaptable to the user size; this problem can be partially solved with the use of soft devices that have a greater adaptability.

The last thing, it is the difficulty to reproduce exactly the human joint, especially the complex ones, like the shoulder and the hips. This difference results in a mismatch between the motion of the human limb and the exoskeleton.

This issue has effect also on this work, the lost in the alignment of the center of rotation of the shoulders of the two body generates forces that influence a lot the results.

In the paper [9] is reported that the total misalignment is the sum of four separate effects: kinematic mismatch, migration of the instantaneous center of rotation, initial offset and movement mismatch.

Like reported by A. Schiele in the reference [10] the misalignment between the bodies can generate force of interaction with a value up to 200 N and torque of about 1.5 Nm. These components are not without effects. They can cause discomfort or pain and, if these devices are used for an extended period, long-term injury like dislocation [10,11,12]. They can also reduce the range of motion (ROM) or even prevent movement, if the incompatibility is too high. These forces influence the activation of the muscles.

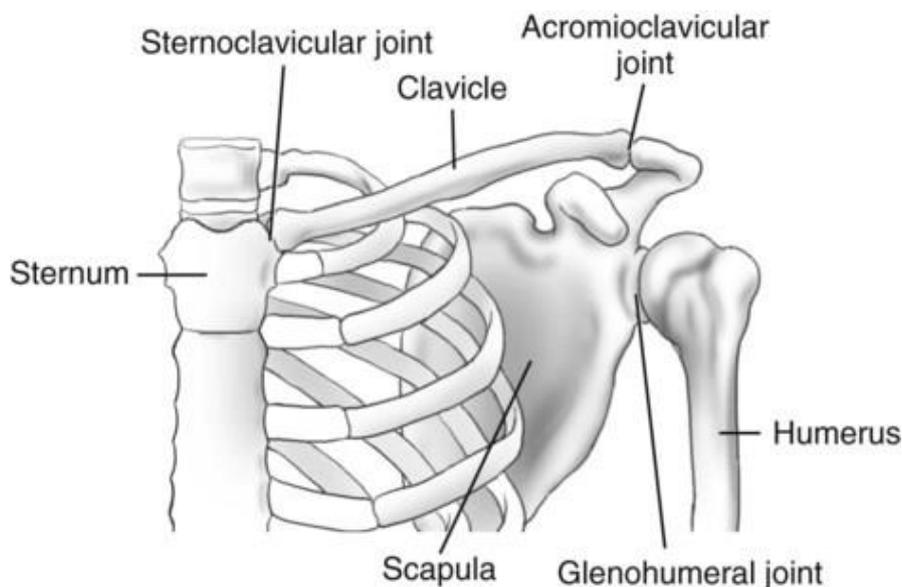
The initial offset is due to the to the difficulty of evaluating the correct rotation axes of the joint due to the presence of soft tissue above it. In [10] is reported that even if there is a perfect initial alignment, during the movement it is possible to reach differences of even 10 cm. This is also caused by a sliding of soft tissues.

The mismatch between the motion of the man and the exoskeleton is mainly due to the difference between the number of degrees of freedom (DOF) of the human joints and their counterpart in the exoskeleton. In fact, usually only the main DOF that allow wider movements are taken into consideration, and this leads to the problems already mentioned

One of the human joints most affected by this phenomenon is the shoulder due to its complexity. It has not only three rotational DOF, but also two translational movement, even if small:

- In the horizontal plane from the front to the back, in a protraction and retraction movement;
- In the vertical direction called elevation and depression.

Human shoulder is the complex of four joints, the glenohumeral (GH) joint and three articulations that form the shoulder girdle: the sternoclavicular (SC), the acromioclavicular (AC) and the scapulothoracic (ST) joint. This last one, like the subacromial joint, is a physiological joint, it does not foresee relations between two skeletal segments.



*Figure 2.4 The human shoulder.
Image taken from [13]*

Shoulder girdle moves the position of the centre of glenohumeral (GH) joint, giving the two translational DOFs at the shoulder. Its kinematics is also tightly dependent to the movements of the humerus, this coupling is known as scapulohumeral (SH) rhythm [14,15].

This joint its often modelled as a spherical joint, only the 3 rotation are taken into account without taking in consideration the two translational DOFs. This discrepancy in the shoulder ROM creates the problems listed above. To reduce this problem, especially in the rehabilitation field, exoskeletons that can follow the shoulder translational movement have been created, like ASSISTON-SE [15].

2.4 Simulation of exoskeleton

Simulation software have revolutionized the world of engineering and industry. Thanks to them it is now possible to simulate the behaviour of a mechanical object with a high level of precision, allowing to save time and money by no longer having to build and try the components. Exoskeletons can also be simulated before being created, this allow to evaluate how well they can follow and support the human movements. One software used for these scopes is Siemens NX Human [16]—based on Tecnomatix Jack technology. Another software used to simulate the kinematics and dynamics of the exoskeleton is SimMechanic [17] of MathWorks.

In order to evaluate an exoskeleton, the most important aspect to underline is its interaction with the human body and in particularly on the level of support that it is able to provide.

This could be studied thanks to musculoskeletal simulation software like OpenSim [18,19] and AnyBody Modelling System [20]. With them, it is possible to simulate and evaluate the resulting muscular activation, in order to study the effectiveness of the exoskeleton in supporting the human task.

Although OpenSim was used for the thesis, two works performed with the AnyBody software will be reported below, they are both relative on exoskeletons for upper limbs. The first and oldest, wrote by Agarwal et al., can be seen in the paper [21]. They simulate an elbow flexion with a downward force of 60 N applied to the hand, in different condition. They made a first simulation with the model free, a second with a constant assistive moment of 16 Nm at the elbow, a third with a variable assistive moment and at the end a simulation with an exoskeleton coupled to the human arm model, that generate a variable torque at its elbow coordinate. The exoskeleton used was a simplified model, it has two links connected with a revolute joint [21] and the upper link was also connected with a revolute joint to the ground.

This job shows the difficulty in modelling the connection between the exoskeleton and the human model. To connect the forearms of the two part of the model they didn't use a standard joint, because the kinematic incompatibility between the two elbows. Whit standard joint hyperstaticity and therefore uncontrolled interaction forces would appear. To connect the two forearms a "kinematic measure" was used. It is an equation that represent a kinematical constraint, like, for example, maintaining distance between two points. In the second work, carried out with Anybody, developed by Zhou et al. in the paper [22], two models of exoskeleton different only in the direction of the first joint were tested. They are more realistic models than one presented in the previous work. The interaction between the exoskeleton and the human arm is modelled as a spherical joint. In both simulations the exoskeletons were tested during the movement of an arm picking up a cup.

In OpenSim the works in the bibliography focus on exoskeletons of the lower limbs, this is probably due to the fact that this program is mostly used for gait analysis, and for the presence of a greater number of musculoskeletal models related to the lower limbs than to the upper ones. We are going to present them below to highlight the aspect of interconnection between the two parts, human and mechanical, of the model. In fact, more connection methods have been used, without finding one that would fit well for all the cases.

One of the first works was made by the researchers Ferrati et al. of the university of Padua, Italy, that can be seen in the paper [23]. In their work they simulate the behavior of an existing lower-limb exoskeleton. They constrained it to the musculoskeletal model at the height of the human pelvis, and then the coordinates of the two models have been paired with a *CoordinateCouplerConstraint*. They made four simulation with ComputeMuscleControl Tool (CMC) of OpenSim to evaluate the forces of the exoskeleton actuators necessary to complete the imposed kinematics. The simulations can be divided into two groups: with the ankle joint of the exoskeleton locked or not. For each group it was simulated the configuration with the exoskeleton free or coupled with the man.

In the paper [24] the interaction between the device and the human during a squat exercise in microgravity conditions was simulated. They used a *WeldConstraint* between the shoulder contact points and two *PointConstraint* between the human foots and the foot plates of the device. A CMC analysis was performed, to evaluate the muscle and the joint reaction forces, by using the experimental data: kinematics and the contact force between the subject and the exoskeleton at the height of the foots and the shoulders. They have encountered problems in the way of connection the two parts of the model. The weld constraint gave problems in the

kinematics behavior of the model due to the great rigidity, while the point constraints don't had enough rotational rigidity, causing problem in the transmission of forces.

The paper [25] is the work of D.C. Pujals where the *BushingForce* were used to connect the human and the exoskeleton model. Since this connection method will also be used in this thesis, the work will be presented in detail later.

In the work [26] the authors evaluate the behavior of a passive exoskeleton, determining the engaging and disengaging timing and optimizing the stiffness of the springs of the device for walking energetic efficiency. They do not specify how the exoskeleton has been connected to the human model.

In the paper [27] the authors develop a method, call EIKPE (Extended Inverse Kinematics Posture Estimation), that allows predicting the motion of the human joints from the knowledge of the angular motion of the exoskeleton frame. They import the experimental kinematics data in OpenSim and use it to evaluate the human joint coordinate values. The exoskeleton was coupled with the human model with three different types of constraint: a *WeldConstraint* between the human pelvis and the exoskeleton corset, a *PointOnLineConstraint* between the tibia and its fixation and a *PointConstraint* between the human foot and the exoskeleton sole.

Other articles are relative to the ankle-foot orthosis (AFO). In the internet OpenSim example page [28] and in the paper [29] can be seen two of them. In both studies one body of the AFO is connected with the foot with a *WeldJoint*. The same way of connection has been used between the tibia and the AFO of the work in OpenSim example page, while in the paper [29] this connection is modelled with a *CustomJoint* that allowed a translation motion along the tibia's longitudinal axis. Both models had *BushingForce* but used in two different way: in the first case these forces connect the AFO bodies while in the second they connect the AFO body to the ankle joint center. There was a torsional spring that models the bending stiffness PD-AFO.

On the upper limb some work simulates the movement of the human arm with OpenSim, and then use this kinematics data to evaluate an exoskeleton with other programs. A work done completely in OpenSim was relative to the fingers, paper [30].

Finally, in the paper [31] it has been simulated an upper limb exoskeleton. The author added a model of the device at the Arm26 model of OpenSim. The model of the exoskeleton is simple, two identical body and one rotational joint that connects them. The upper part of it is attached to the shoulder of the Arm26 model in a rigid way, the lower is constrained to the wrist of the musculoskeletal model.

There is an actuator at the point of connection between the two bodies, the torque that it generates is based on the constraint force between the two model, the resultant torque cancels 75% of this force: the exoskeleton becomes nearly transparent.

The purpose of the work is to improve the existing user-exoskeleton models by introducing a simulation architecture that can reproduce its dynamic interaction, thereby altering the initial motion of the user. A simulation architecture is developed that uses the musculoskeletal models from OpenSim and implements an exoskeleton control algorithm and human response model in MATLAB.

In the first part of the study there is the research of the parameter in the dynamic response of a human to a perturbing force that has a close resemblance to human reflexes.

To do this they made 5 participants perform an elbow flexion, at a constant speed that was then disturbed by a torque applied at the elbow of a real exoskeleton. In this part they used the CMC tool of OpenSim to calculate the torques of the unperturbed and perturbed motion, with the musculoskeletal model free.

After that they asked at the participants to catch an object of 0.9 kg from a height of 0.3 m, while looking in the opposite direction with the exoskeleton attached. They did two simulation in OpenSim for this test, one with the weight catch bay the human hand and the other by the exoskeleton.



Figure 2.5 Exoskeleton used in the study [31].

3 OpenSim



Figure 3.1 OpenSim symbol

OpenSim is an open source and freely available software package. It enables you to build, exchange, and analyze computer models of the musculoskeletal system and dynamic simulations of movement. OpenSim version 1.0 was introduced at the American Society of Biomechanics Conference in 2007, and with version 2.0, an application programming interface (API) has been added, allowing researchers to access and customize OpenSim core functionality [32].

The last version is the 4.0 released on November 13, 2018; which is the version used for the studies in this thesis.

OpenSim has a graphical user interface (GUI) where we can use and modify the model. Furthermore, we can access to all the tools that OpenSim makes us available to carry out our analyzes and see the result of our simulations

OpenSim is based on the Simbody library. This is an open-source multibody dynamics engine developed to create mathematical models of biological dynamics.

3.1 OpenSim Tools

There are many tools available that can be used to carry out different analyses on our model. Below, all of them will be listed, but we will present more in detail only those used in this thesis.

- **Plot Tool:** allow us to plot the results of our analysis, but also to export them; the result of the simulations can be open also with other program, in our work the results were open and plotted with MATLAB.
- **Scaling Tool:** it allows to scale your model in order to match the anthropometrical dimensions of the subject of the study. The tool modifies the dimensions and the inertial parameters of the bone or of other bodies, the length of the muscles and tendons.
- **Inverse Kinematics (IK) Tool:** this tool allows us to obtain the articular positions (angles) in each time step, starting from the experimental data relative to the position of the marker during the movement.
- **Inverse Dynamics Tool:** the ID tool computes the generalized forces acting at each joint of the model during the motion. In input it requires the kinematics data of the joint (articular angle, angular speeds and accelerations) and the external force data, the force and torque acting on the subject.
- **Static Optimization Tool:** it is an extension of the ID, it calculates the muscle forces that cause the movement, the inputs that this tool want are the same needed by the ID tool.
- **Residual Reduction Algorithm (RRA) Tool:** it tries to minimize the effects of modeling and marker data processing errors that aggregate and lead to large nonphysical compensatory forces called residuals. [33]
- **Computed Muscle Control (CMC) Tool:** it allows us to compute the muscle excitation pattern that accomplish to the kinematics and dynamics data given as input. It is also able to manage other types of actuators applied to the model and calculate their control value, or their influence on the muscle activation.
- **Forward Dynamics Tool:** it calculates the kinematics, or other type of data depending on the added analysis, that result from the muscle excitation pattern calculating by the CMC tool, for example. It is also able to simulate the behavior of the model when

functions or actuators, driven by a controller, are put in the coordinate of the OpenSim model, thanks to the API.

- **Analyze Tool:** as the name suggest, it allows to analyze the results of the other tool in various ways. It can't record the kinematic of all the model or of a single point of our interest. It can analyze the behavior of a muscle or an actuator of the model. In our work we will use it to calculate the joint reaction loads with the results of the CMC Tool given as input.

In our thesis having no experimental data the Scaling Tool, the Inverse Kinematics Tool, and the RRA tool have not been used. The Scaling Tool wasn't used since the model was left in its original dimensions and the exoskeleton adapted to it. The Static Optimization Tool has also not been used since the musculoskeletal model utilized for our work present constraint between the coordinate of the model. With these components this tool fails to calculate the right forces generated by the muscle. So, the CMC tool has been used instead of the aforementioned.

Below we will better present the tools of OpenSim used.

3.1.1 Inverse Dynamics Tool

The ID Tool perform an inverse dynamic analysis to calculate the generalized forces (forces and torques) that act on the coordinates and so on the joints of the model, during the imposed motion. The ID Tool asks to input obviously the model to be analyzed, the file containing the kinematic history of its coordinates and the external loads acting on it, like the ground reaction force.

The ID Tool solve iteratively the classical equation of motion that can be written in this form:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau$$

With:

- q, \dot{q}, \ddot{q} vectors of the generalized positions, velocity and acceleration respectively;
- $M(q)$ system mass matrix;
- $C(q, \dot{q})$ vector of the Coriolis and of the centrifugal forces;
- $G(q)$ vector of the gravitational forces;
- τ vector of the generalized forces.

The left part of the equations is completely known from the input files, so the ID Tool solves this equation to find the generalized forces acting on the model.

3.1.2 Computed Muscle Control (CMC) Tool

It allows to calculate the control value, the excitation, of the muscles or of other type of actuators, that permit to reproduce the kinematic behavior of the model given as input.

In addition to the file with the kinematics of the articulations of the model, the tool asks to input the tracking tasks file, which specifies which coordinates to follow and with which level of precision. This is done with a system of weights, greater is this one and greater is the precision with which the coordinate is followed.

Other files that can be given as input are those relating to the residual and reserve actuators. The residual actuators acting between the model and the ground are used to control the global position and orientation of the model. They can provide both positive and negative forces. The reserve type is used to make up to an insufficient muscle force and do not let the simulation fail. They were applied to the coordinates of the model. To disincentivize their use from OpenSim a high control value should be used with a low level of maximal force that they can provide.

Another input file that can be imported was the control constraints one. It contains the limits of the excitation for the all the type actuators of the model. It can be also used to define the control profile of an actuator of the model. In this way we can get the actuator to provide a force with a profile predetermined.

The CMC tool calculate the muscle excitation level by using a combination of proportional-derivative (PD) control and static optimization.

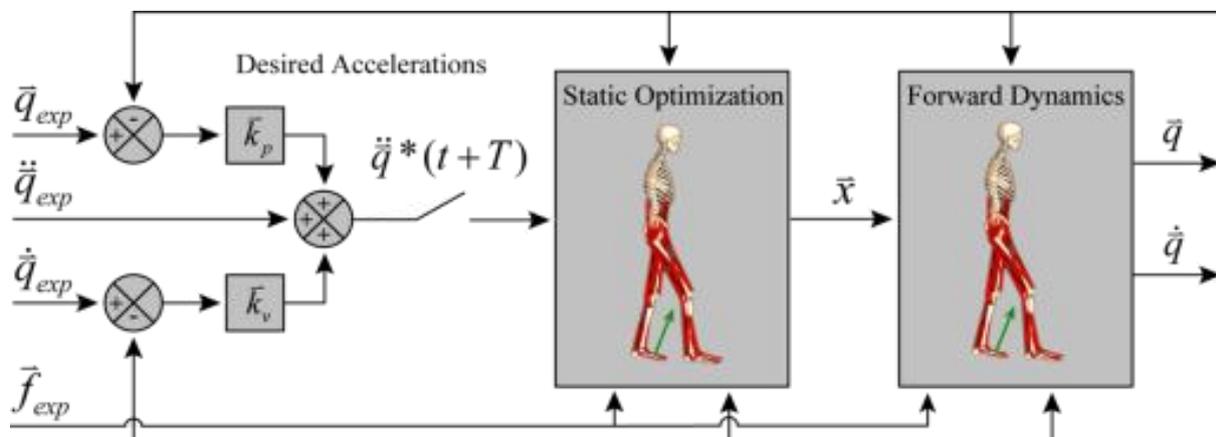


Figure 3.2 Schematic of the CMC algorithm applied to gait.
Image taken from [34]

The PD control law used was the following:

$$\ddot{\vec{q}}^*(t + T) = \ddot{\vec{q}}_{exp}(t + T) + \vec{k}_v[\dot{\vec{q}}_{exp}(t) - \dot{\vec{q}}(t)] + \vec{k}_p[\vec{q}_{exp}(t) - \vec{q}(t)]$$

It was the first step of the CMC algorithm where the tool calculates the desired acceleration $\ddot{\vec{q}}^*$. $\vec{q}(t)$ are the model coordinate that must reach the experimentally coordinate \vec{q}_{exp} . This is a function where the error between the model and experimentally coordinate was driven to 0. \vec{k}_v and \vec{k}_p are the feedback gains on the velocity and position errors, respectively. If their value is chosen with the following relation:

$$\vec{k}_v = 2\sqrt{\vec{k}_p}$$

The error will be driven to zero in a critically damped way. Good values for these two parameters are 20 for \vec{k}_v and 100 for \vec{k}_p .

T is the time interval where the desired accelerations are computed. It is chosen to be about 0.01 s, it is small but big enough to allow muscle forces to change, as they cannot do it instantly.

The second stage of the CMC is a static optimization, the actuator controls \vec{x} are computed. These values are calculated in order to achieve the desired accelerations calculated in the first step.

In OpenSim the static optimization problem can be formulated in two different ways:

1. Slow target formulation: this formulation consists of a performance criterion J that is a weighted sum of squared actuator controls, this summation distributes and minimizes the loads across the actuator of the model, plus the sum of desired acceleration errors, that drives the model accelerations toward the desired accelerations. [35]

$$J = \sum_{i=1}^{n_x} x_i^2 + \sum_{j=1}^{n_q} w_j (\ddot{q}_j^* - \ddot{q}_j)^2$$

2. Fast target formulation: it is the sum of the squared controls augmented by a set of equality constraints $C_j = 0$.

$$J = \sum_{i=1}^{n_x} x_i^2 \text{ with } C_j = \ddot{q}_j^* - \ddot{q}_j \quad \forall j$$

The second formulation turns out to be faster and with a higher capability of tracking but if its condition can't be achieved, the simulation will fail. To prevent the simulation to fail due to insufficient muscle strength, which is very common cause, as mentioned before, we can add the reserve actuators to the coordinates of the model.

The final stage of the CMC algorithm is a standard forward dynamics simulation. The actuators control values calculated in the second step are used to conduct a FD, by a time interval "T". These three steps are performed in succession by OpenSim, until the end of the movement interval is reached.

To start the CMC simulation the software, take 0.03 s to compute the initial state of the model. Since the initial kinematics data are known from the input during this interval OpenSim compute the initial muscle states (muscle activation levels and fiber lengths) that are generally unknown. In this interval the muscle forces can change dramatically since the model isn't in equilibrium, the results are not correct. So, it is a good thing start the simulation 0.03 second before the interval of our interest.

3.1.3 Forward Dynamics Tool

This tool performs a forward dynamics simulation. It solves (integrates) the differential equations that define the dynamics behaviour of a musculoskeletal model.

It's the opposite of the inverse dynamics, here in input there is the forces that drive the motion that we want to know.

The tool solves the Newton's second law, for a multibody system, to find the accelerations of the coordinates of the body:

$$\ddot{q} = [M(q)]^{-1}\{\tau + C(q, \dot{q}) + G(q) + F\}$$

With:

- q, \dot{q}, \ddot{q} vectors of the generalized positions, velocity and acceleration respectively;
- $[M(q)]^{-1}$ inverse of the mass matrix;
- τ vector of the joint torques;
- $C(q, \dot{q})$ vector of the Coriolis and the centrifugal forces;
- $G(q)$ vector of the gravitational forces;
- F vector of other forces applied to the model.

The muscular forces acting on the system can be studied with the following mathematical functions:

$$\tau_m = [R(q)] f(a, l, \dot{l})$$

With:

- τ_m moments due to muscle forces;
- $R(q)$ moments arms;
- f muscles forces, which are function of:
 1. a , muscle activations;
 2. l , muscle lengths;
 3. \dot{l} , speed of contraction of muscle fibers. This parameter depends on the muscle contraction dynamics (Λ) which has as variables the length and the activation level of the muscle (l and a) and the position and speed of the articulation (q and \dot{q}) on which the muscle act.

$$\dot{l} = \Lambda(a, l, q, \dot{q})$$

The last differential equation that rule the musculoskeletal dynamics was:

$$\dot{a} = A(a, x)$$

Where the activation rate of the muscles (\dot{a}) depend on the activation dynamics of the muscles (A), with “x” the model controls (neural excitations).

The output of the tool is then the kinematics behavior of the model due to input files. These files can be the one containing the time histories of the model controls, found with the CMC tool for example, or the files containing additional actuators or external loads acting on the model.

It is possible adding actuator directly to the OpenSim model thanks to the API. It is important to say that an actuator, also the muscles, without a controller or a file that contain the time histories of the control value don't provide any type of force during an FD simulation. The resulting force provided by the actuator is the result between the control value supplied to it and the optimal force parameter.

So, the FD tool allow us to obtain the kinematics file with the behavior of the model when a function or an actuator is imposed to one or more of its elements, whether they are coordinates or bodies, thanks to the API.

Finally, at the FD analysis can be added other types of analysis selectable from the specific panel of the tool or can be carried out later thanks to the Analyze Tool.

3.2 OpenSim API

An application programming interface (API) is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between them and facilitates their interaction. [36]

Using the API it is possible to make OpenSim communicate with other programs like C++ or Python programming software and with MATLAB making them access to the source code of OpenSim.

This software was written with the C++ language and so with this programming language new types of components can be created to extend its capabilities for example, plugins can be created. They are used to extend the OpenSim functionality like new analyses or model components. With MATLAB and Python only the existing components can be use and the plugins can only be launched. Thanks to the API it is also possible to launch the simulations and tools of OpenSim directly from the MATLAB environment. Theoretically you can launch and carry out your entire simulation without open OpenSim. There is also a basic graphical interface that can be opened to see the progress of the simulation started in MATLAB.

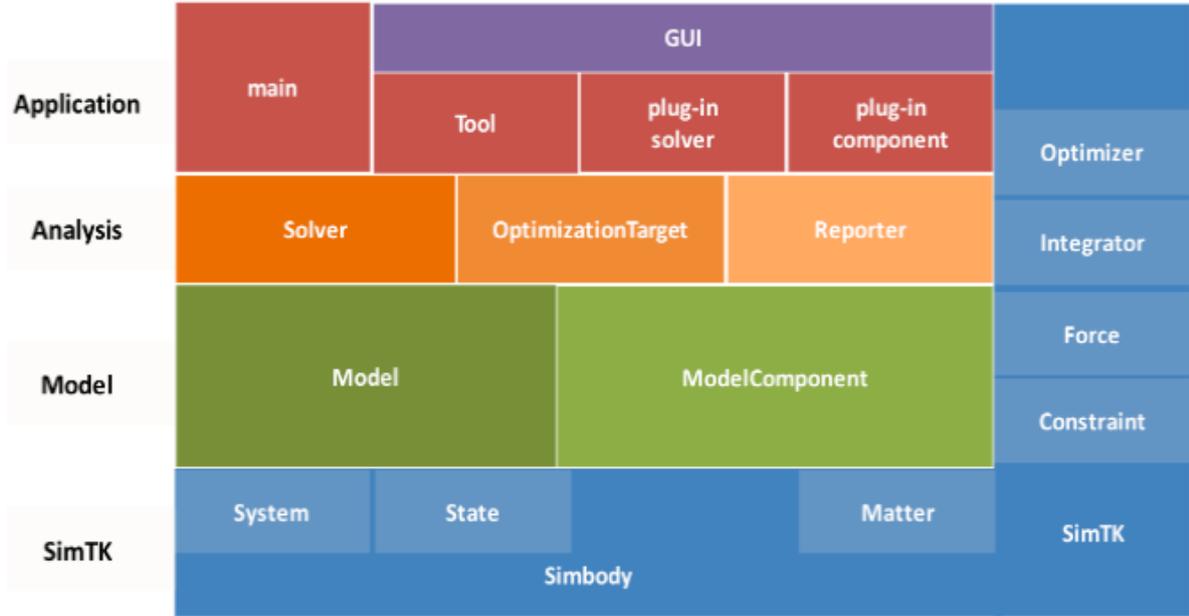


Figure 3.3 The three interface layers of OpenSim built on SimTK.
Image taken from [37]

OpenSim is written using object-oriented programming. There are seven most common classes of components that are used in OpenSim: Physical Frame, Joint, Constraint, Force, Actuator, Controller, and Probe. Each of these classes has sub-classes, like the muscle for the Actuator class which in turn has sub-classes like: Schutte1993Muscle_Deprecated, Thelen2003Muscle and Millard2012AccelerationMuscle. These are some of the muscle models available in OpenSim, with C++ you could create your own. The seven classes mentioned are part of the *ModelComponent*, that is the base set of all components that can be used to compose a Model [37].

The core of OpenSim is SimTK (Simbios "simulation toolkit"), it contains Symbody and LAPACK, a math and matrix algebra libraries. So OpenSim basically calls those Tools, Models, and underlying computations in SimTK via the GUI or the API.

With doxygen it is possible to see all the classes and methods of OpenSim, that is an overview of all the codes available, it can be found in the installation folder of OpenSim at the address: C:\OpenSim4.0\sdk\doc\OpenSimAPI.html or in the specific web page https://simtk.org/api_docs/opensim/api_docs/index.htm.

3.3 OpenSim files

In the description of the software OpenSim and of its tools made so far, some different types of files have been presented.

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <OpenSimDocument Version="40000">
3  <Model name="Human_exo">
4  <!--The model's ground reference frame.-->
5  <Ground name="ground">
21 <!--Acceleration due to gravity, expressed in ground.-->
22 <gravity>0 -9.8066499999999994 0</gravity>
23 <!--Credits (e.g., model author names) associated with the model.-->
24 <credits>Matteo Musso</credits>
25 <!--Publications and references associated with the model.-->
26 <publications>Simulation HMI whit OpenSim</publications>
27 <!--Units for all lengths.-->
28 <length_units>meters</length_units>
29 <!--Units for all forces.-->
30 <force_units>N</force_units>
31 <!--List of bodies that make up this model.-->
32 <BodySet name="bodyset">
3184 <!--List of joints that connect the bodies.-->
3185 <JointSet name="jointset">
5039 <!--Controllers that provide the control inputs for Actuators.-->
5040 <ControllerSet name="controllerset">
5130 <!--Constraints in the model.-->
5131 <ConstraintSet name="constraintset">
5387 <!--Forces in the model (includes Actuators).-->
5388 <ForceSet name="forceset">
8541 <!--Markers in the model.-->
8542 <MarkerSet name="markerset">
8571 <!--Geometry to be used in contact forces.-->
8572 <ContactGeometrySet name="contactgeometryset">
8576 </Contact>
8577 </Model>
8578 </OpenSimDocument>

```

Figure 3.4 Structure of an OpenSim model file

The file that contains the musculoskeletal model is in the format: “.osim”. It is a file with the contents organized according to an XML code structure. The XML format requires the use of tags to identify and manage all the information. An example can be the gravity tags (<gravity></gravity>) in the fig. 3.4. They contain the information about the gravity of the model, the first indicator represents the opening while the second the end of the tag. In this file the model elements are divided into sets like BodySet, JointSet, ForceSet, that contain all the elements of a single typology, and the relative parameters.

The motion file format is “.mot” was developed for the software SIMM (Software for Interactive Musculoskeletal Modeling) and it is also usable by OpenSim. The file has two distinct parts. The first is the motion header, it contains the number of rows and columns of the second part of the file: the data table. The format of the header changes a bit if it is for SIMM only or also for OpenSim. The header end with the line endheader. It can also contain lines of comments inside it. The data table has for first line the names of the column labels, they should match the name of the object of our model, to which they relate, for example, the name of the coordinate of the model of which define the values. The first column is usually the time, its values must be uniformly spaced. If this column is not present, for example in the SIMM only format, OpenSim take the time range reported in the header and divides it in steps.

The type of files .sto (storage file) was developed by the OpenSim team. Those files had a format very similar to that of the motion file previously described. They have a header and a data table but the first column of it must contain time, the values contained in this column do not have to be uniformly spaced.

The Track Row Column (.trc) files are used to specify the marker trajectory, acquired through movement analysis. It is used in IK Tools to calculate the joint coordinates of the model. They are not used in this thesis, since we had not carried out this type of analysis.

Finally, there are .xml file that are used to set up the tools or adding external actuator, like the reserve actuators, define the actuator controls or even to define the tracking task for the CMC Tool. They are well described in the OpenSim documentation where the meaning and use of the tags is explained.

3.4 Available musculoskeletal models

There are a good number of model available in OpenSim, which are present directly in the software installation folder or on the site “simtk.org”, where can be find also the forum of the community or other useful material like plugins, documentation etc.

The musculoskeletal models in OpenSim bibliography concern mainly the lower limbs of the human body, while those concerning the upper limbs are in small number.

Since this thesis relates to an exoskeleton for the upper limbs, only the related musculoskeletal models are presented below.

Arm 26

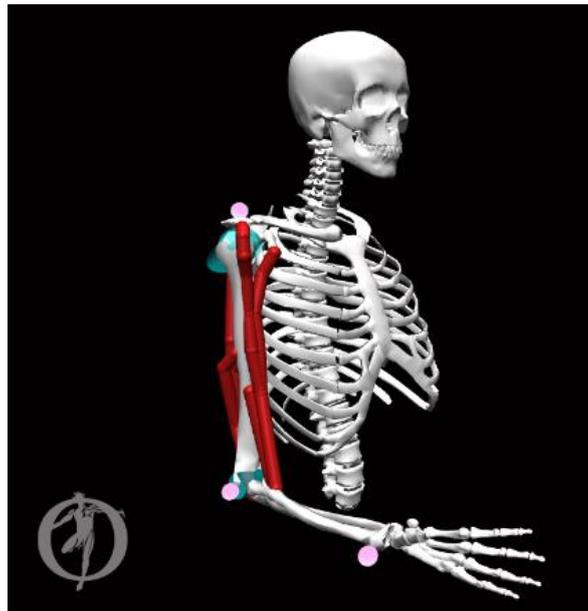


Figure 3.5 Arm26 model

It is a simple model of arm, the right one. It is composed of three bodies the thorax, the upper arm and the forearm, which combines radio, ulna and hand in a single body. The thoracic body comprises the geometries of the head, the dorsal spine and the thoracic cage, visible in the figure 3.5. Only the inertial parameters of the two arm bodies are defined.

The model has six muscles representing the three heads of the brachial triceps, the two of the biceps and the brachialis muscle. The muscle model used is the Thelen2003Muscle type.

There are only two degree of freedom one for the shoulder and one for the elbow. The two joints of the human model are equated at a rotary joint with a range of motion of 270 degree for the shoulder (from -90° to 180°) and of 130° degree for the elbow.

This model is presented in the installation folder of the software and is used to present the program and carried out some examples.

Wrist model

It's a model of a right hand and of the forearm. It presents a body for each bone of the hand and arm; whose geometry is visible. In addition, there are five bodies without geometry used to add enough joints to reproduce the ROM of the human wrist. To reproduce the movements of the wrist, the degrees of freedom of these joints were linked together. For no one of the bodies the inertial parameters are defined.



Figure 3.6 Wrist model.

It has a total of 10 degrees of freedom, one for the elbow, one for the radio-ulnar joint, two for the wrist, two for the thumb (flexion and abduction), two for the forefinger and the other two for the distal and intermediate phalanges of this finger. The other three fingers have no degree of freedom.

Finally, it has 25 muscles, controlling mainly the movements of the wrist. The *Schutte1993Muscle_Deprecated* model was.

Stanford VA Upper Limb Model

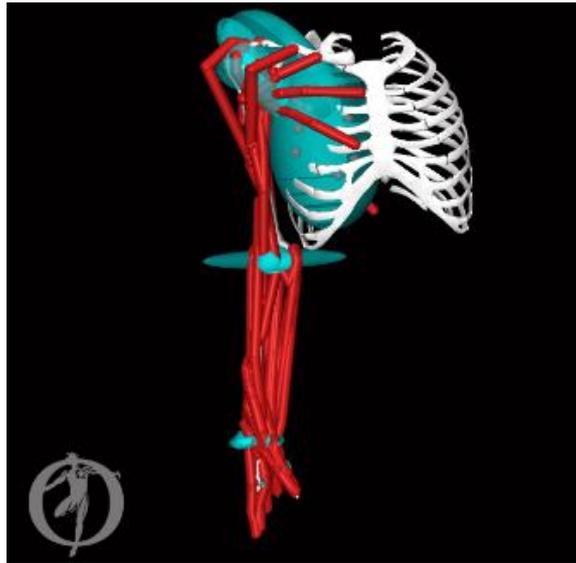


Figure 3.7 Stanford VA Upper Limb Model

Stanford VA Upper Limb Model is a model of the right arm developed by the Stanford researchers that present it and carry out some initial tests in the paper [38]. This model can only be used for kinematic analysis as the inertial parameters of the bodies that constitute it are not present. The length of the bones is based on the 50th percentile male. It has 50 muscles that simulate the behavior of the human ones. The model has 15 degree of freedom, 3 for the shoulder and the other DOF are implemented like the Wrist model to which 2 DOF are added for the two phalanges of the thumb.

Upper Extremity Dynamic Model (MoBL-ARMS)

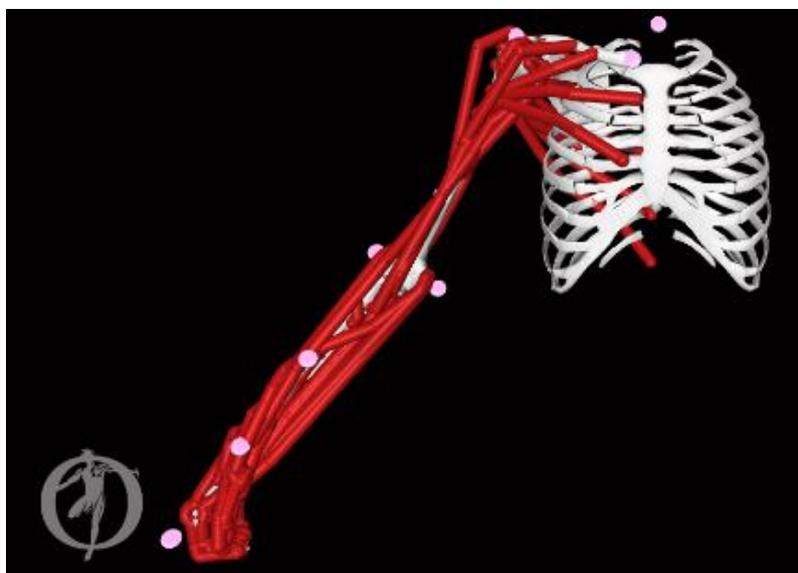


Figure 3.8 MoBL-ARMS model

The MOBL model is an evolution of the model previously described, it was developed by the same authors and well-presented and tested in their paper [39].

Like the name suggest it is usable for dynamic simulation because the inertial proprieties of the bodies were implemented. The inertial parameters of the bodies of the model have been defined except for the thorax, the values are reported in the table 3-1.

There are also other four bodies having a mass of 1.0E-4 Kg and an inertia tensor of (1 1 1 0 0 0) Kg*m² that are used to create the shoulder articulation and reproduce its movement. The reason for them is that to recreate the real ROM of the human shoulder more joints are needed, a joint in OpenSim can connect only two bodies and therefore more bodies than the real ones are needed.

The model has 50 muscles based on the Hill's muscle model; the muscles are of the type called: *Schutte1993Muscle_Deprecated* in the OpenSim API. The definition deprecated means that OpenSim no longer supports this type of muscle due to the presence of implementation error.

Bodies	Mass (Kg)	Centre of mass (m)			Inertia (Kg × m²)		
		C_x	C_y	C_z	I_{xx}	I_{yy}	I_{zz}
Clavicle	0.1560	-0.0111	0.0064	0.0054	0.0002	0.0003	4e ⁻⁵
Scapula	0.7039	-0.0547	-0.0350	-0.0437	0.0012	0.0012	0.0014
Humours	1.9976	0.0181	-0.1401	-0.0128	0.0123	0.0026	0.0126
Ulna	1.1053	0.0097	-0.0959	0.0243	0.0054	0.0012	0.0049
Radius	0.2336	0.0336	-0.1816	0.0156	0.0004	9e ⁻⁵	0.0004
Proximal_row	0.0001	0	0	0	1e ⁻⁵	1e ⁻⁵	1e ⁻⁵
Hand	0.5819	-0.0030	-0.0425	-0.0011	0.0001	6e ⁻⁵	0.0002

Table 3-1 Inertial parameters MoBL model bodies.

Compared to the model from which it derives the latter has three degrees of freedom less, i.e. 7, those of the fingers were eliminated, and the whole hand was considered as a single body, modelled in a gripping position.

CoordinateLimitForce (CLF) have been added at the coordinates of the model. These forces serve to limit the ROM of the coordinates, especially in Forward dynamics analysis because the imposed ROM is not respected in this type of simulation. Each coordinate of the arm of the MoBL model has two of this force. One has only the elastic component and a the other only damping one. As described by the author of the model in [39] the one with only the elastic component is active when the movement exceed the ROM of the coordinate, and the other it is active in all the ROM of the coordinates, to impose a constant damping.

Real Joint	DOF	Range	Coordinate name in OpenSim
Shoulder	Elevation angle	-90°-130°	elv_angl
	Elevation	0°-180°	shoulder_elv
	Rotation	-90°-20	shoulder_rot
Elbow	Flexion	0°-130°	elbow_flexion
	Pronation-supination	-90°-90	pro_sup
Wrist	Deviation	-0.175°-0.435°	deviation
	Flexion	-1.222°-1.221	flexion

Table 3-2 Coordinates of the MoBL model



Figure 3.9 UBL_Projet model

It is a full body model. It combine the Stanford VA Upper Limb Model described above used for the two arms, with the lower limbs model developed by Delp S.L. et al. named "Gait2354_Simbody". This last lower limbs model has 23 DOF and 54 muscles, it is present in the installation folder of OpenSim and a description can be found on the internet page [40]. This model does not present the inertial parameters for the upper limbs and the thoracic body (which includes the geometries of the head, thoracic cage and spine). Given the high number of degrees of freedom of the model (53) and of the muscles present (154) the model is difficult to use and has very long simulation times.

3.5 Creation and importing of the exoskeleton geometry

The work developed in this thesis leads us to import and test a model of an exoskeleton. To do that we had to create and subsequently upload the geometry of the elements of the device in OpenSim.

This software allows to import CAD model in the file formats *.obj* or *.stl*. Only the geometry of the body is being imported and other data and properties of the object like inertial parameters or joints and constraints must be redefined in OpenSim.

We had created the elements of our exoskeleton with the CAD software SolidWorks. Our device is composed by three parts. The first is connected with the back of the musculoskeletal model. The second part models the upper arm and the third the forearm of the exoskeleton. These last two elements were used also as simple loads in the first tests. How these bodies are connected to each other and to the musculoskeletal model will be seen in the following paragraphs.

OpenSim don't need the geometry of the bodies to handle and carry on the simulation, it needs only its inertial parameters and its connection with the rest of the bodies. Having a geometry to look at, helps us both in the preparatory phase and in the subsequent evaluation phase.

The bodies of the of the exoskeleton have been separately created in the SolidWorks environment starting from a cylindrical geometry with a radius of 25 mm. The length of the two elements have been taken in order to reproduce the length of humerus and forearm of the human model used. The exoskeleton upper arm had a length of 350 mm while the forearm of 250 mm. The extremities that will be connected to create the exoskeleton elbow have been worked in order to give them the shape of the two part of a revolute (or pin) joint.

The body that constitute the back of the exoskeleton has a "L" shape. This form has been chosen to match the human form and align the shoulders of the two parts of the model.

It has been connected with the center of the back in the human model and to the arm bodies of the exoskeleton, along its shorter side, to reproduce the human shoulder. The "L" shape element had a length of 270 mm for its longest side and of 190 mm for the shortest. The section is of 40 x 50 mm.

Once created the geometry the inertial properties were evaluated thanks to SolidWorks, by clicking on the Mass Properties option of the Tools toolbar or going to Evaluate, Mass Properties, in the Tools drop-down menu. The weight of the three bodies was imposed of 1 Kg. OpenSim need in addition to the mass of the body also its center of mass, expressed in the local reference system, and its tensor of inertia, measured about the center of mass of the body. Furthermore, whit SolidWorks the local reference system has defined, by clicking on the specific icon in the toolbar. In this way can be positioned the local reference system respect which the inertial properties are assessed.

OpenSim works with the system of unit MKS (meter, kilogram, second), to ensure the compatibility of the model created, SolidWorks must be set up with this system. It is also important to remember that the direction of the X and Z axis of the global reference system are different between SolidWorks and OpenSim, they are exchanged. For OpenSim the X axis is the anterior-posterior one (fig. 4.1) but for SolidWorks is the medio-lateral. For the Z axis is the opposite.

Once their geometries were created and their properties evaluated, the three bodies are been saved in the .stl format, with the option “Do not translate STL data to positive space” checked. In this way the file saved also contains the local reference system created by us. It is read and imported by OpenSim.

Body	Mass (Kg)	Inertia ($Kg \times m^2$)			Center of mass (m)		
		I _{xx}	I _{yy}	I _{zz}	C _x	C _y	C _z
Back “L”	1	0.007322	0.009827	0.003012	0.053402	0.035000	0.175603
Upper Arm	1	0.007221	0.002965	0.007252	0.025004	0.209006	0.002006
Forearm	1	0.007090	0.000280	0.007030	0.019570	0.139100	0.0250300

Table 3-3 Inertial parameters of the bodies of the exoskeleton mode

To import and connect our geometry to the OpenSim model that we want to use, we had to modify its .osim file adding our new components. It is a .xml and we can modify it directly with a program like Notepad++, or in a more efficient way by using the OpenSim API. For this thesis MATLAB has been used to import, connect and define the proprieties of these bodies.

When you import a body in OpenSim you have to define how it is connected with the rest of the space and the other bodies. To do that you can use different types of joints, they define the GDL of the body respect the one to which it is connected. OpenSim utilize a hierarchical structure, so the body respect to which the joint allows movements it is called “Parent body” and the body that moves is the “Child body”. Each body can have only a “Parent body” but more than one “Child body”.

OpenSim provides eight types of joints:

- *WeldJoint*: this type of joint doesn't allow relative motion between the bodies of it;
- *SliderJoint*: it provides only a translational DOF along the X axis of the reference system of the parent and child joint frames;
- *PinJoint*: only a rotational DOF is permitted, about the Z axis of the joint;
- *PlanarJoint*: it has two translational DOFs about the X and Y axis and a rotational one about Z;
- *BallJoint and GimbalJoint*: they allow the three rotations about the point where the two bodies are fixed, the *BallJoint* has not the singularity that is present in the *GimbalJoint* when the Y is near $\pi/2$.
- *EllipsoidJoint*: it provides three mobilities – coordinated rotations and translations along the surface of an ellipsoid fixed to the parent body [41];
- *FreeJoint*: three rotations and three translations about the axis are allowed;
- *CustomJoint*: you can define behavior of the joint, specify the position of the child compared to the parent body as a function of coordinate.

To position the body correctly with respect to the model or the rest of the OpenSim environment you can use the GUI. You can add a marker in the point of the model where do you want to create the connection. In this way you can take the coordinates of this point in the frame of the model body that will be used as “Parent body” for the joint.

In the next figure 3.10 it can be seen a piece of the MATLAB code written to import one of the new bodies into the model. It is also an example of the API codes used to interface the two programs.

```

5 - import org.opensim.modeling.*
6 - path='C:\OpenSim 4.0\Geometry';
7 - ModelVisualizer.addDirToGeometrySearchPaths(path);
8
9 - [Model_In, path] = uigetfile('.osim');
10 - filepath = [path Model_In];
11 - model = Model(filepath); % Create a copy of the original OpenSim model for the Modified Model
12 - r = model.getBodySet().get('radius');
13
14 - foreram= Body();
15 - foreram.setName('Exo_forearm');
16 - foreram.setMass(1);
17 - foreram.setMassCenter(Vec3(0.01957,0.13810,0.02503));
18 - foreram.setInertia(Inertia(0.00703,0.00028,0.00709));
19 - foreram.attachGeometry(Mesh('Exo_forearm.STL'));
20 - model.addBody(foreram);
21
22 - exoElbow = WeldJoint('w',r,Vec3(0.0018163,-0.139088,0.101171),Vec3(0), ...
23 - foreram,Vec3(0.024403,0.130846,0),Vec3(0));
24 - model.addJoint(exoElbow);
25
26 - m=Marker();
27 - m.setName('M_f');
28 - m.setParentFrame(foreram);
29 - m.set_location(Vec3(0.024403,0.130846,0));
30 - model.addMarker(m);
31

```

Figure 3.10 MATLAB code wrote to insert the new body.

It refers to the model of the first set of tests that have been done in this thesis, where the body, representing the forearm of the exoskeleton, is connected to the musculoskeletal model and acts as a simple weight.

The first lines, not visible in the figure, are used to clear the MATLAB workspace and the command window. This is quick way to "reset" the software.

The next line, the fifth, of the code is used to import the OpenSim libraries that contain the OpenSim classes of object. The lines 6 and 7 indicate to MATLAB the address of the folder where the bones geometries of the model are in our computer. For convenience, it is recommended to insert in this folder also the files containing the geometries of the bodies that have been created. The lines of code 9-10-11 take the model that we want to modify and make a copy. Then in line 12 it is defined the body of the model to which we want to connect our weight.

The lines from 14 to 20 define the new body: its name, its proprieties, its geometry and add it to the model. If we don't define how this body is connected to the rest of the model, it is automatically added with a *FreeJoint* to the ground. From the line 22 to 24 the type of connection used for this body has been defined, a *WeldJoint* was used. The point of action of the joint and the orientation on the two bodies has been defined.

The lines 26 to 30 define a marker that has been added to display in the GUI the point where the joint is present, otherwise it would be invisible. The defined marker has also been used to identify the correct position of our metallic structure related to the arm of the model.

The MATLAB file end with three lines of code:

- `model.finalizeConnections();`
- `state = model.initSystem();`
- `model.print('Name_of_our_model.osim');`

These three lines of code, are used to finalize the connections in the model, initialize the state. The state contains the variables of the model such as the position, the speed of its coordinates, the length and activation of its muscles. The last command prints the model file that we have created.

4 Materials and methods

Our work on the musculoskeletal model start from the evaluation of simple movements regarding only one joint of the model, in different condition. We compare a model free with a model with one or two bodies linked, acting like simple weights. These initial trials are done to better understand the kinematic and dynamic behavior of the “MoBL-ARMS” model. This is the human model that we will be interfaced with our exoskeleton. The purpose of these tests is also to become familiar with OpenSim and the tools that it makes us available. Furthermore in this way we have control cases to use for comparing the behavior of our exoskeleton.

4.1 Elbow tests

The first performed tests were relative to the elbow coordinate of the model (“elbow_flexion”). The behaviour of the model under kinematic and dynamic input has been tested. For each of the two cases a model free and one linked to a metallic element were used, it acts has a simple weight. This body has been linked to the forearm of the MoBL model (fig. 4.1). It has been connected with a *WeldJoint* on the radius of the musculoskeletal model. This element was the one described previously and that will then be used as forearm of our exoskeleton. As reminder it has a mass of 1 kg with a length of 250 mm and a radius of 25 mm.

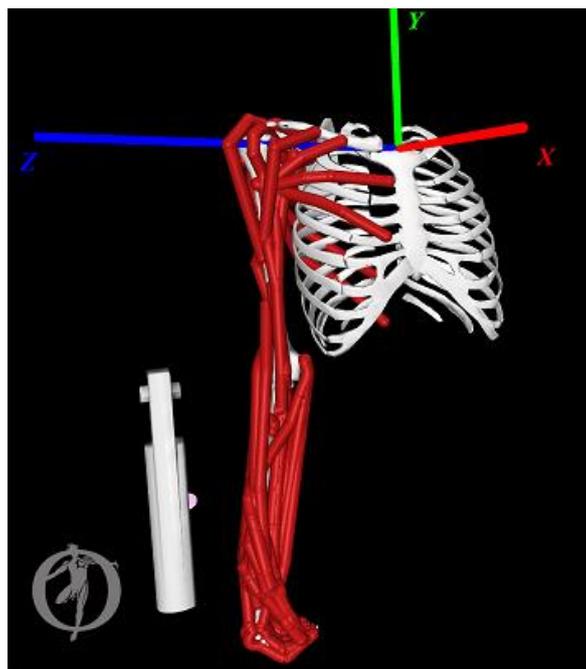


Figure 4.1 The linked model and the global reference system.

The first test consists of a simple flexion of the elbow from 0° to 140° in 2 seconds, all other coordinates were locked.

To simulate this movement the Forward Dynamics Tool of OpenSim has been used. To perform the simulation, we have defined a function at the "elbow_flexion" coordinate of the musculoskeletal model, so that OpenSim can perform the task. This function was written thanks to MATLAB in the .osim file of the musculoskeletal model which will then be read by OpenSim. The *StepFunction* method of the OpenSim API was used by setting the initial and final value of the coordinate taken in consideration, and the interval of time during which the movement takes place.

The muscles of the model were not active, no control value was provided so it was 0, the default one. During the simulation the muscles will follow the movement of the forearm passively. To make a comparison between the free model and the linked model used for this test, the torque needed to carry out the movement has been plotted. This was done by running an inverse dynamics (ID) analysis, thanks to dedicated OpenSim tool, with the kinematic results obtained with the FD analysis.

In the second group of tests relative to the elbow, the behaviour of the model under dynamic input has been evaluated. A torque is applied to the elbow coordinate of the human model with and without an added link to induce a flexion. This torque has a constant value of 5 Nm. It has been implemented adding, always with MATLAB, a *CoordinateActuator* at the "elbow_flexion" coordinate of the models.

According to the doxygen of OpenSim this type of actuator provides a generalized force in the direction of the coordinate. It can generate a force or a torque depending on the type of DOF provided by the coordinate, translational or rotational respectively. A value called optimal force must be set, OpenSim will multiply it with the control value provided to the actuator.

To allow the actuator to provide its couple a controller must be added. If a controller is added to the model it will control all the actuators, and so also the muscles, of the model. In these simulations it was set to provide a control value of one to the actuator, and zero to muscles of the model. As reminder, in the FD tool an actuator does not provide any force if it does not get a control value different from the default one, zero. The control values can also be defined with a specific file, without defining a controller.

So, two tests have been done:

- Kinematic test: elbow went from 0° to 140° in 2 seconds, with a constant speed;
- Dynamic test: torque of 5 Nm imposed at the elbow

4.2 Elbow tests results

To make a comparison between the behaviour of the free model and the linked model used for kinematic test, the torque needed to carry out the movements has been plotted. This was done by running an inverse dynamics (ID) analysis, thanks to dedicated OpenSim tool, with the kinematic results obtained with the FD analysis.

The result can be seen in the fig. 4.2, as it was expected, the moment needed by the model with the metallic body linked is bigger than the free one. The results aren't too different, the torques needed for the two movements differ by no more than 2 Nm, but this result was expected due to the low weight of the linked body.

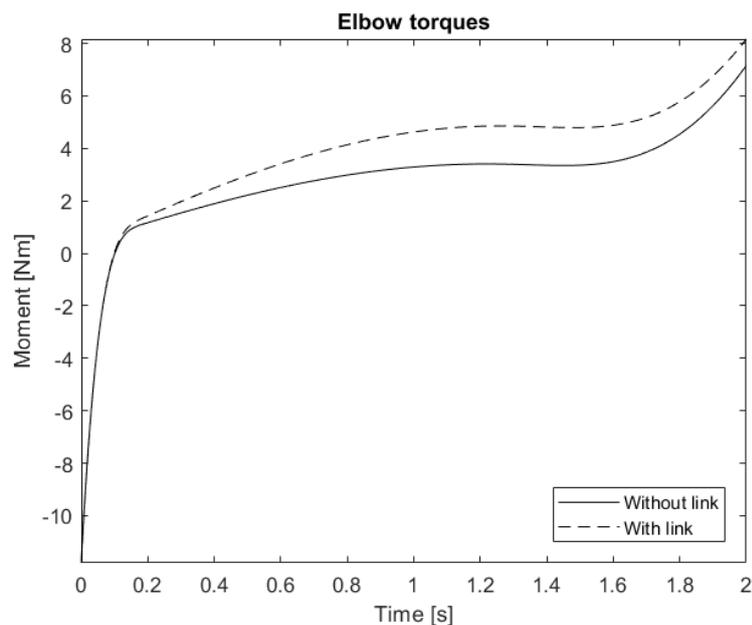


Figure 4.2 Elbow torques necessary for a flexion from 0° to 140° .

Fig. 4.3 shows the results of the second group of simulations, where the models were tested under dynamic input. There is the comparison between the angles (fig. 4.3A) and the flexion speeds (fig. 4.3B) of the elbow of the two models. As expected, the kinematics of the free model is faster than the linked one.

In these two tests, two particular behaviors were found, the second of which is a direct consequence of the first one.

In the test where the flexion of the elbow was simulated by imposing a function, the movement goes from 0 to 140° , even if the ROM of the elbow of the model used has an upper limit of 130° . The possibility of exceeding this limit is due to the fact that using the Forward Dynamics Tool of OpenSim the ROM of the coordinates is not respected. To avoid to overcome the limits

it must be inserted the *CoordinateLimitForce* (CLF); the MoBL model has two of this force for all the coordinates of the human arm. This force has an elastic component and a damping one, and this is why the oscillatory behaviour visible in the graphs of the fig. 4.3 was obtained.

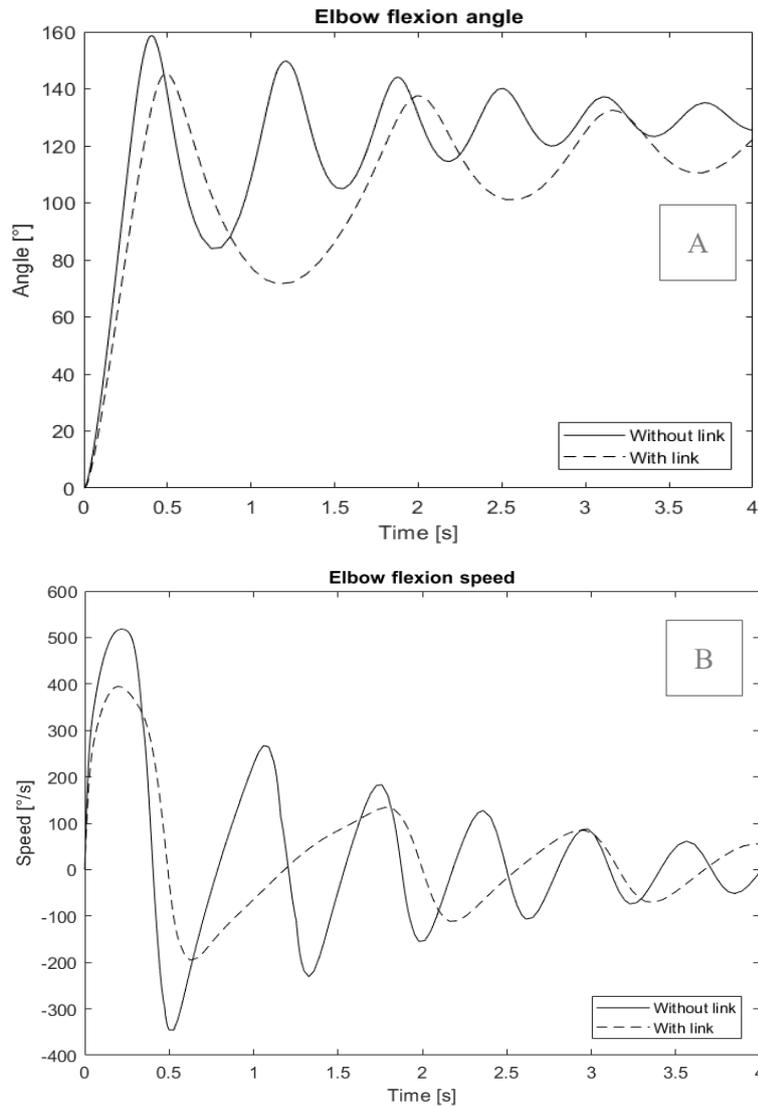


Figure 4.3 A) Elbow flexion angle for the two models.
 B) Elbow flexion speed for the two models.

The sinusoidal trend of the value of the angle suggests that there is an average value and if the simulation was continued for a longer time, we would see that the value will become constant equal to the upper limit of the ROM. As said in the presentation, the MoBL model has two groups of *CoordinateLimitForce*, one has only the elastic component and it is active when the movement exceeds the ROM, and the other has only the damping component and it is active in all the ROM of the coordinates. The values of parameter used in the CLF are reported in the

paper [39] where the model is presented. These forces are a mathematical adaptation of the contribution that the tissue of the human body passively provides during movement.

It is interesting to see the value that these forces assumed during the elbow flexion.

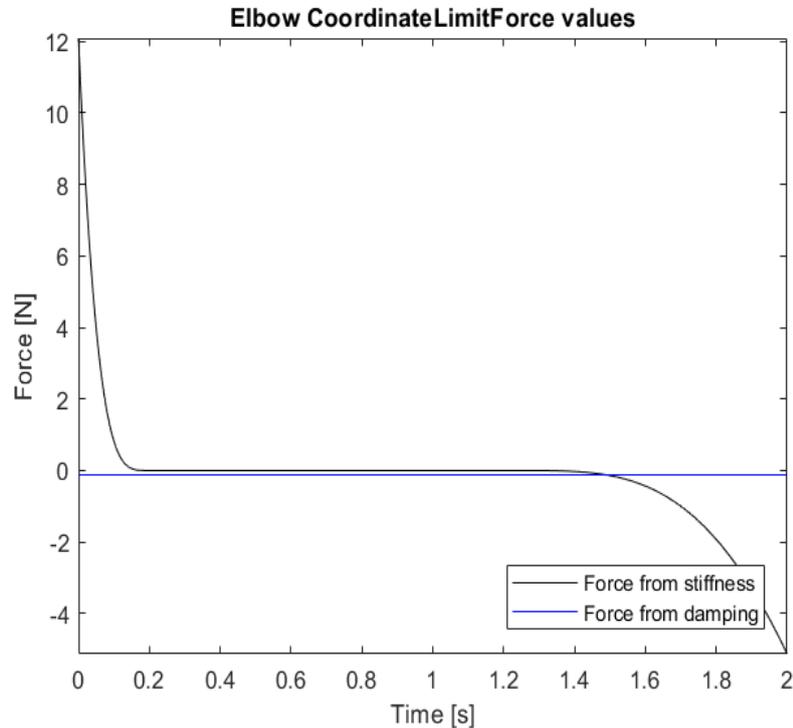


Figure 4.4 Force generated by the two CLF of the elbow

The figure 4.4 shows the value of the two CLF that act on the elbow coordinate during the first test, the one with the coordinate at the elbow, for the free model. These values of force were obtained adding a ForceReporter analysis at the FD simulation.

The first thing that can be noted is that only the *CoordinateLimitForce* with just the stiffness component generates a considerable force. This CLF influence a lot the torque needed to make the flexion especially in the first and last part of the motion. In these two intervals the torque required is the opposite of the one generated by the CLF.

The last thing that can be noted is that the CLF gave a force before that the ROM was overtaken. For the elbow the CLF generates a force when the coordinate exceeds the 100°. The maximum value is reached at a value of 160°. This is due to the fact that the CLF don't provide instantly the max level of force but there is a transition interval.

4.3 Shoulder tests

Then we started to analyse the shoulder. We have again carried out two types of simulation, with a kinematic or with a dynamic input.

As for the elbow two models were used, one free (the MoBL model fig. 3.8) and one with two connected bodies that can be seen in fig. 4.5. The second body is the upper arm presented above (mass of 1 kg, length 350 mm, radius 25 mm). As the forearm body, this second one is connected to the MoBL arm with a *WeldJoint*, between it and the humerus.

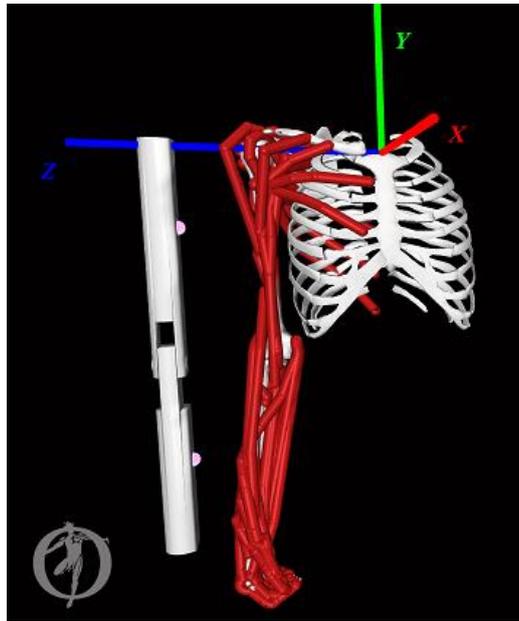


Figure 4.5 The model with two links at the upper arm and at the forearm and the global reference system.

For the first of these tests a motion from 0° to 120° in 2 second has been imposed to the arm of the model. A *StepFunction* that allows us to obtain the movement is defined in the “shoulder_elv” coordinate of the musculoskeletal model. The other coordinates were locked at the default value except the “elv_angle” set at 90° . With this configuration a shoulder flexion in the sagittal plane of the body is obtained.

For the dynamic test a constant torque of 20 Nm was applied at the shoulder coordinate of the two models, the free one and that of the fig. 4.5, to obtain the shoulder flexion. Like before a *CoordinateActuator* has been added at the “shoulder_elv” coordinate. For each of the two tests the muscles were not active and passively followed the movement.

So, two tests have been done:

- Kinematic test: shoulder moves from 0° to 120° in 2 seconds, with a constant speed;
- Dynamic test: torque of 20 Nm imposed at the shoulder.

4.4 Problems and errors in the MoBL model

In these trials, relative to the shoulder, some problems have arisen in the musculoskeletal model. The main problem is that, under certain conditions, OpenSim failed to calculate the right path of some muscles, which could result to the interruption of the simulation.

As we will see this problem happens especially for two muscles of the model: the "DELT1" and the "BIClong". The first one represents the anterior deltoid of the human body, the second muscle is the biceps long heads.

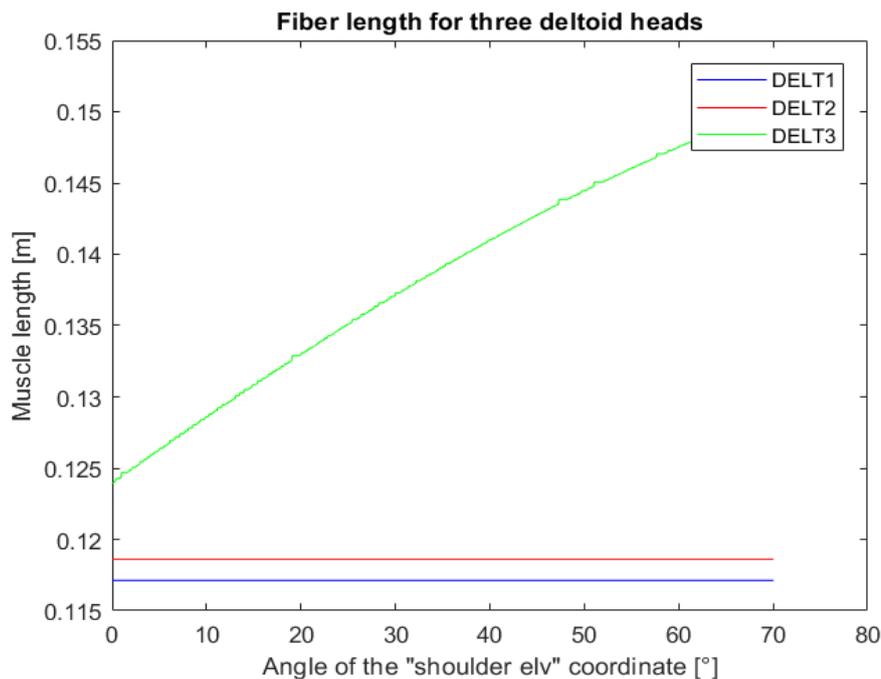


Figure 4.6 DELT muscles length during the shoulder flexion, for the *Schutte1993Muscle Deprecated* model.

In the fig. 4.6 it is represented the length of the three DELT muscles of the model and how it changes during shoulder flexion. These muscles represent the three heads of the human deltoid. As can be seen only the one relative to the muscle that stretches, the "DELT3", is correctly represented while that of the muscles that should contract remains unchanged. This issue happens for all the muscles of the MoBL model and it's probably due to the muscle model used for them: *Schutte1993Muscle Deprecated*. Like the name suggest this muscle model is no longer upgraded and some functionality of OpenSim are no longer supported. Probably the software calculates the right length of the muscle during the simulation but don't plot the results. We also tried to change the muscle model with the *Thelen2003Muscle* typology.

To do that we use Notepad++ to edit the model file and change the three muscles. Most of the parameters of the two muscles models are in common and the remaining, necessary for a correct implementation of the model *Thelen2003Muscle*, are standard parameters for human muscles and therefore have been taken from the paper [42]. The parameters relative to a young adult have been used, the same values are also adopted in other models implementing this type of muscles, like the Arm26 one. We want to remind that in this test the muscles were not activated. The change of muscular model was made only to see what the right trend of the muscular length of the three deltoids would be (fig. 4.7). In this figure it can be seen the trend that we expected. The posterior deltoid (DELTA3) stretches in the same way of the figure 4.6, and the anterior and the lateral deltoid (DELTA1 and DELTA2 respectively) shorten following the flexion movement of the shoulder.

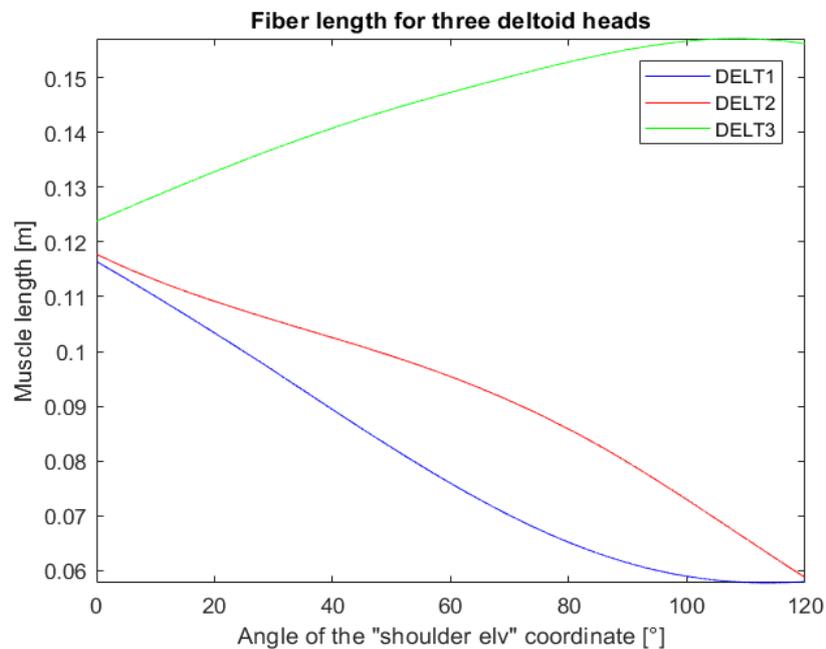


Figure 4.7 DELT muscles length during the shoulder flexion, for the *Thelen2003Muscle* muscle model.

From the figure 4.6 we can see that our simulation stops when the shoulder flexion angle reach about the 70° and does not reach the final value imposed (120°); this is not due to the previously described plotting error.

The model has been further investigated and it has been found that, at the angular value where the simulation stop, there is a sudden change in the path and therefore in the length of the “BIClong” muscle at the height of the humeral head.

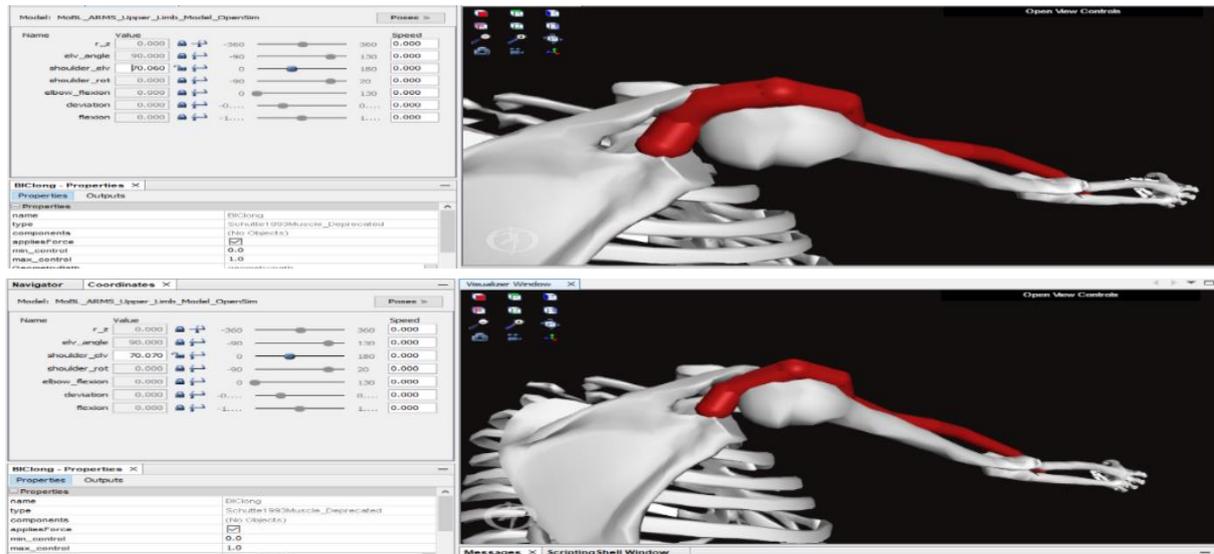


Figure 4.8 Change in BIClong change of path.

The fig. 4.9 shows that, in 0.01° the “BIClong” passes from following the geometry of the bone to intersect it. This configuration remains unchanged for all angles greater than 70.07° .

This wrong behavior is probably the cause of the failure of our simulation. We tried to analyze the trend of the length of this muscle, and we found an unexpected error.

The figure 4.9 shows a second issue, at a “shoulder_elv” angle of 63.64° OpenSim fails in the calculation of the right path of the muscle “BIClong” that instead of contracting it stretches.

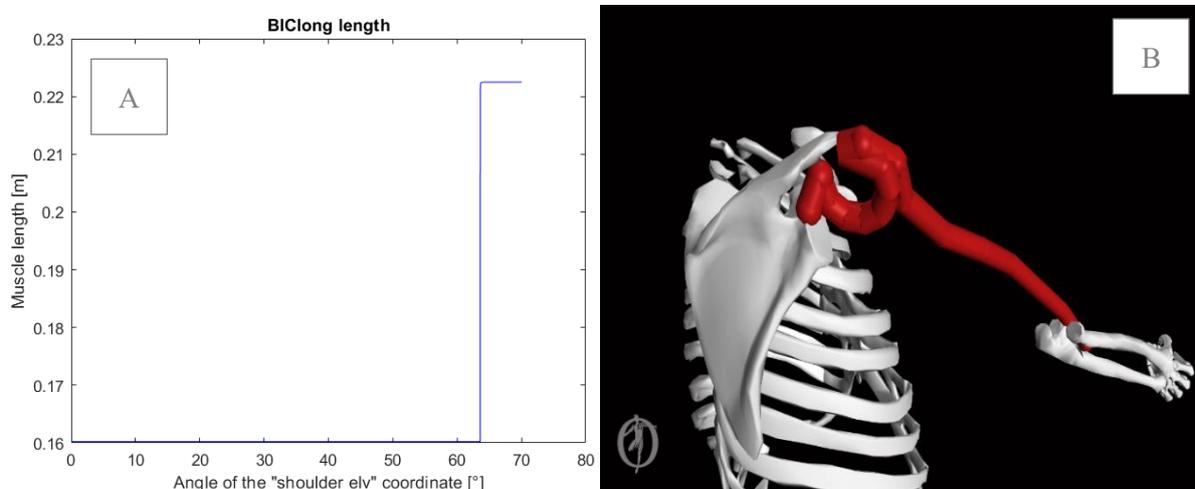


Figure 4.9 A) BIClong length during a shoulder flexion.
B) Muscle during the simulation.

By taking a screenshot (figure 4.9B) of the model when the shoulder angle reaches this value, we can see what happens. OpenSim fails to calculate the right path of the muscle between two points *PathPoint*. They are the points used to define the muscular path in OpenSim and can be fixed or mobile. This error is not due to the type of muscular model used, but precisely to a calculation error in the path. We tried to change the model used with the *Thelen2003Muscle*, but the error nevertheless occurred. By changing model it was seen that when the shoulder exceeds about 64.2° OpenSim calculates again the path and so the length correctly.

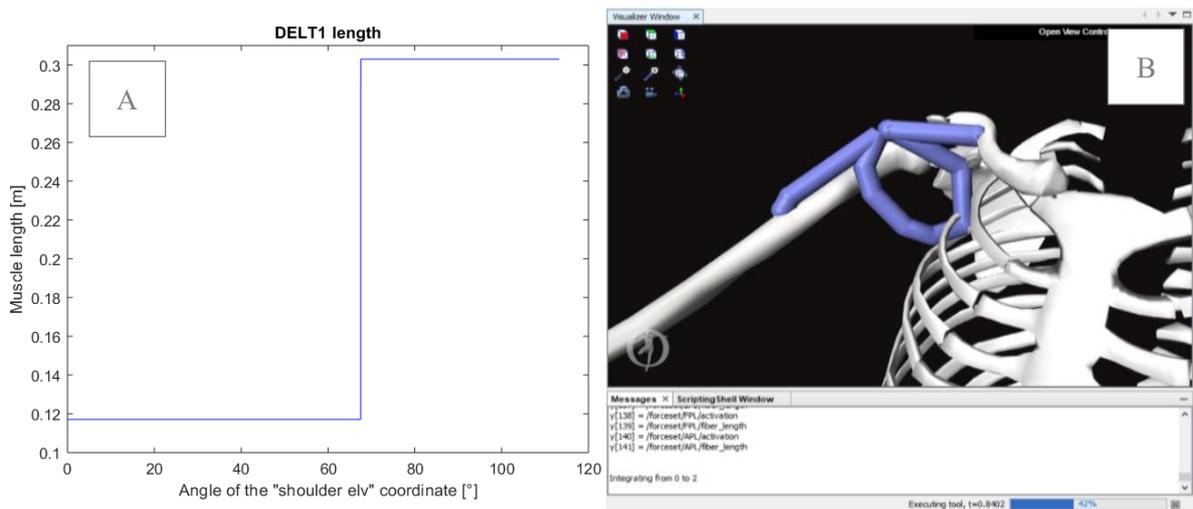


Figure 4.10 A) DELT1 length during a shoulder flexion with an *elv_angle* of 80° .
 B) Muscle during the simulation.

We notice that this type of error occurs also for other muscles of the model, like the “DELTA1”. Analyzing this muscle, it has also been seen that the value of flexion angle at which this type of error happens depend from the value of other coordinates of the shoulder of the human model.

For example, for the “DELTA1” it happens at a value of flexion of about 67.5° if the "elv_angle" coordinate of the shoulder is set at 80° . But, if this last angular value is set at 30° the “shoulder_elv” coordinate reach a value of 110° , before the error occurs.

The behavior of the “DELTA1” muscle at a “shoulder_elv” coordinate angle of 67.5° can be seen in the figure 4.10. Instead of contracting it stretches, in a step function way from 0.117 to 0.302 m.

From these figures (4.9 and 4.10) it is also clearly visible how for this muscle model OpenSim plots the muscles correctly only when they stretch and not during contraction.

It was seen that for the “BIClong” the error occurs at angle of about 47.4° if the “shoulder_rot” coordinate has a value of 10° and the “elv_angle” was set a 90° .

This type of error can lead to the failure of the simulation. However, it has been seen that increasing the “Maximum number of steps” parameter of the FD Tool helps to complete the simulation. This parameter modifies the step size of the analysis, it helps but don’t solve the problem. To complete our simulations, we sometimes had to disable the “BIClong” muscle, as the presence of the two errors prevented it from running properly.

The last thing that was noticed is the following: if from the obtained kinematic data, we make an ID analysis we get peaks in the torque necessary to make the movement. This behavior can be seen in the figure 4.11 were, after that a flexion of the shoulder was performed, the torque necessary to perform the movement was calculated. The first peak in this figure was generated by the biceps and the second by the deltoid.

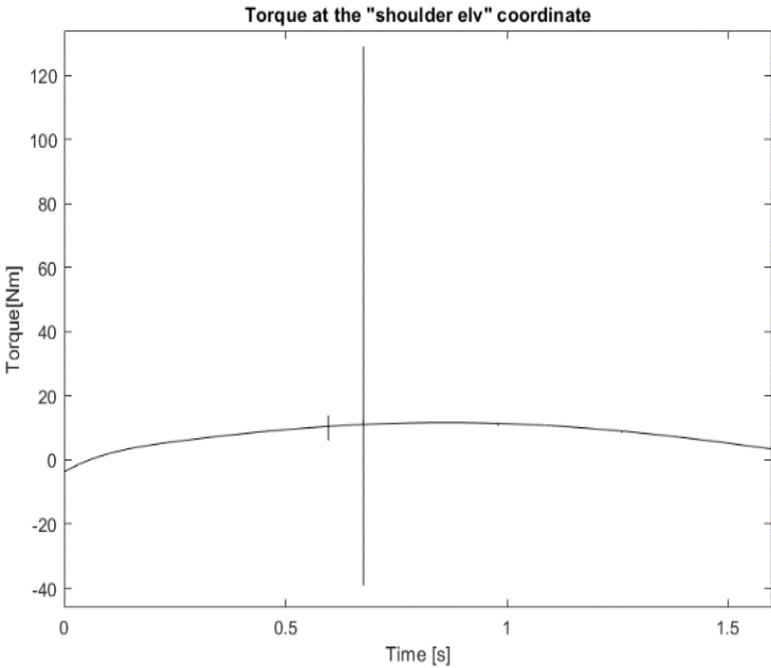


Figure 4.11 Torque necessary for a shoulder flexion

4.5 Shoulder tests results

After making this digression we can return to the initial purpose of this test, the comparison between the model with and without the two links, when a kinematics was imposed at the shoulder coordinate.

To confront the two models, we perform an ID analysis with the kinematics result obtained from the FD one. To complete the FD the muscle BIClong was disabled, however the muscles, during the simulation, were not activated and followed the movement passively.

In the fig. 4.12 there is a confront between the couple necessary to perform the movements with the two different model. The peak joint torques are reached at about 90° of shoulder flexion angle. The maximum value achieved by the two curves is 11.58 Nm for the one relative to the model free and 16.91 Nm for the linked model. The curves start form a negative value of torque. This is due to the presence of the *CoordinateLimitForce*, it is active until the angle exceeds 30 degrees. Without it the curve, relative to the free model, would start from a value of about 1 Nm.

The results can be compared with those reported by the paper [43]. The maximum torque values obtained are comparable with their, 16.281 Nm at an angle 90° for a shoulder flexion with a weight of 1 kg in the hand of the subject.

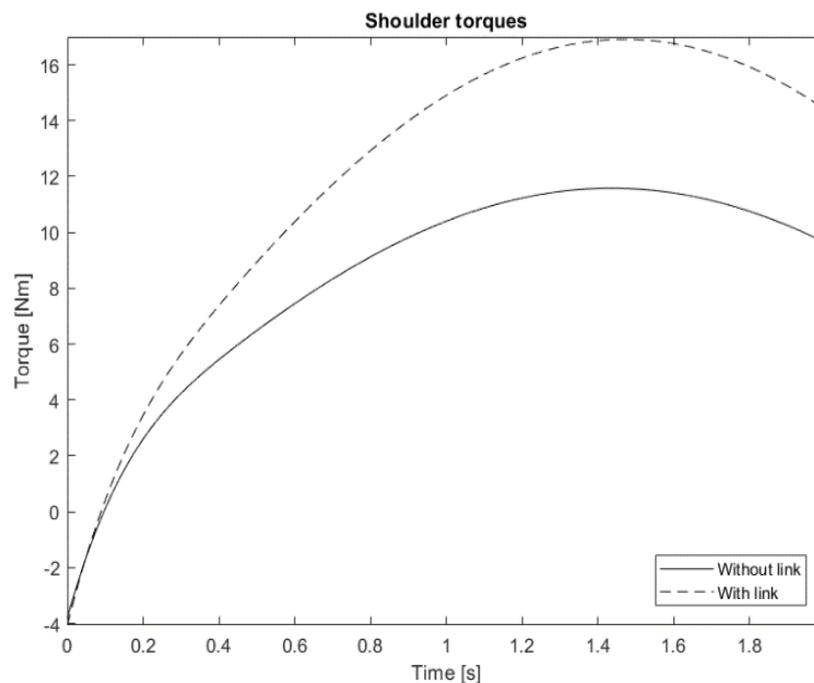


Figure 4.12 Torque at the “shoulder_el” coordinates.

After that we can proceed to compare the results of the dynamic tests. In these trials there is a greater importance of the problem relating to the muscle "BIClong", so, to not let the simulation fail, it was disabled.

In the figure 4.13 there is the comparison between the resulting kinematics of the two models. As expected, the flexion velocity is lower for the model with two links than for the free one.

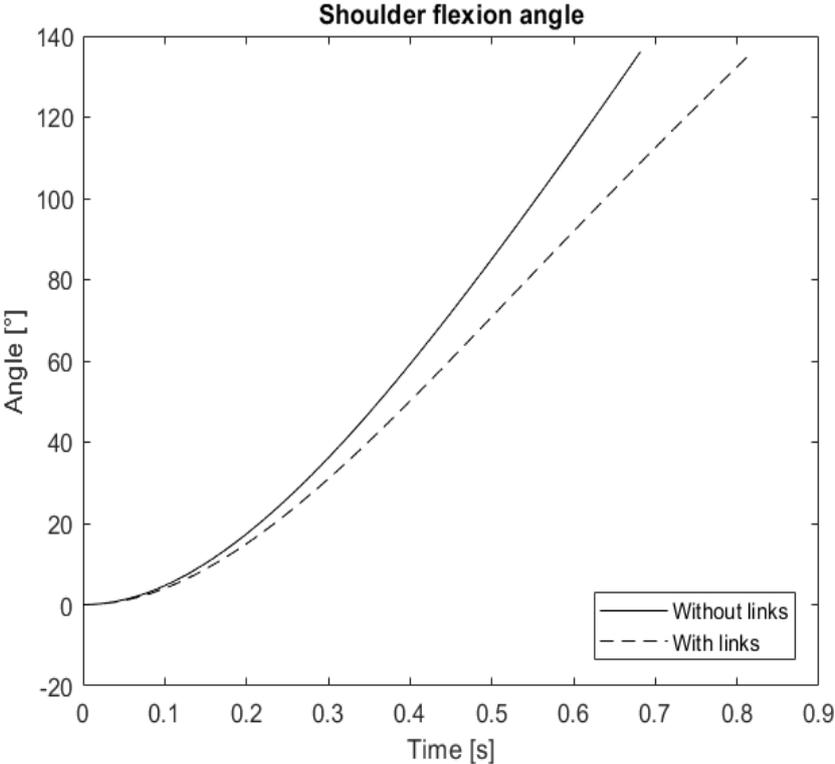


Figure 4.13 Shoulder flexion angles.

5 Exoskeleton development

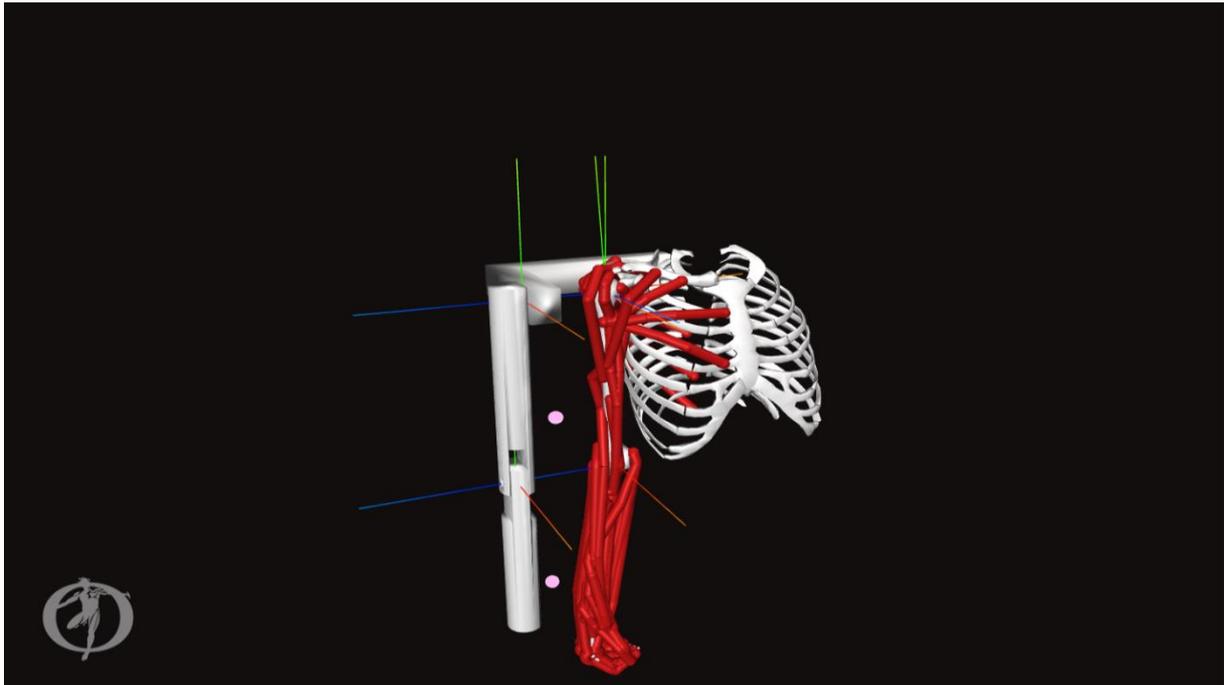


Figure 5.1 Musculoskeletal model coupled with the exoskeletal one.

After verifying the functioning and behaviour of the available musculoskeletal model we can start the main argument of this thesis: coupling an exoskeleton and studying the resulting behaviour.

The final configuration obtained is shown in the figure 5.1. The three metal elements previously described have been coupled to create our exoskeleton.

The device was connected to the musculoskeletal model with a *WeldJoint* between the back of the thorax of the human model and the back “L” body of the exoskeleton. In this way we simulate a wearable device that unloads his weight directly on the chest.

The upper arm body of the exoskeleton has been initially connected to the back one with a *PinJoint*. The joint obtained represent the shoulder of our exoskeleton model. The axis of rotation of this joint match the Z axis of the refence system expressed on the child body frame of the “shoulder1” joint in the MoBL model. The “elv_angle” coordinate has been set at 90°.

The upper arm body is then connected to the forearm with a *CustomJoint* to create the elbow of the exoskeleton. This joint was a reproduction of the human model one. The two rotation axes used to define the movement were the same of the elbow joint of the human model. This has been done to reproduce the movement of this articulation, to not introduce kinematical mismatch between the two joints.

It was not possible to make this simplification for the shoulder of our exoskeleton, due to a higher complexity of this articulation.

To recreate the movement of the real human shoulder a system of seven joints have been used, in the MoBL model. Furthermore, the movement of the *shoulder_elv* coordinate binds the movements of 10 coordinates belonging to five of these joints. There is a *CoordinateCouplerConstraint* that allows to define the value of one coordinate (dependent) consequently to another (independent). The second coordinate of the joint of the *shoulder_elv* is also coupled to the *shoulder_rot* coordinate. This allows the movement of the shoulder from the frontal plane to the sagittal.

It is impossible to reproduce the movement with only a *PinJoint*. We anticipate that it will be replaced by a *PlanarJoint* due to a too high kinematic incompatibility between the two shoulder models.

Finally, the arms of the exoskeleton and the human model were linked together in two points. These points have been positioned between arm and forearm of the two models, these points can be seen in the fig. 5.1, where there are the two pink spheres.

In the real exoskeleton the arms are usually linked with one or more bands around the human arm. To model the connection between these points we initially thought of testing two different ways.

The first is to use constraints between the exoskeleton and the body, as it was done in some articles described before, for example the one on the exoskeleton for upper limb, paper [31], where the *WeldConstraint* method was used. The second possibility is to use the *BushingForce*, as done by D. Coll Pujals in the paper [25]. This type of force can be thought as being composed of 3 linear and 3 torsional spring-dampers that act between the bodies. [44]

We first tested the method with the constraints, like *WeldConstraint*, but we ran into some problems. Using the constraints, we had great difficulty to obtain a value for the parameter “assembly error tolerance” below $1 \cdot e^{-10}$ as requested by OpenSim. The method *WeldConstraint* define a constraint between two frames, that can be two bodies, of the model. To reach the requested value of assembly error tolerance it has been tried to obtain very precise coordinates of the point of application of the constraint in the system of reference of both interested bodies. To do that we had to run a PointKinematics analysis with the dedicated tool, with a high value of precision of the output. This was however useless. OpenSim tries to satisfy the constraints of the model by changing the coordinate values. This generates a strange kinematic behavior with the model that assumes particular positions, especially during the

movement. This is probably due to the high kinematics incompatibility of the two part of the model, and the high number of joint of the human model shoulder. For these problems the method with the constraints has been discarded.

The method with bushing force was taking in to consideration and investigated. The principal aspect to define for this type of force is the value of its damping and stiffness components. In [25], four bushing forces have been used to model the exoskeleton body grippers. In all of them, the following values of stiffness and damping have been used: 10000 N/m for the translational stiffness component, 1000 Nm/rad for the rotational one, 0.1 N/(m/s) for the translational damping and 0 Nm/(rad/s) for the rotational damping component.

Some tests have been initially performed to see the effectiveness of this connection method. The same values of the paper have been used for the damping and stiffness parameters of the *BushingForce*. The two point where these forces have been used are those highlighted with a pink dot in the figure 5.1.

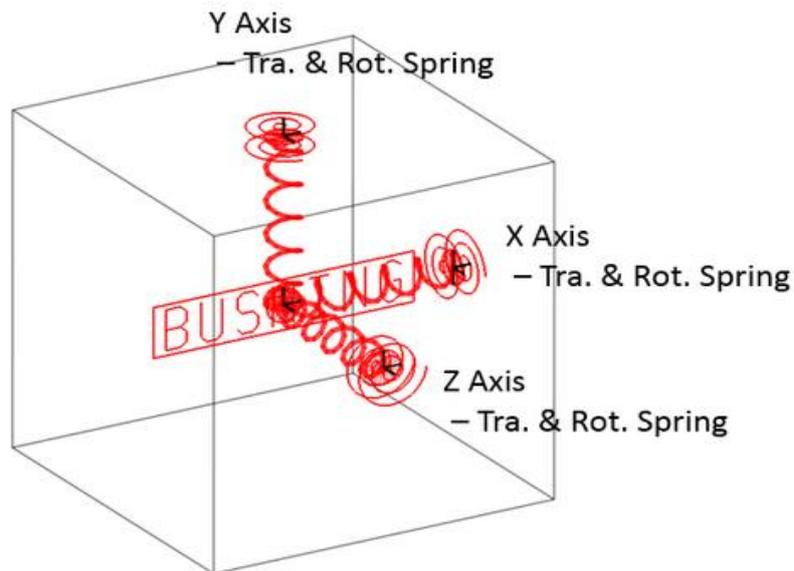


Figure 5.2 Scheme of a bushing force.
Image taken from [45]

5.1 Elbow tests and optimization of *BushingForce* parameters

We have again made the model perform a flexion movement of the human elbow with the exoskeleton passively attached, to compare its behaviour with the model free and the model with one or two links. The elbow moves from the resting position at 0° to 130° in 2 second

In the figure 5.3 it can be seen the results of this simulation, thanks to the force generated by the *BushingForce*, the device follows the human movement well.

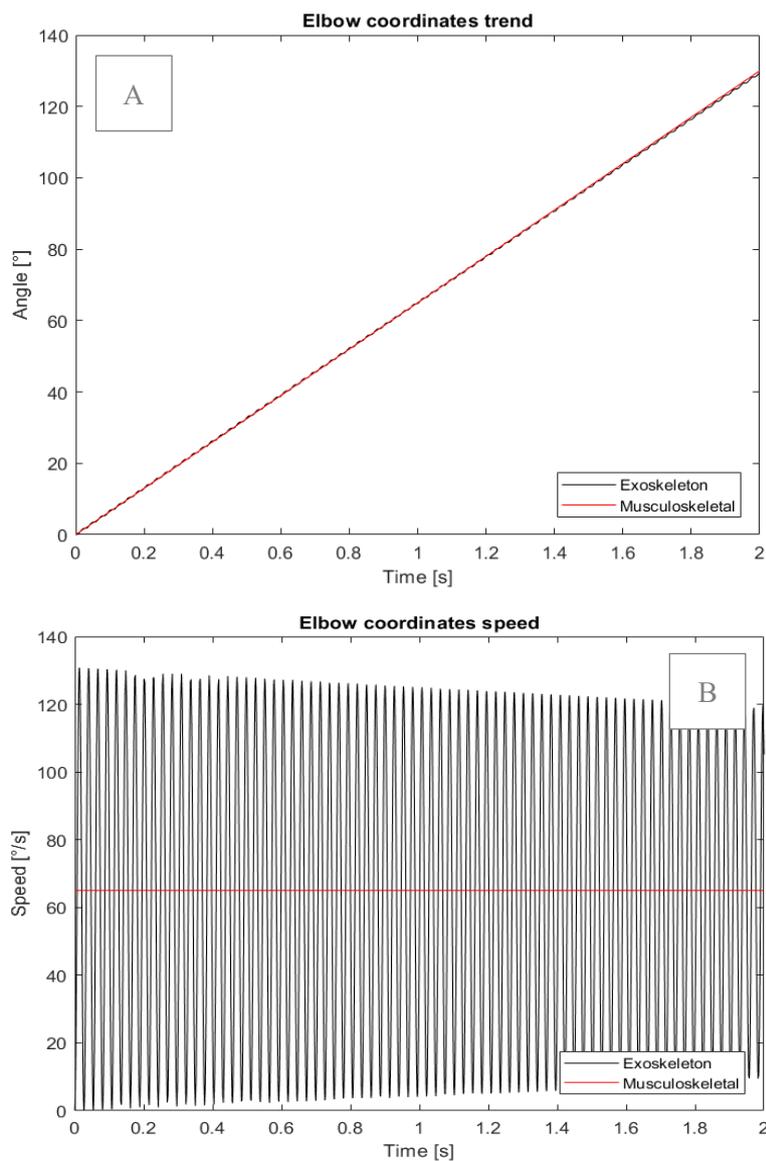


Figure 5.3 A) Elbow coordinates trend
B) Speed.

There is however a particular behavior that can be seen especially in the results of the velocity figure 5.3 B, but it occurs for both the parameters taken in consideration. The exoskeleton had a sinusoidal trend for both. This is due to the *BushingForce*, because there is no perfect correspondence of the points on the two bodies between which the force acts. This problem it's seen also by D. Coll Pujals in their paper.

This is not a real behaviour; however, such a vibration would create problems. Therefore, the best way to eliminate it has been investigated.

This sinusoidal behavior can be observed also in the result of the ID analysis and in force reporter one, as can be seen in the next figure 5.4.

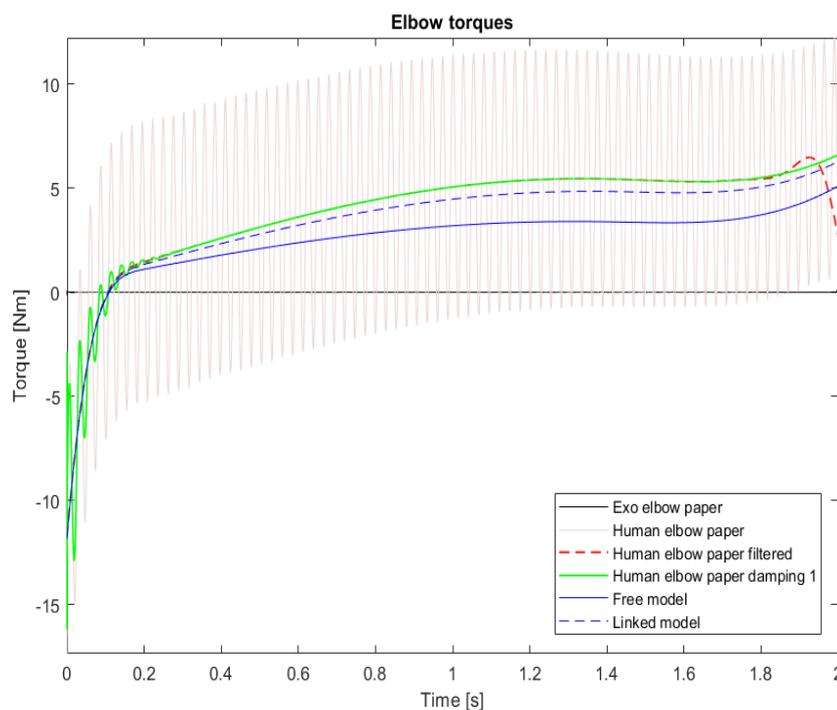


Figure 5.4 Confront between the elbow torques required to complete an elbow flexion. for different models.

In this figure there is the comparison between the values of torques required to complete the elbow flexion from 0 to 130° in 2 s for different models, obtained with an ID analysis. This test was done to see the influence of the exoskeleton on the moment needed at the elbow coordinate to complete the flexion. The comparison between different models and different methods of connection has been made.

The first two lines, of the figure 5.4, the black one and the pink-grey one represents the trend of the moments at the elbow of the exoskeleton and that of the human model, respectively. For this case the components of the *BushingForce* had the values equal to that of the paper. The exoskeleton was taken in consideration only this time because the torque needed in this coordinate is almost always zero since the movement was generated by the human.

The dashed red line was obtained from the ID analysis of the filtered kinematics results from which the pink-grey line was obtained. The filter used was the standard one of OpenSim, a low pass filter with a cut-off frequency of 6 Hz. At the end of the dashed red line, between 1.8 and 2 second, there is a deviation from the trend of the mean value of the not filtered one. Here OpenSim makes the exoskeleton produce a torque equal to the deviation from the average trend. The green line is relative to the torque at the elbow coordinate for the simulation with an increasing of the damping components of the bushing force. Values were set at 1 (N/(m/s) or Nm/(rad/s)) for the translational and rotational components. There is still a sinusoidal pattern in the first part of the movement, but it runs out quickly.

The changing in the damping values did not change the average dynamic behaviour of the model. The requested torque with the increased damping parameters matches the mean value of the predetermined one, coincides with the trend of the filtered analysis. There is also to note that the couple requested to complete the movement, with the exoskeleton coupled (green or dashed red line), is bigger than the one necessary to complete the movement both for the free model and for the one linked.

The expected result was that the torque needed to make the movement was an average between the one relative to the free model and that of the linked model. The reason for this error was searched in the method used to connect the two arms. A *ForceReporter* analyses on the result of the elbow flexion was carried out thanks to the Analyze Tool of OpenSim, to see the values of the forces exchanged by the two forearms through the *BushingForce*.

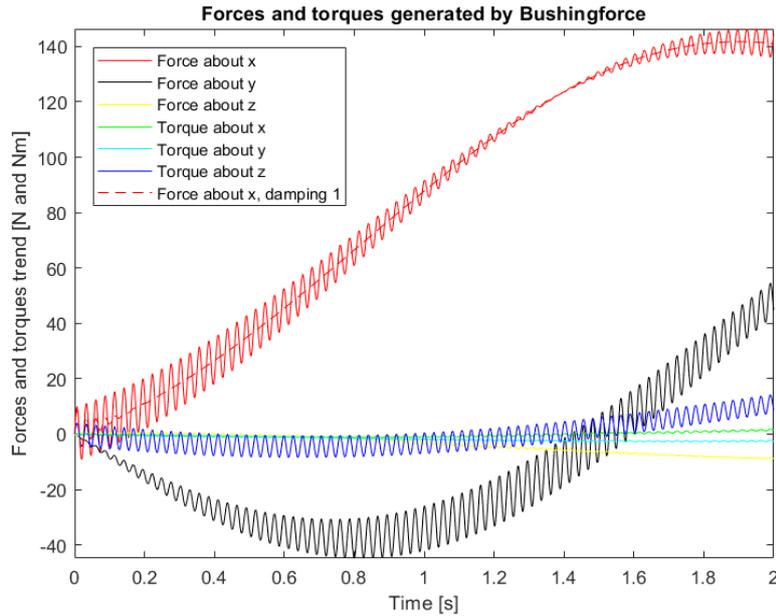


Figure 5.5 Forces and torques generated by the *BushingForce* on the exoskeleton forearm body.

From the figure 5.5 it can be seen the trend of the forces and torques that the *BushingForce* generate on the forearm of the exoskeleton, expressed in the global reference system (figure 4.1). If these forces had been calculated on the human model, they would have the same modulus but opposite sign.

From this figure it can be seen that the forces are greater in the X and Y directions and the torque around Z is the most relevant. The other components have a low absolute value and are more linear. The red dashed line in figure 5.5, it is relative to the model where the damping components were set to 1 (N/(m/s) and Nm/(rad/s)). It follows the mean trend of the X force calculated for the model where the value of the components of the *BushingForce* are the same of the paper [25].

From these tests it results that the force generated by springs of the *BushingForce* are very important in the behaviour of the model. These components tighten up the system. For the movement of the elbow their effect is tolerable because there isn't a great increasing in the torque needed to make the movement. The problem occurs trying to simulate shoulder movement. In this case there is a higher movement between the two arms, due to the kinematical mismatch between the two model, and therefore there is a generation of very high forces.

The main reason to reduce these forces is to make the CMC tool work. The ultimate goal of this thesis is to study how an exoskeleton influences the muscular activation of a subject during a given movement. To calculate this parameter, we must use the CMC tool, which being very

sensitive is highly influenced by such high forces. If they are not reduced, they would lead to incorrect results or to the failure of the simulation

To reduce these forces the value of the spring components of the *BushingForce* were reduced. They were set at 100 N/m and 100 Nm/rad for the translational and rotational components. The damping components were set at 10 N/(m/s) and 10 Nm/(rad/s).

As can be noted in the figure 5.6 the kinematic behavior is good for this configuration. The sinusoidal trend of the exoskeleton has been eliminated. The elbow coordinate of the exoskeleton at the end of the movement remain at 129°, a little bit lower of the human model elbow.

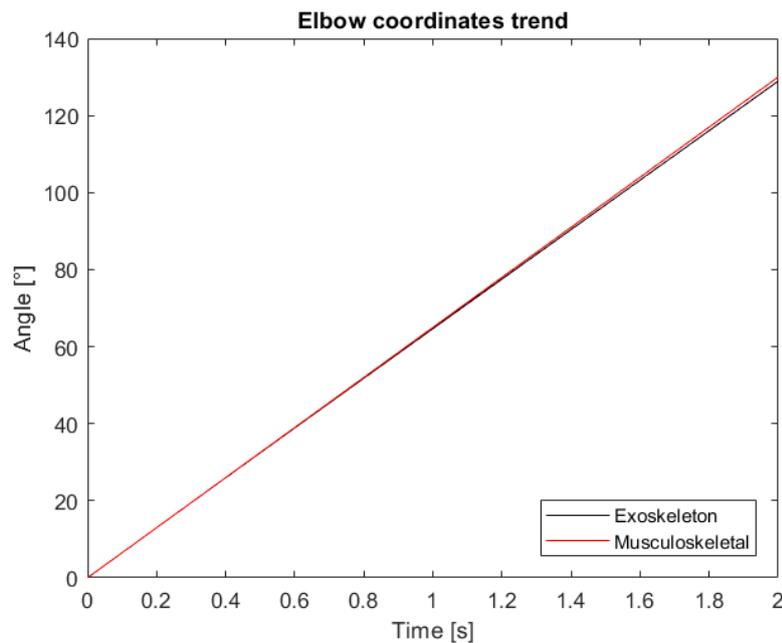


Figure 5.6 Elbow coordinates trend for the model with the new parameters of the *BushingForce*.

The torque needed to complete the movement (red line of the figure 5.7) is now lower than the one needed for the linked model (dashed black), except in the final part of the movement where it remains slightly greater.

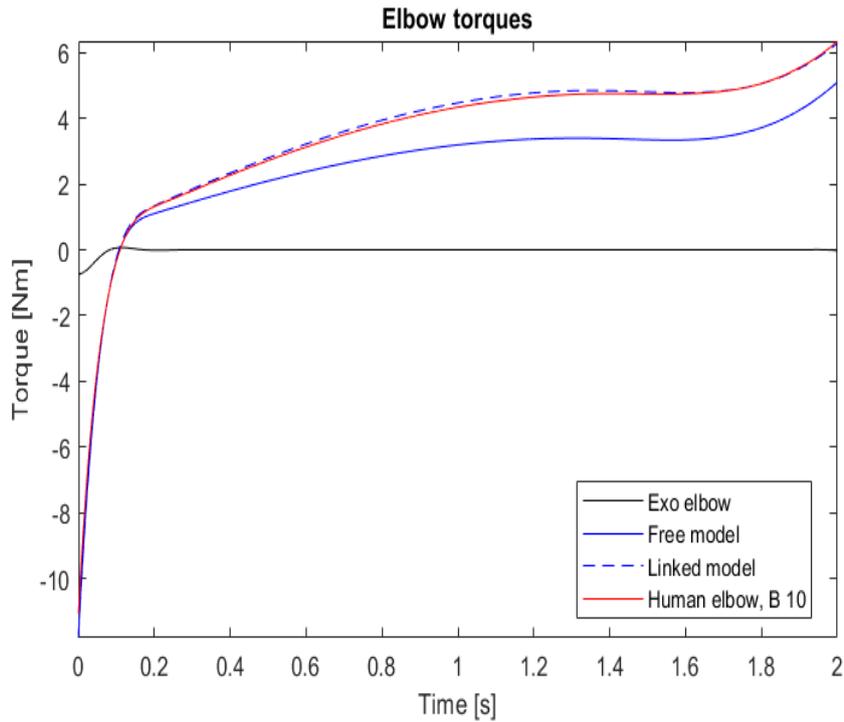


Figure 5.7 Elbow torque for different models.

In the next figure (5.8) we can see the forces and torques generated by the bushing forces on the arm body of the exoskeleton. This chart can be compared with the previous one (fig. 5.5), and we can see that there is a big difference between them.

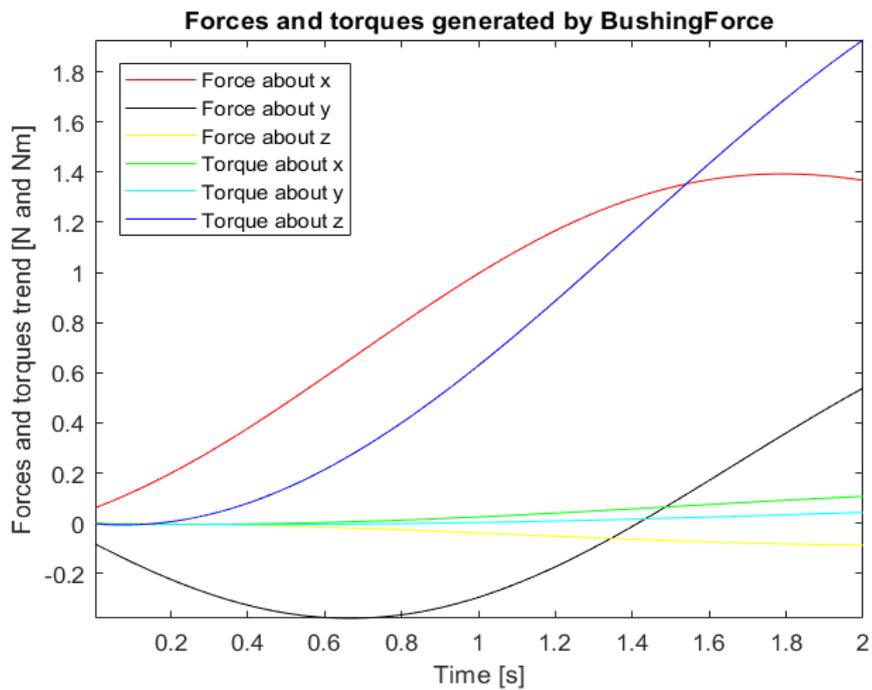


Figure 5.8 Forces and torques generated by the BushingForce, with the new parameters, on the exoskeleton forearm body.

The values of all the forces and torques are decreased considerably, by a factor of 100. From this we can suppose that they are generated mainly by the spring components of the *BushingForce*. We have gone from a translational stiffness of 10000 N/m and a rotational stiffness of 1000 Nm/rad to an absolute value of 100 for both, obtaining a reduction in the forces generated by them of about the same factor. The shape of these forces and torques remained substantially unchanged. The torque around the Z axis has changed its trend with, in proportion, a significant increase in its value. In fact, this moment is the one that is proportionately less reduced.

5.2 Modelling of the exoskeleton shoulder

With this configuration of the value of the bushing force, a shoulder flexion from 0° to 90° in two second has been simulated. The movement take place in the sagittal plane of the model. A function as been imposed at the “shoulder_elv” coordinate of the exoskeleton and the muscles were not active. The “elv_angle” coordinate was locked ad 90° and the “shuolder_rot” at 10°, the others at 0°.

During this type of movement, the position of the center of rotation of the shoulder change a lot during the flexion. It moves of about 2.7 cm backwards along the X direction, of 2.5 cm upwards and 0.9 cm along Z.

To accomplish this movement and prevent the formation of high force that would deviate from a real behavior, the type of joint used to simulate the exoskeleton shoulder was changed. A *PlanarJoint* was used, it provides three ordered mobilities; rotation about the Z axis and translation in X and Y. The translational DOFs of the joint were constrained with rotational one, in this way the position of the center of rotation of the two shoulders had a more similar kinematics. This was done by implementing a *CoordinateCouplerConstraint* between the rotational and the two translational DOF, one constraint for the X axis and one for the Y. The difficulty and importance to align the centers of rotation of the shoulder of human body and exoskeleton is also a problem that occurs in the reality, as reported in the works [9,10,14].

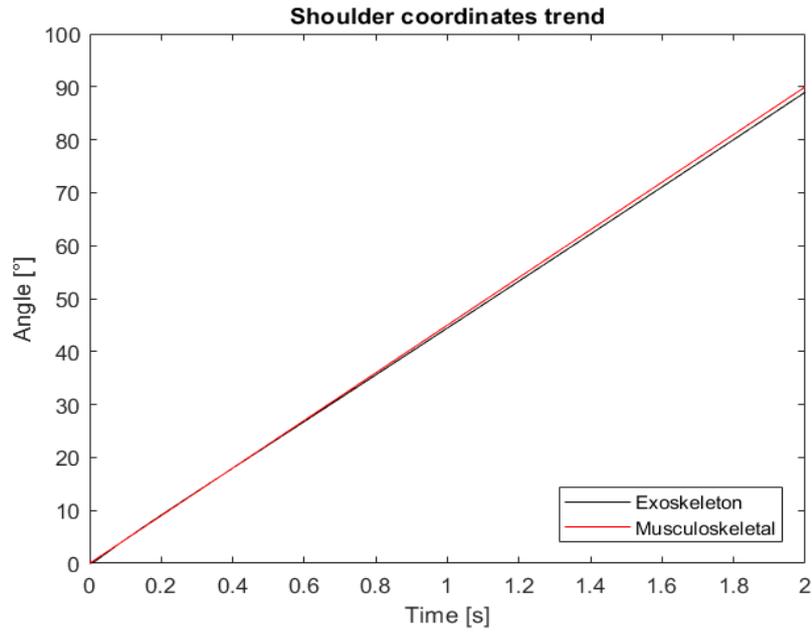


Figure 5.9 Shoulder coordinates trend.

The kinematics results (fig. 5.9) are good, the exoskeleton follow well the movement of the human arm. As for the elbow the exoskeleton doesn't reach the same value of the human model and stop its shoulder when it reaches a final angle of 89°.

The torque at the *shoulder_elv* coordinate of the model (red line fig. 5.10), needed to perform the movement, is correctly a mean between the one needed by the free and the linked model (black line and dashed black line).

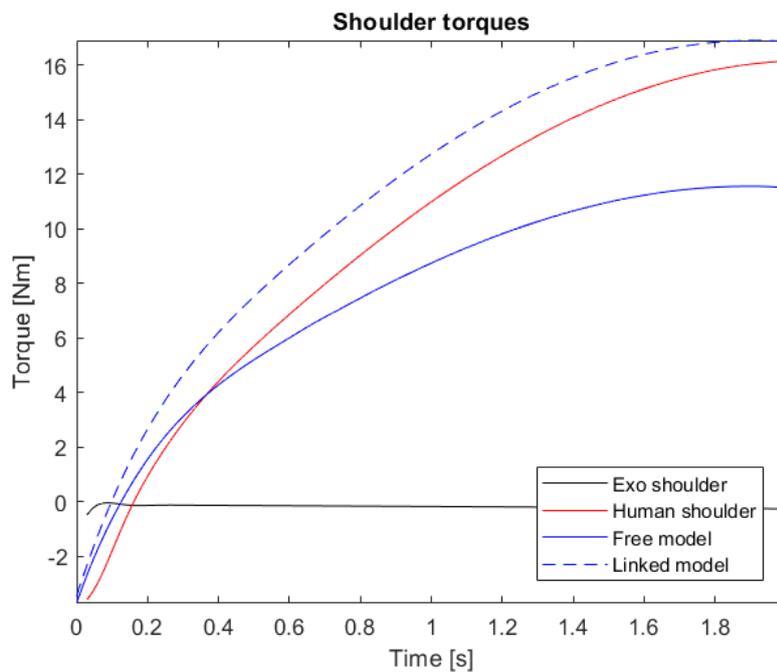


Figure 5.10 Shoulder torques for the different models.

The forces and torques generated by the *BushingForce* have an acceptable value. Both bushing forces create the same value of forces and moments on the two bodies of the exoskeleton arm on which they act respectively. In the fig. 5.11 those that act on the upper arm body of the device have been plotted but could be those on the forearm, indifferently.

As it can be seen in the figure 5.11, the bushing forces generate principally moments on the exoskeletal bodies especially around the X and Y axes. The forces generated are small, less than 5 N.

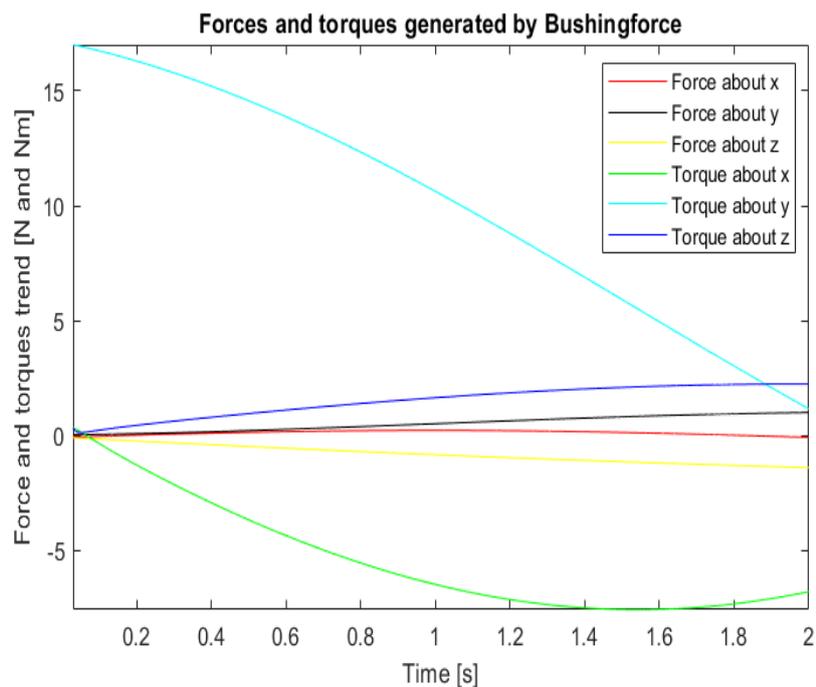


Figure 5.11 Forces and torques generated by the *BushingForce*, with the new parameters, on the exoskeleton upper arm body.

Having allowed the exoskeleton to follow the translational movements of the shoulder rotation center has allowed to obtain excellent results in terms of low force generation. It was also important to change the values of the bushing force used. The figure 5.12 represents the forces and torques generated by the *BushingForce* on the exoskeleton upper arm bodies, they are again the same for both elements, during the usual flexion movement of the shoulder. The values of the components of the *BushingForce* used to obtain the results showed in this figure are the initial ones taken from [25]. As can be seen, although the two further DOF have been introduced, the effect generated remains high.

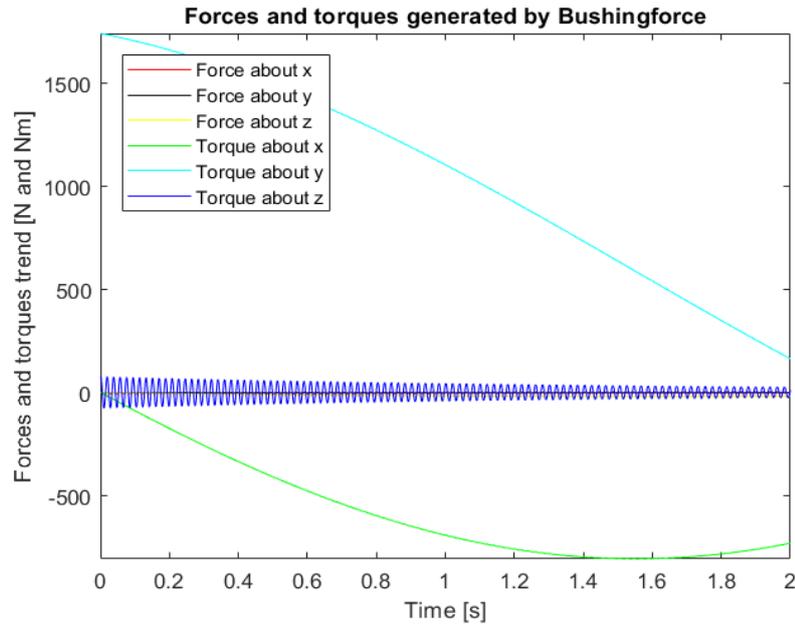


Figure 5.12 Forces and torques generated by the *BushingForce*, with the initial value of the parameters, on the exoskeleton upper arm body.

Now the behavior of the model with a torque of 20 Nm imposed to the human shoulder has been simulated. The movement obtained is a shoulder flexion in the sagittal plane with the exoskeleton that follows the movement passively. The result of this test (fig. 5.13) can be compared to the ones in the figure 4.13.

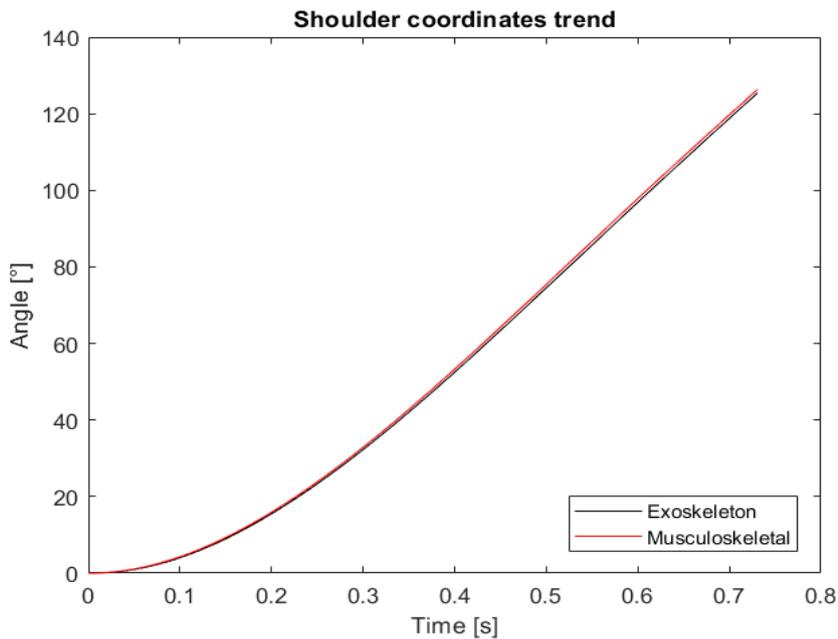


Figure 5.13 Shoulder coordinates trend when torque was applied at the MoBL shoulder.

The result is good, with the exoskeleton that follow well the movement of the human arm. The difference between the two shoulder coordinates is low, it reaches a value of 1° at the end of the movement. As said this result can be compared with the one where the same torque was applied at the shoulder of the model free and linked. As can be noted the speed with which the two arms move, is an average of those obtained for the two models used previously.

5.3 Test on the overall model

The next test performed consist in evaluating the kinematics behavior of the system in a more complicated complete movement. The whole arm moves simultaneously or one coordinate at a time. The movement this time is impressed by the exoskeleton that guides the human arm, which moves passively.

The whole task takes 11 seconds to complete. It starts from the resting position with the arms at the human model side. The first coordinated to move is that relative to the elbow of the exoskeleton, which in 2 seconds reaches 90° and subsequently remains in this position for 1 s. At this point the shoulder of the device begins to move. The final position of this motion is reached after a second, 4 s from the start of the task, and occurs when the shoulder reaches an angle of about 55° while the elbow reach a stretched position, the wrist has run a straight line parallel to the ground.

This configuration lasts 1 second, the 5th second of simulation was reached, and after that it goes back to the previous position, with the elbow at 90 degree, and the shoulder at 0° .

The next position is with each coordinates of the exoskeleton at 90° the shoulder reaches this value in 1 second, the elbow remains steady. From this moment, the eighth second of simulation, until the end of the movement, the shoulder remains still.

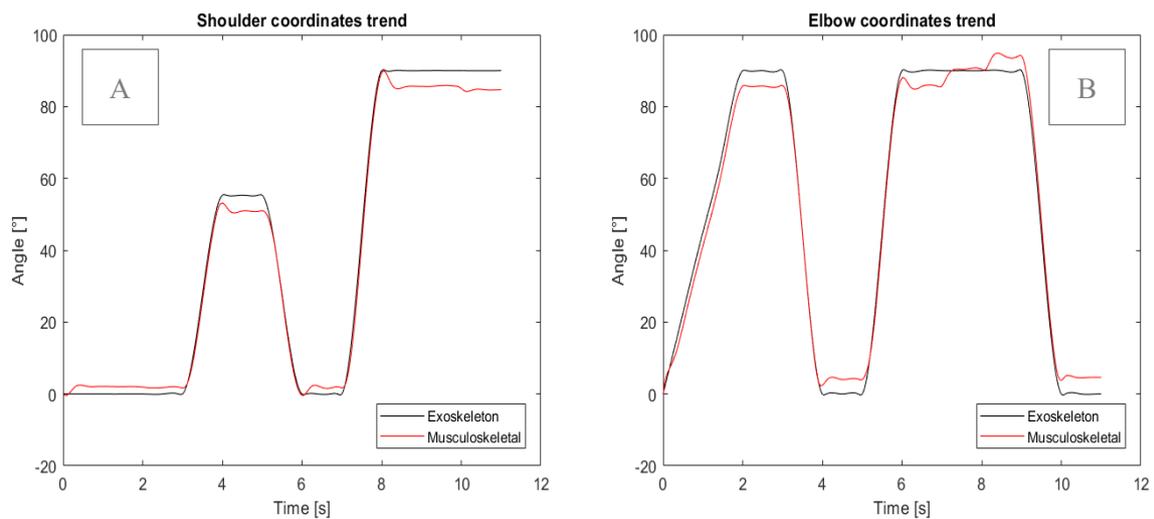
The elbow moves after 1 second to reach the final position with the whole arm extended, with the shoulder at 90 degrees.

The following table (5-1) summarizes better the movements of the arm, indicating at each second the values assumed by the two coordinates concerned.

<i>Summary of the complete movement</i>											
<i>Time (s)</i>	0	2	3	4	5	6	7	8	9	10	11
<i>Shoulder angle (°)</i>	0	0	90	55	55	0	0	90	90	90	90
<i>Elbow angle (°)</i>	0	90	90	0	0	90	90	90	90	0	0

Table 5-1 Summary of the complete movement

The figure 5.14 show the result of this simulation. It allows you to see that this way of connection had a great performance during the movement, but it gets worse in a static position. This is more evident in the finale position when the difference between the shoulder coordinate of the two model is five degree. In addition, it can be seen how the *BushingForce* cannot block the movement of the human shoulder when the one of the exoskeleton is still, and the forearm is in motion.



*Figure 5.14 A) Shoulder coordinates trend during the movement
B) Elbow trend.*

5.4 Analysis on the impact on muscle activation

In previous simulations the muscles were not active and follow the movement passively. The purpose of this thesis is seen if with OpenSim is possible to evaluate the impact of an exoskeleton in the muscle activation.

To carry out a muscle analysis the Compute Muscle Control (CMC) Tool of OpenSim was used. It allows to calculate the control value or excitation of the muscles that allows to reproduce the kinematic behavior of the model given as input.

The tests below are relative to an elbow flexion from 0° to 130° in 2.5 second, with a constant speed. Three simulation were made.

In the first a free musculoskeletal model was used; no external forces are imposed or devices added. The second one is on the human model coupled with a passive exoskeleton. The values of the *BushingForce* components used, are those found from the previous tests: 100 N/m for the translational stiffness component, 100 Nm/rad for the rotational one, 10 N/(m/s) for the translational damping and 10 Nm/(rad/s) for the rotational damping component. It's passive and simply follow the movement of musculoskeletal model.

In the third simulation the exoskeleton is active. It has a *CoordinateActuator* at its elbow that provides a constant torque of 3 Nm after 0.75 s. This transition time has been left not to increase the muscle activation since the couple necessary to make the movement, before that time, was low. Without this delay greater muscle activation to respect the movement would have been required.

The kinematic file to be used as input was obtained from a Forward Dynamics analysis of the two type of models, the actuator of the exoskeleton was added only in the CMC analysis. It was obtained in the usual way, by imposing a function at the elbow coordinate of the human model in order to simulate its movement with the FD tool.

For those simulations a simplified MoBL model was used. The number of muscles were reduced. Those that control the movement of the wrist were eliminated. If we will open the model in OpenSim we would see that all the muscles of the model after the *BRD* have been removed. This was due because we are not interested in the position of the hand, which was left in its default position by locking the wrist coordinates. This also allows us to speed up the simulations given the smaller number of muscles, 24, less than half of the original MoBL model. Simulation times have gone from more than three hours to about one. Moreover, a comparison was made between the muscle activation of the four muscles, which we will analyze once the

exoskeleton is applied, when they are part of the complete model or of our simplified model. It has been noted that the trend of their activations has not changed significantly, during elbow flexion.

The CMC analysis focus on the human elbow joint so it has the highest weight in the task file of 50, while the wrist coordinates were not present since they are locked. In the simulations with the exoskeleton its elbow coordinate had a weight of 30. As already mentioned, the file defines with which level of precision the coordinate will be followed during the simulation, the higher the imposed weight the higher the level of precision.

The reserve actuators were applied to all the coordinate except those locked and at the exoskeleton elbow coordinate, when this device was connected.

Unfortunately, it was not possible to use the algorithm Fast target formulation of the CMC but we had obtained however good results, with errors in the position of the elbow coordinates less than 0.015° and with the reserve actuator of the elbow which provide a maximum of about 0.0150 Nm.

5.4.1 Results

Below the results of these simulations are presented. The first graph, figure 5.15, is relative to the value of the elbow coordinates of the human and of the active exoskeleton, the third simulation. As it can be seen the exoskeleton follow the human movement very well, the kinematics in input has been reproduced well.

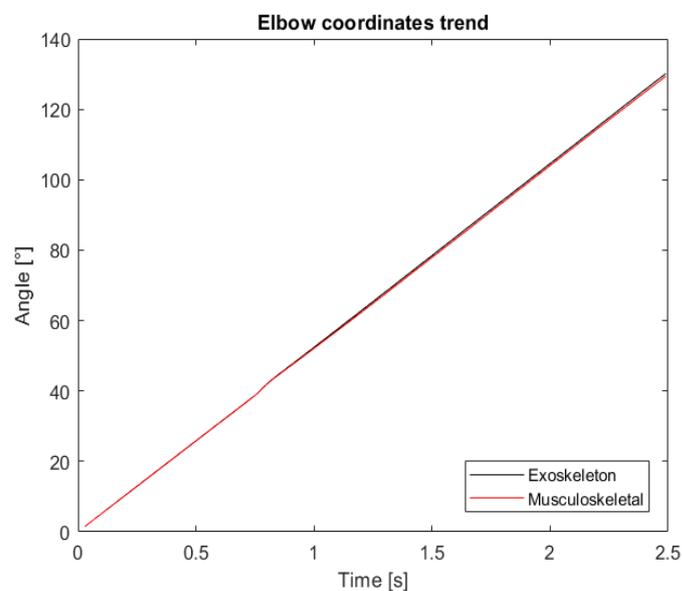


Figure 5.15 Elbow coordinates trend for the simulation with the active exoskeleton.

The only error occurs when the actuator is activated, at this time there is a small divergence in the kinematics trends of the model. One thing that can be noted is that the simulation doesn't start from 0 second but a little later, this happens because the CMC tool takes 0.03 s to initialize.

Now let's analyse the muscular activation resulting from the three simulations.

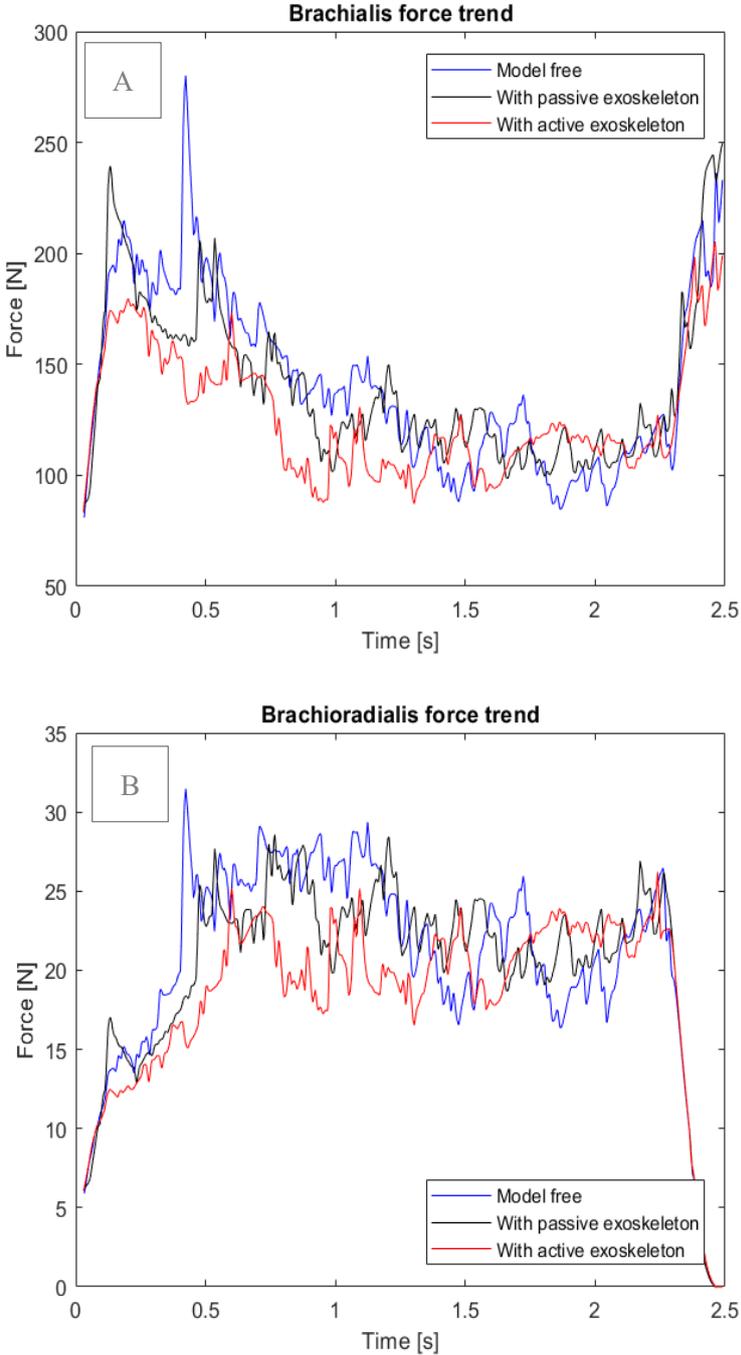


Figure 5.16 A) Brachialis (BRA) force trends
B) Brachioradialis (BRD)

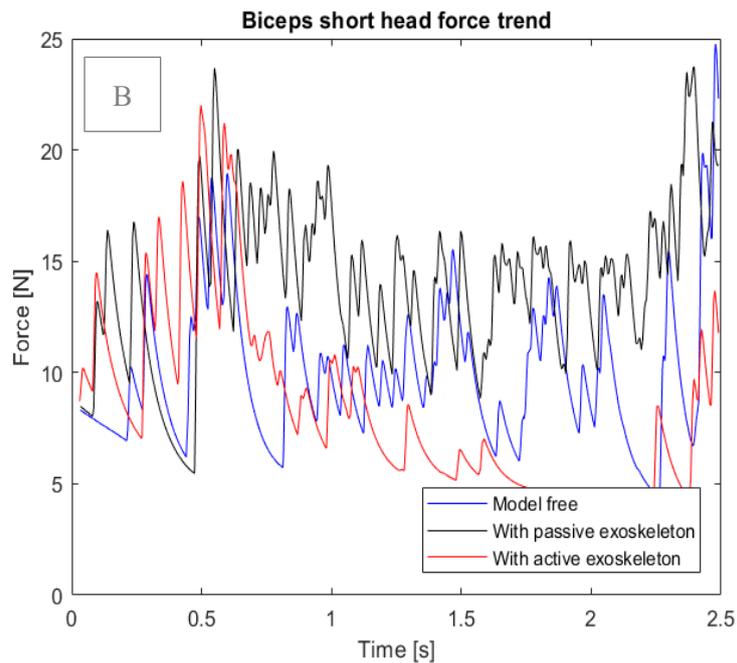
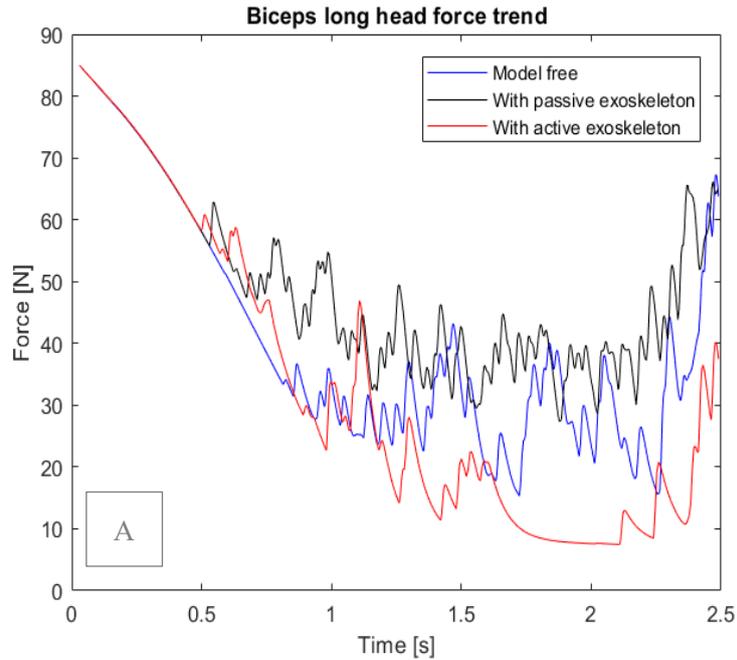


Figure 5.17 A) Biceps long head (BIClong) force trends.
 B) Short head (BICshort) force trends.

From the figures 5.16 and 5.17 it can be seen the forces generated by the muscles Brachialis (BRA), Brachioradialis (BRD) and the two heads of the Biceps. The first thing that can be noted is that the presence of the passive exoskeleton increases the activation of the biceps and decreases that of the other two muscles. That happen especially for the BRD muscle. This difference is not to high due to the low weight supported by the arm. The exoskeleton arm body weight only 1 Kg and is weight is supported by the upper arm of the device.

The reason for this change in the muscle activation has to be found in the fact that the exoskeleton shoulder has been blocked, and therefore stabilizes the arm. If, on the other hand, this joint was free, there would be a more similar muscular activation pattern. With a greater activation of the muscles. Unblocking the exoskeleton shoulder and using the active model of the device, there would be a peak in muscle activation due to the actuator activation. This is due to the fact that OpenSim tries to stabilize the movement of the exoskeleton by activating the muscles of the human arm. In the kinematics input the shoulder coordinate of the device is locked.

In all four figures it can be seen that there is a reduction in muscle activations after the active exoskeleton actuator starts operating. The forces developed by the muscles in this simulation are also lower than those generated when the human model is free.

These results are congruent with expectations, with the exoskeleton that manages to support human movement, providing effective help.

A note can be made on the long head of the biceps (BIClong). The force generated initially in the first 0.8 s of the results related to the free model, is a passive force. The muscle in this time range isn't active, the activation wasn't plotted. As it can be seen in the figure 5.17A the presence of the exoskeleton anticipates the activation of this muscle compared to the free model, there are peaks of force before of 0.8 s.

Now let's analyze the reaction force at the elbow of the human model in the three simulations. This analysis was performed with the Join Reaction analysis of the Analyze Tool of OpenSim by importing the control and kinematic files obtained as results of the CMC Tool.

The Join Reaction analysis calculate the reaction load as the forces and moments required to constrain the body motions to satisfy the joint as if the joint did not exist. The reaction load acts at the joint centre (mobilizer frame) of both the parent and child bodies and either force can be reported and expressed in either the child, parent or ground frames. [46]

In this case the reaction moment is returned as it were applied at the origin of the distal body, the ulna body of the model, and expressed in the Ground frame.

The results of these simulations are very different depending on the direction taken into consideration.

On the X axis the trend of the forces for the three models are very similar, the presence of the exoskeleton raises or lowers the resulting force depending on the joint angle. On the Y axis the forces are similar in the initial part of the movement. They start to differ, curiously, from the

moment of activation of the actuator of the third model. From this time for the model with the passive exoskeleton it can be seen a reduction in the force acting on this axis. From both the graphs relative to the X and Y axes it can be seen the activation of the actuator of the exoskeleton.

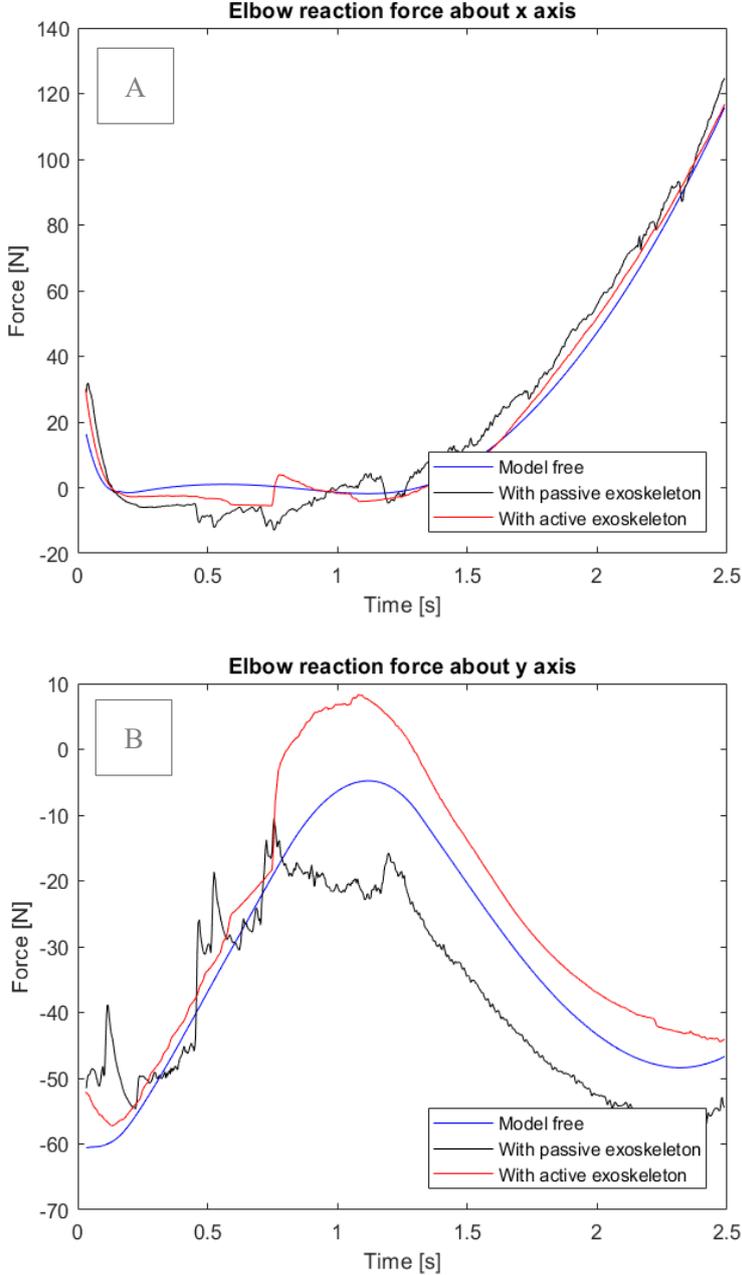


Figure 5.18 A) Joint reaction forces on X.
B) On Y axis.

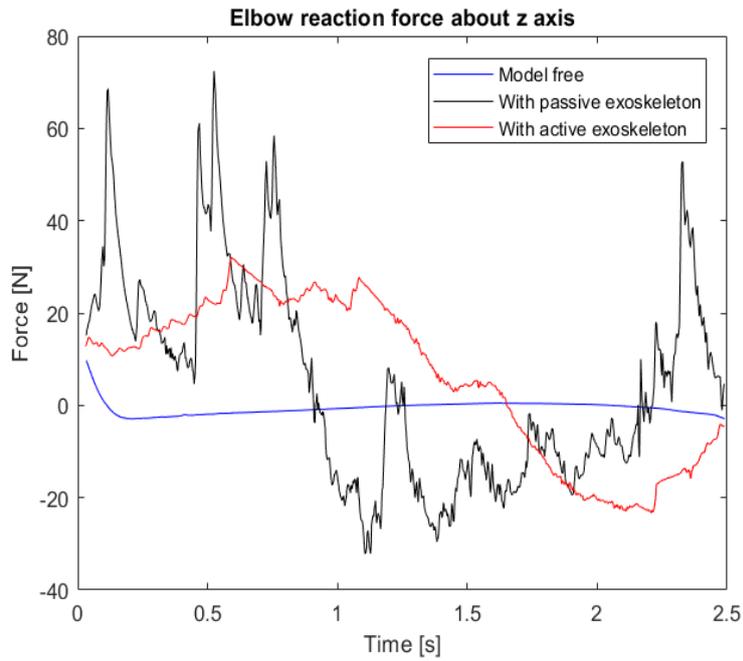


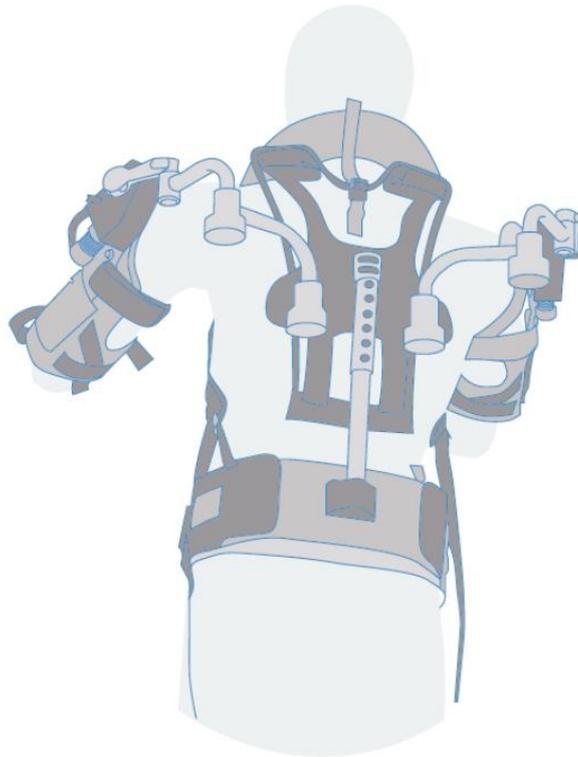
Figure 5.19 Joint reaction forces on Z axis.

The force on the Y axis is the one where the peak of force generated by the actuator is most visible. The presence of the actuator generates a greater force than in the free model, when the presence of the exoskeleton lowers it.

The last figure (5.19), relative to the force on the Z axis, show a high difference in the results of the three simulations. The free model has almost no force generated along this axis, which instead is generated by the interaction of the exoskeleton.

6 Comparison with real case and results

Now a more complex movement was taken in consideration, the drilling task. The next part of this studies is based on the one described in the paper [6], conduct by Alabdulkarim et al..



*Figure 6.1 EksoVest
Image taken from [47]*

6.1 Real experiment

In this work the authors have tested three type of exoskeletons to see how these devices affect man during the performance of a task. The evaluation has been conducted both on movement accuracy and in terms of aid to the operator, reducing of the physical demand.

The motion that the subjects participating in the experiment had to perform was the drilling task. This type of movement has been considered because it's very common and working with arm elevation is a well-documented contributor to shoulder injuries [6].

The task performed and studied takes in consideration a drilling tool that had a mass of about 2 Kg and that moves at eye height of the participant at the study. The subjects of the experiment had to generate a drilling force of at least 66.7 N which had to be sustained for a total duration of 2.5 s.

In their study Alabdulkarim et al. have compared the behavior of three passive exoskeletons. Two of them were attached directly to the drill that transmits the loads to the hip or to the floor. The third exoskeleton used was the EksoWorks™ (v.3, Ekso Bionics Company, Richmond, CA, USA), an upper body anthropomorphic vest that supports the arms [6], that is the one regard to which the results of our simulations will be compared.

The exoskeleton can be seen in the figure 6.1. From the exoskeleton manufacturer page [48], we can see the device specifications.

It is called EksoVest and it is a passive exoskeleton. It has a spring system that act on the shoulder to support the arm movement. It can support both the arms.

Its weight is unloaded on the hip of the subject who wears it, the device has a belt. The upper arm is supported thanks to a connection band that transfers the force generated by the passive actuator positioned at shoulder height. It does not provide support to the elbow, at the forearm there is a sleeve connected with the cuffs for the upper arm through two straps. This compression sleeve has a security function, and the producers recommend wearing them. The support obtainable from springs system of the actuator is adjustable in a range from 2.2 kg to 6.8 kg.

6.2 Setting up our simulations

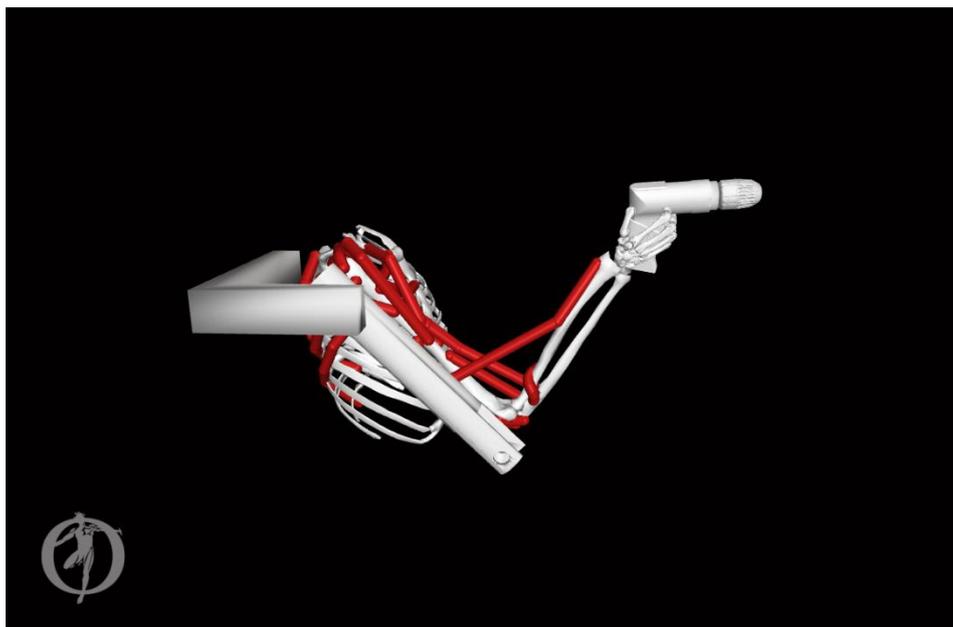


Figure 6.2 The model used for the simulation.

In the following it is described the simulation done to reproduce the configuration of the experiment reported in the paper [6].

There were some limitations, the first is that there are no precise kinematic data in the paper. The starting position was inspired by the one of the figure 1.e of the paper. One difference is that the movement is perfectly in the sagittal plane, the “elv_angle” coordinate was locked at 90° and the “shoulder_rot” at 10°.

The shoulder flexion coordinates of the human model and of the exoskeleton were set at 45° and the elbow coordinate in the initial position of 96°. Subsequently the arm extends frontally by advancing the drill bit of 5 cm in a straight line. To do that the shoulder reach an angle of 91.5° and the elbow of 53°.

The movement take 2.5 s to be completed and for the entire duration a force, acting on the drill bit, of 66.7 N was applied. It was expressed on the X axis of the global reference system to contrast the movement, negative direction.

Two simulations were carried out one with the human model free and one with an exoskeleton model that reproduce the EksoVest.

A body with the geometry and the mass proprieties of the drill was connected, with a weld joint, to the hand of the MoBL model. The CAD model was obtained from the site grabcad.com, it was designed by Clément Chappat [49]. The mass proprieties are been evaluated with SolidWorks and then imported in OpenSim, as for the bodies of the exoskeleton.

The movement was initially simulated with the Forward Dynamics Tool to obtain the kinematic file to use in the next tools. After that an ID analysis was carried to evaluate torque at the shoulder, needed to complete the movement. The last tool used was the CMC one to analyze the muscle activation.

```

<?xml version="1.0" encoding="UTF-8" ?>
<OpenSimDocument Version="40000">
  <ExternalLoads name="externalloads">
    <objects>
      <ExternalForce name="externalforce">
        <!--Name of the body the force is applied to.-->
        <applied_to_body>Drill</applied_to_body>
        <!--Name of the body the force is expressed in (default is ground).-->
        <force_expressed_in_body>ground</force_expressed_in_body>
        <!--Name of the body the point is expressed in (default is ground).-->
        <point_expressed_in_body>Drill</point_expressed_in_body>
        <!--Identifier (string) to locate the force to be applied in the data source.-->
        <force_identifier>Drill_force_v</force_identifier>
        <!--Identifier (string) to locate the point to be applied in the data source.-->
        <point_identifier>Drill_force_p</point_identifier>
        <!--Identifier (string) to locate the torque to be applied in the data source.-->
        <torque_identifier></torque_identifier>
        <!--Name of the data source (Storage) that will supply the force data.-->
        <data_source_name>Unassigned</data_source_name>
      </ExternalForce>
    </objects>
    <groups />
    <!--Storage file (.sto) containing (3) components of force and/or torque and point of application. Note
    <datafile>C:\Users\Matteo\Desktop\drill_f.mot</datafile>
    <!--The option is deprecated and unnecessary to apply external loads. A motion file (.mot) or storage
    <external_loads_model_kinematics_file />
    <!--Optional low-pass cut-off frequency for filtering the model kinematics corresponding used to trans
    <lowpass_cutoff_frequency_for_load_kinematics>-1</lowpass_cutoff_frequency_for_load_kinematics>
  </ExternalLoads>
</OpenSimDocument>

```

Figure 6.3 The .xml file for the external force.

The ID and the CMC analysis have been carried out giving the resistive force. To add an external force acting on the model two files must be defined. They define all the parameters of this force: modulus, direction and the point on which it acts.

The first file is in format .xml and its content can be seen in the figure 6.3. This file is used to define on which body the force act, which reference system is used to define the position of the point on which it acts and its direction, as can be seen they may be different. Furthermore, it gives to OpenSim the identifier to read the motion (.mot) file that contain the data of the force and the address of this .mot file.

The motion file (figure 6.4) have the structure described in the previous paragraph. It has a header and a data table.

The first column of the table is the time, the next three indicate the components and the direction of the force along the three axes of the chosen reference system. In our case it will be on the X axis, in the negative direction as it is opposed to the motion.

```
drill_f.mot
version=1
nRows=26
nColumns=7
endheader
time Drill_force_vx Drill_force_vy Drill_force_vz Drill_force_px Drill_force_py Drill_force_pz
0 -66.7 0 0 0.0652408 -0.0373104 0.00982643
0.1 -66.7 0 0 0.0652408 -0.0373105 0.00982643
0.2 -66.7 0 0 0.0652408 -0.0373106 0.00982643
0.3 -66.7 0 0 0.0652408 -0.0373107 0.00982643
0.4 -66.7 0 0 0.0652408 -0.0373108 0.00982643
0.5 -66.7 0 0 0.0652408 -0.0373109 0.00982643
0.6 -66.7 0 0 0.0652408 -0.0373110 0.00982643
0.7 -66.7 0 0 0.0652408 -0.0373111 0.00982643
0.8 -66.7 0 0 0.0652408 -0.0373112 0.00982643
0.9 -66.7 0 0 0.0652408 -0.0373113 0.00982643
1 -66.7 0 0 0.0652408 -0.0373114 0.00982643
1.1 -66.7 0 0 0.0652408 -0.0373115 0.00982643
1.2 -66.7 0 0 0.0652408 -0.0373116 0.00982643
1.3 -66.7 0 0 0.0652408 -0.0373117 0.00982643
1.4 -66.7 0 0 0.0652408 -0.0373118 0.00982643
1.5 -66.7 0 0 0.0652408 -0.0373119 0.00982643
1.6 -66.7 0 0 0.0652408 -0.0373120 0.00982643
1.7 -66.7 0 0 0.0652408 -0.0373121 0.00982643
1.8 -66.7 0 0 0.0652408 -0.0373122 0.00982643
1.9 -66.7 0 0 0.0652408 -0.0373123 0.00982643
2 -66.7 0 0 0.0652408 -0.0373124 0.00982643
2.1 -66.7 0 0 0.0652408 -0.0373125 0.00982643
2.2 -66.7 0 0 0.0652408 -0.0373126 0.00982643
2.3 -66.7 0 0 0.0652408 -0.0373127 0.00982643
2.4 -66.7 0 0 0.0652408 -0.0373128 0.00982643
2.5 -66.7 0 0 0.0652408 -0.0373128 0.00982643
```

Figure 6.4 The motion file of the external force.

The last 3 columns indicate the point of application of the force on the drill expressed in the reference system of this last one. If we had expressed it on another reference system obviously the position of the point would not be constant. The point is the drill bit.

The other files used for the CMC tool, in addition to the already mentioned and the model file, are:

- the task file, which defines the weights with which the coordinates are followed;

- the actuator constrain file, that defines the shoulder actuator control value.
- the file containing the reserve actuator;

6.3 Results

In this paragraph we present the results obtained from the two simulations carried out, and we will compare them with the real ones taken from the reference article.

First, the shoulder torque of the free model needed to perform the movement was evaluated. It was found to vary at 10 to 12.5 Nm, at the end of the task.

The paper doesn't specify the level of support that the exoskeleton provide, the participants to the study could choose the level of support they thought best suited for the task. From the operator manual of the EksoVest it can be seen that it can provide four ranges of support: between 2.2 and 3.1 Kg, 3.1-4 Kg, 4-5.4 Kg and between 5.4-6.8 Kg. Once you choose the range, the level of support provided by the spring actuator of the exoskeleton depends on shoulder angle. For our actuator we therefore chose a value of torque of 11 Nm.

The results below, figures 6.5 and 6.6, show that each of the heads of the muscle deltoid reduce the force supplied thanks to the presence of the exoskeleton. The lateral deltoid is the most affected one by the presence of the device.

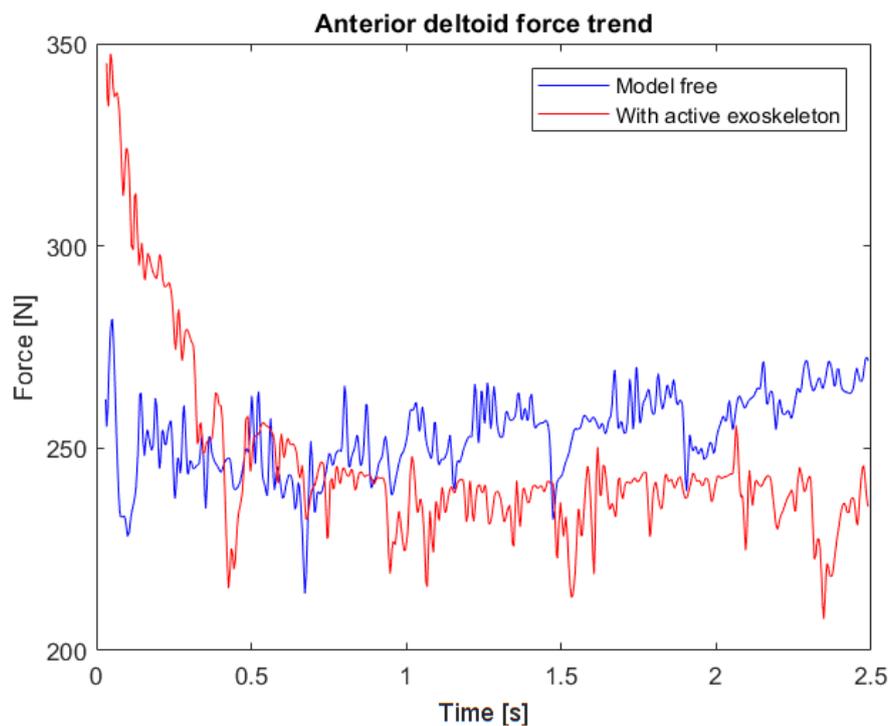


Figure 6.5 Anterior deltoid (DELTA1) force trend.

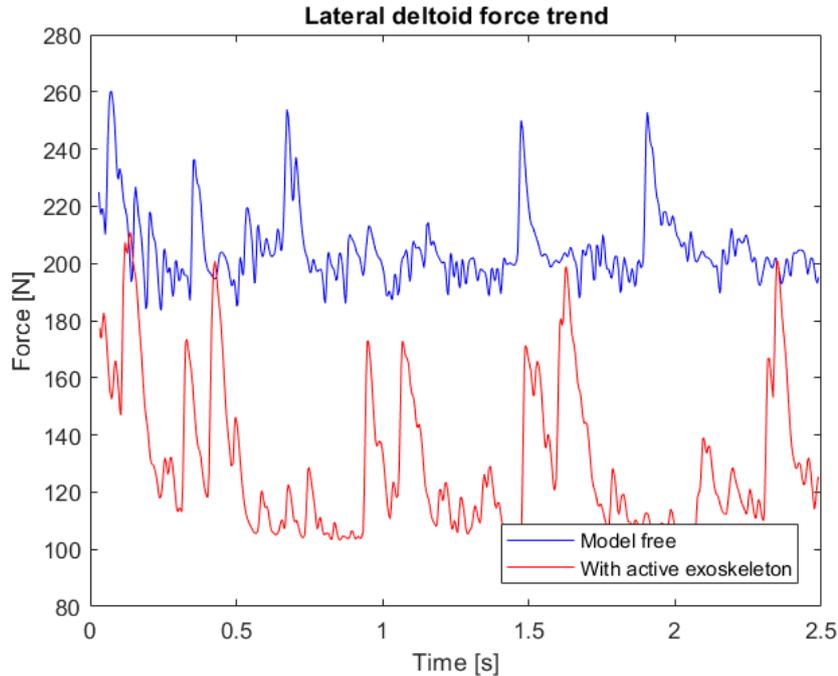


Figure 6.6 Lateral deltoid (DEL2) force trend.

The initial peaks mainly visible in the force of the DELT1 muscle, for the simulation with the exoskeleton (red line of figure 6.5), is probably due to the fact that the actuator was not present when the kinematic data is created, but only inserted in the CMC tool. This disturbs the initial kinematics.

The results therefore confirm expected ones and are congruent with those reported in the article. There is a reduction in deltoid activation, as also reported by the paper. However, as we will see, the levels of muscle activation we have calculated are higher than those found during the real experiment.

To have an objective evaluation of the exoskeleton support in the article they analyze muscle activation. First, they evaluated the maximum voluntary contraction (MVC) of the three analyzed muscles. After that, during the performing of the task, they took surface EMG signal. It was filtered, and then the EMG root mean square (RMS), with a time constant of 100 ms, was determined. This RMS value was then normalized, for the RMS value obtained from the MVC trials, to obtain the nEMG (normalized EMG). To calculate the static, median, and peak activation levels of the muscle, the 10th, 50th and 90th percentile of the nEMG were used.

The data obtained by them are quite different from ours. The medium muscle activation for the anterior deltoid muscle, the only muscle in our model that they also used to evaluate exoskeletons, is the 25% of the max level (1) for our free model and 23% for the model with

the exoskeleton. The reported values are very different, with a mean activation of 6% for the subject free and 4% of the MVC for the subject supported by the exoskeleton.

7 Conclusion

The work done shows that the *BushingForce* method can be used to connect an exoskeleton to a musculoskeletal model in the OpenSim environment, with good results. There are some problems that need to be solved and additional implementations that should be done.

The available musculoskeletal model should be improved especially with regards to the muscular field looking at the errors that have been found in the path of some muscles.

The method with which the two model parts are connected is perhaps not the best available, but it approximates the real behavior well and makes it possible to simplify the exoskeleton positioning problem when the model is very complicated, as in our case.

In the cited works where constraints were used as a connection method, the musculoskeletal model was less complex and, above all, the single real joint was modelled by a single joint. OpenSim was able to manage the constraints and calculate the correct angular values of each coordinates. OpenSim modifies the latter to meet the constraints imposed, which can lead to non-real coordinate values both in the initial position and during the movement.

However, the problem of correct positioning and alignment of the two components of the model remains present. The two parts must be placed in the correct initial position, and the points of application of the *BushingForce* on them must be correctly aligned. Otherwise we would encounter the generation of a wrong dynamics and kinematics due to the greater elongation of the springs.

The chosen stiffness value is sufficient to couple the movement of the two arms of the system, without generating too big forces that would go against the motion. The damping component is also important to eliminate sinusoidal behaviour without generating significant forces

The order of magnitude of the stiffness used is also the same as that calculated to model the man-exoskeleton interaction by André Schiele in the work 10, the value found in this paper is between 200 N/m and 300 N/m.

The final values of the *BushingForce* components are: 100 N/m for the translational stiffness component, 100 Nm/rad for the rotational one, 10 N/(m/s) for the translational damping and 10 Nm/(rad/s).

The tests on the muscle activation shows how, with the use of an exoskeleton, we can effectively support human movement. Above all you can see how OpenSim is able to simulate its operation and therefore can be used to simulate and develop this type of devices.

8 References

- [1] <https://www.ge.com/reports/post/78574114995/the-story-behind-the-real-iron-man-suit/>
- [2] exoskeletonreport.com
- [3] <https://www.digitaltrends.com/cool-tech/robot-exosuits-ohio-state-experiment/>
- [4] A. Voilqu'e, J. Masood, JC. Fauroux1, L. Sabourin, O. Guezet, Industrial Exoskeleton Technology: Classification, Structural Analysis, and Structural Complexity Indicator, Wearable Robotics Association Conference (WearRAcon), 2019
- [5] <https://ohsonline.com/Articles/2018/10/01/Industrial-Exoskeletons-What-Youre-Not-Hearing.aspx?Page=1>
- [6] S. Alabdulkarim, S. Kim, M. A. Nussbaum, Effects of exoskeleton design and precision requirements on physical demands and quality in a simulated overhead drilling task, Applied Ergonomics, Volume 80, pp. 136-145, 2019
- [7] S. Kim, M. A. Nussbaum, M. I. M. Esfahani et al., Assessing the influence of a passive, upper extremity exoskeletal vest for tasks requiring arm elevation: Part I – “Expected” effects on discomfort, shoulder muscle activity, and work task performance, Applied Ergonomics 70, pp. 315–322, 2018
- [8] https://en.wikipedia.org/wiki/Powered_exoskeleton
- [9] M. B. Näf, K. Junius, M. Rossini et al., Misalignment Compensation for Full Human-Exoskeleton Kinematic Compatibility: State of the Art and Evaluation, Applied Mechanics Reviews 70.5, pp 1-19, 2019
- [10] A. Schiele, Fundamentals of Ergonomic Exoskeleton Robots, Ph.D.thesis, Delft, The Netherlands. 2008,
- [11] G. Colombo, M. Jorg, R. Schreier, V. Dietz, Treadmill Training of Paraplegic Patients Using a Robotic Orthosis, Journal of Rehabilitation Research and Development 37.6, pp. 693–700, 2000
- [12] G. Colombo, M. Jorg, R. Schreier, V. Dietz, Driven Gait Orthosis to Do Locomotor Training of Paraplegic Patients, Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3159–3163, 2000
- [13] <https://radiologykey.com/upper-limb-and-shoulder-girdle/>
- [14] M. A. Ergin, V. Patoglu, ASSISTON-SE: A self-aligning shoulder-elbow exoskeleton, IEEE International Conference on Robotics and Automation, pp.2479-2485, 2012
- [15] J. Lenarcic, M. Stanisic, A humanoid shoulder complex and the humeral pointing kinematics, IEEE Transactions on Robotics and Automation 19.3, pp. 499 – 506, 2003
- [16] www.plm.automation.siemens.com/global/en/products/tecnomatix
- [17] www.mathworks.com/products/simmechanics.html
- [18] www.opensim.stanford.edu

- [19] Seth A., Hicks JL., Uchida TK., Habib A., Dembia CL., Dunne JJ. et al., OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement, PLoS Comput Biol 14(7): e1006223, 2018
- [20] www.anybodytech.com
- [21] P. Agarwal, M.S. Narayanan, Leng-Feng Lee, F. Mendel, V.N. Krovi, Simulation-Based Design of Exoskeletons Using Musculoskeletal Analysis, Proceedings of the ASME Design Engineering Technical Conference, 2010
- [22] Zhou Lelai, Li Yibin, Bai Shaoping, A human-centered design optimization approach for robotic exoskeletons through biomechanical simulation, Robotics and Autonomous Systems, Vol.91, pp.337-347, 2017
- [23] F. Ferrati, R. Bortoletto, E. Pagello, Virtual Modelling of a Real Exoskeleton Constrained to a Human Musculoskeletal Model, Biomimetic and Biohybrid Systems, pp 114-125, 2013
- [24] E. J. Stetz, Simulation of Squat Exercise Effectiveness Utilizing a Passive Resistive Exoskeleton in Zero Gravity, Cincinnati; 2016.
- [25] D. Coll Pujals, Simulation of the Assistance of an Exoskeleton on Lower Limbs Joints Using OpenSim, Barcelona, 2017
- [26] Z. Shen, S. Sam, G. Allison, L. Cui, A Simulation-Based Study on a Clutch-Spring Mechanism Reducing Human Walking Metabolic Cost. International Journal of Mechanical Engineering and Robotics Research, Vol. 7, No. 1, pp. 55-60, January 2018.
- [27] Torricelli D., Cortés C., Lete N., Bertelsen Á, Gonzalez-Vargas JE., del-Ama AJ., Dimbwadyo I., Moreno JC., Florez J. and Pons JL., A Subject-Specific Kinematic Model to Predict Human Motion in Exoskeleton-Assisted Gait. Front. Neurobot. 12:18; doi: 10.3389/fnbot.2018.00018, 2018
- [28] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/Simulation-Based+Design+to+Prevent+Ankle+Injuries>
- [29] E. Arch, S. Stanhope, J. Higginson, Passive-dynamic ankle-foot orthosis replicates soleus but not gastrocnemius muscle function during stance in gait: Insights for orthosis prescription, Prosthetics and orthotics international 40, 2015
- [30] P. Agarwal, Pei-Hsin Kuo, R.R. Neptune, A.D. Deshpande, A Novel Framework for Virtual Prototyping of Rehabilitation Exoskeletons. IEEE International Conference on Rehabilitation Robotics, Seattle, Washington USA, 2013
- [31] Bas J. de Kruif, E. Schmidhauser, K.S. Stadler, L. W. O'Sullivan, Simulation Architecture for Modelling Interaction Between User and Elbow-articulated Exoskeleton. Journal of Bionic Engineering 14, pp 706–715, 2017
- [32] F. Anderson, S. Delp, M. DeMers et al., OpenSim User's Guide, Release 2.4, 2012
- [33] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/Getting+Started+with+RRA>

- [34] D. G. Thelen, F.C. Anderson, Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of Biomechanics* 39, pp. 1107–1115, 2006
- [35] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/How+CMC+Works>
- [36] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/Introduction+to+the+OpenSim+API>
- [37] https://simtk.org/api_docs/opensim/api_docs/md_doc_APIGuide.html
- [38] K. Holzbaaur, W. M. Murray, S. L. Delp, A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Contro; *Annals of Biomedical Engineering* 33, pp. 829-40, 2005
- [39] KR. Saul, X. Hu, CM. Goehler, M. Daly, ME. Vidt, A. Velisar, WM. Murray, Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering*, 2015
- [40] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/Gait+2392+and+2354+Models>
- [41] https://simtk.org/api_docs/opensim/api_docs/classOpenSim_1_1EllipsoidJoint.html#details
- [42] D.G. Thelen, Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *ASME Journal of Biomechanical Engineering* 125(1), pp. 70–77, 2003
- [43] Tzong-Ming WU, Dar-Zen CHEN. Design and preliminary evaluation of an exoskeleton for upper limb resistance training. Higher Education Press and Springer-Verlag Berlin Heidelberg 2012
- [44] https://simtk.org/api_docs/opensim/api_docs/classOpenSim_1_1BushingForce.html#details
- [45] <https://support.functionbay.com/en/technical-tip/single/91/types-recurdyn-forces>
- [46] <https://simtk-confluence.stanford.edu:8443/display/OpenSim/Joint+Reactions+Analysis#JointReactionsAnalysis-Output>
- [47] <https://www.design-engineering.com/ford-largest-adoption-exoskeleton-1004031111/eksoworks-ekso-vest-linear/>
- [48] <https://eksobionics.com/it/eksoworks/eksovest/>
- [49] <https://grabcad.com/library/3-8-heavy-duty-drill-speed-2-200-rpm>

Acknowledgements

In this last part I would like to thank my parents for having given me the chance to do this academic experience.

My uncle for helping me several times during these years of study and above all for having reviewed and corrected the English language mistakes in this thesis.

Finally, my sister because if I don't mention her, she will get angry.