27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0120
SOLVING OPTIMAL CONTROL PROBLEMS APPLIED TO NEUROMUSCULOSKELETAL MODELS: A COMPARISON BETWEEN OPENSIM MOCO AND FMINCON SOLVERS

Denis Mosconi
Adriano Siqueira
University of São Paulo, Mechanical Engineering Department, São Carlos, Brazil
Av. Trabalhador São-Carlense, 400, São Carlos-SP, 13566-590, Brazil
denis.mosconi@ifsp.edu.br, siqueira@sc.usp.br

Abstract. *Optimal control problems applied to neuromusculoskeletal models are useful to allow predictive simulations in the field of biomechanics. In this context, a challenge that biomechanical engineers face is the formulation and solution of such problems, as even with several solvers available, writing algorithms is not always simple or the results are not always satisfactory, given the limitations of the solver. The objective of this work was to compare the use of OpenSim Moco and Fmincon to solve an optimal control problem applied to a neuromusculoskeletal model of a human leg performing an open kinematic chain movement. The results obtained proved that both solvers are able to lead well with human neuromusculoskeletal models, providing physically possible solutions on human torque, muscle activations and movement, being useful for the understanding of human movement as well as the development of devices and protocols related to the biomechanical field. When comparing the usability of the two solvers, the OpenSim Moco proved to be easier to use in addition to being fast and effective in solving the problem, while Fmincon, despite being reasonably more complex and slower, proved to be more flexible.*

Keywords: *Biomechanical simulation, predictive simulation, optimal control problem, neuromusculoskeletal model*

1. INTRODUCTION

Human neuromusculoskeletal models and computer simulations are powerful tools in the field of biomechanics, as they allow the analysis and understanding of human movement, which is important for the development of devices and protocols in the broad areas of healthcare, sports, industry, military and entertainment.

To ensure that the simulations are reliable, it is necessary that, in addition to having the anthropometry similar to that of a real person, the neuromusculoskeletal model is driven by a control capable of reproducing human control as closely as possible. A possible approach is to use optimal controls, where the movement that should be realized by the model does not follow a predefined trajectory, but is performed trying to minimize a certain functional cost, for example, muscle activation. This approach is admissible, since it is natural for living beings to carry out their movements in order to minimize a biological cost.

For that, a neuromusculoskeletal model is chosen and an optimal control problem (OPC) is formulated, defining the cost function to be minimized and the constraints imposed on the model and the movement. Then, this problem is solved using a solver, obtaining then the states and controls related to the execution of the movement (e.g. joint positions and muscle activations). Computational simulations that are based on solve optimal control problems are called *predictive simulations*. Such a simulations are powerful and flexible, allowing to answer "what-if" type questions, since it is possible to consider a large variety of conditions (Lee and Umberger, 2016). As a limitation, this type of simulation requires a higher computational cost. However, the technological advances allow it to be successfully used to study pedaling movement (Park *et al.*, 2022), influence of lower limb exoskeleton (Bianco *et al.*, 2022), interaction between human and upper limb assistance device (ChengXin *et al.*, 2020), assessment of different metabolic cost calculations of gait (Koelewijn *et al.*, 2019), simulation of musculoskeletal movement (Lee and Umberger, 2016) and analysis of loaded and inclined walking (Dorn *et al.*, 2015).

In this context, some challenges that biomechanical engineers face are the development of the neuromusculoskeletal models and formulation and solution of optimal control problems applied to these models, because even with several solvers available, writing algorithms is not always simple or the results are not always satisfactory, given the limitations of the solver.

Although many works use predictive simulations, most of them employ specific models and algorithms, which makes portability to other research groups difficult and sometimes unfeasible. The platform OpenSim (Delp *et al.*, 2007; Seth *et al.*, 2018) provides neuromusculoskeletal models that can be edited, adapted and shared in a relatively easy way, being highly portable. However, although the platform works well with tracking simulations (a type of simulation where the neuromusculoskeletal model is compelled to perform a movement according to a reference trajectory) through computed muscle control (Thelen *et al.*, 2003), it does not support predictive simulations by itself.

Lee and Umberger (2016) developed an algorithm in MATLAB, using the OpenSim API to perform predictive simulations using the OpenSim models, but, despite having produced good results, the algorithm requires from the user a reasonable knowledge of optimal control and direct collocation method to be adapted to other applications and models. Furthermore, even for simpler models and tasks, the algorithm is reasonably extensive and complex, which may discourage biomechanists from using it to perform predictive simulations in their research.

Dembia *et al.* (2020) developed a software toolkit called *OpenSim Moco* that uses direct collocation method to solve optimal control problems applied to optimization of motion and control of neuromusculoskeletal models built in the OpenSim. The software was conceived in a way to allow the users to perform predictive simulations without worrying about implementing the direct collocation method themselves. Despite having presented good results initially, meeting the researchers expectations, a comparison between Moco and other solvers, such as the well-known Fmincon (MathWorks, 2023), in order to verify advantages and limitations was not made.

The purpose of this work was to compare the use of two different solvers to find the solution of an optimal control problem applied to a neuromusculoskeletal model of a human leg executing an open kinematic chain movement, evaluating the advantages and limitations of each solver, providing a basis for scientists from the area of biomechanics choose between the two which is the best option for their projects. The chosen solvers are Fmincon, as it is well known, and OpenSim Moco, which, in addition to being free, was developed especially for biomechanical models. The comparison of both solvers will be based on the evaluation of their flexibility, complexity of writing the algorithm, time to solve the problem and coherence of the results. This work will provide data so that biomechanical scientists can make quick and objective decisions about which of the two solvers to use, better understanding the advantages and limitations of each one in relation to their needs.

2. METHODOLOGY

In this section we present in details the model used, the movement proposed, how the optimal control problem was formulated and solved and how the results were analyzed. The methodology used here was based on the work of Lee and Umberger (2016), for purposes of comparing results.

2.1 Human Neuromusculoskeletal Model

The human neuromusculoskeletal model is based on the *leg6dof9musc* provided by OpenSim and consists of a single human leg, without the trunk and upper limbs. The model has 7 bodies representing four physiological segments (pelvis, thigh, shank, and foot) which are interconnected by the hip, knee and ankle joints and actuated by 9 muscles. As done by Lee and Umberger (2016), in this work the pelvis was fixed in space and the hip, knee and ankle joints were allowed to move only in the sagittal plane, limiting the original model to 3 degrees of freedom (DOF) and contributing to a reduction in computational cost. To make the model suitable for direct collocation, the default Thelen2003 muscles (Thelen, 2003) have been replaced by DeGrooteFregly2016 muscles (Groote *et al.*, 2016).

To facilitate the analyzes that were carried out in this work, all the muscles in the model were organized into groups according to the main joint activated by the muscle in question. Figure 1a shows the neuromusculoskeletal model indicating its muscles. A breakdown of the function of each muscle, as well as the definition of the main joint actuated by it, is provided below.

GROUP HIP: The hip is actuated mainly by the gluteus maximus (GM) and psoas (PS) that are the extension and flexion muscles, respectively.

GROUP KNEE: The flexion of the knee is performed by the biceps femoris long head (BFLH) and biceps femoris short head (BFSH) muscles while the extension is performed by the rectus femoris (RF) and vastus intermedius (VI) muscles. The biceps femoris long head and the rectus femoris also contribute for the flexion and extension of the hip, respectively, but the main articulation moved by these muscles is the knee.

GROUP ANKLE: The medial gastrocnemius (MG) and the soleus (SL) muscles perform the extension of the ankle while the tibialis anterior (TA) is responsible by the flexion of the joint. The medial gastrocnemius also helps the knee flexion, but its main contribution is for the ankle.

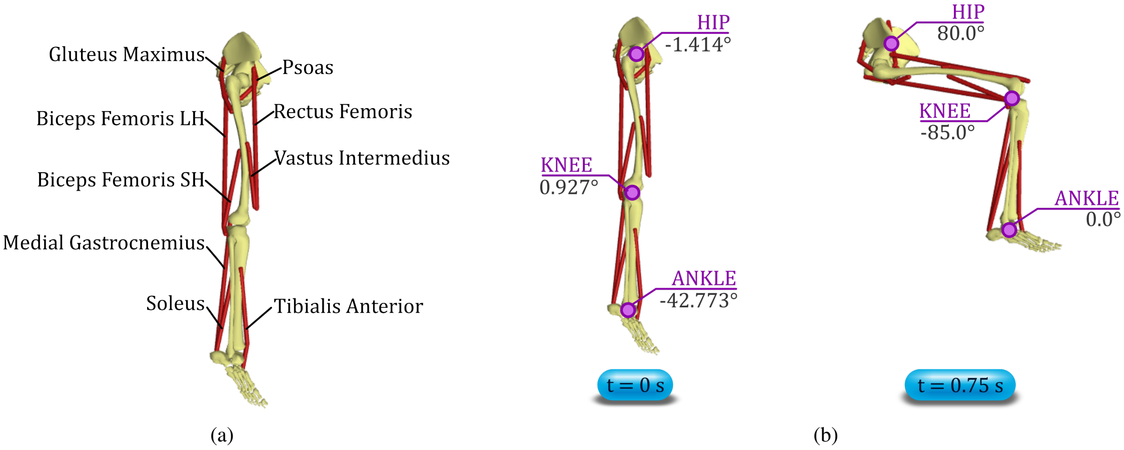


Figure 1. The neuromusculoskeletal model used in this work (a) and the point-to-point movement simulated (b)

2.2 Point-to-point Movement

The movement simulated in this work was a point-to-point one, with the model required to move between a initial, relaxed and hanging position and a final target pose, in a period of 0.75 seconds. The joint angles for the initial position are $-1.414°$ for the hip, $0.927°$ for the knee and $-42.773°$ for the ankle. The joint angles for the final position are $80°$ for the hip, $-85°$ for the knee and $0°$ for the ankle. So, it is expected a flexion in the hip, knee and ankle joints, as can be seen in Figure 1b.

This movement was chosen because it is the same one simulated by Lee and Umberger (2016), allowing us a comparison with their work

2.3 Optimal Control Problem

To solve an optimal control problem involves determining the states and controls that drive a dynamical system from an initial state to a final desired state, minimizing a determined cost functional and satisfying a set of constraints.

In this work we used a cost function seeking to minimize the muscle activations. Such objective function is expressed by Equation (1), where T is the period (0.75 s), a_i is the activation of the ith muscle, m is the number of muscles (that is, $m = 9$).

$$J = \frac{1}{T}\left[\sum_{i=1}^{m}\int_{t_0}^{t_f} a_i^2(t)dt\right] \tag{1}$$

The constraints considered in the optimal control problem of this work are related to the dynamics of the neuromusculoskeletal model (Equation (2)), path constraints on the states (Equation (3)), muscles activations (Equation (4)) and boundary constraints on the time (Equations (5)) and initial and final states (equations (6) - (7), respectively). The symbols in bold represent vectors.

$$\dot{x}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \tag{2}$$

$$\boldsymbol{x}_{min} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{max} \tag{3}$$

$$0 \leq \boldsymbol{a}(t) \leq 1 \tag{4}$$

$$0 \leq t_0 \leq t_f \leq 0.75 \tag{5}$$

$$\boldsymbol{x}_{0,L} \leq \boldsymbol{x}(t_0) \leq \boldsymbol{x}_{0,U} \tag{6}$$

$$\boldsymbol{x}_{f,L} \leq \boldsymbol{x}(t_f) \leq \boldsymbol{x}_{f,U} \tag{7}$$

To solve this OCP we used the OpenSim Moco[1]which is a software toolkit developed by Dembia *et al.* (2020), easy-to-use, extensible, customizable and the first musculoskeletal direct collocation tool that handles kinematic constraints. With such a toolkit, biomechanists are free from implement both equations of motion and direct collocation allowing them to focus on their scientific research questions. The OpenSim Moco has an embedded CasADi library (Andersson *et al.*, 2018) which is used to transcribe the continuous optimal control problem into a finite dimensional nonlinear programming (NLP) which is then solved by the open-source solver IPOPT using gradient-based methods.

Then, the results obtained were compared with the ones obtained by solving the same OCP using the Fmincon from MATLAB. In this case we used the results provided by Lee and Umberger (2016), who developed an algorithm using Fmincon to solve the aforementioned problem.

2.4 Analysis Procedure

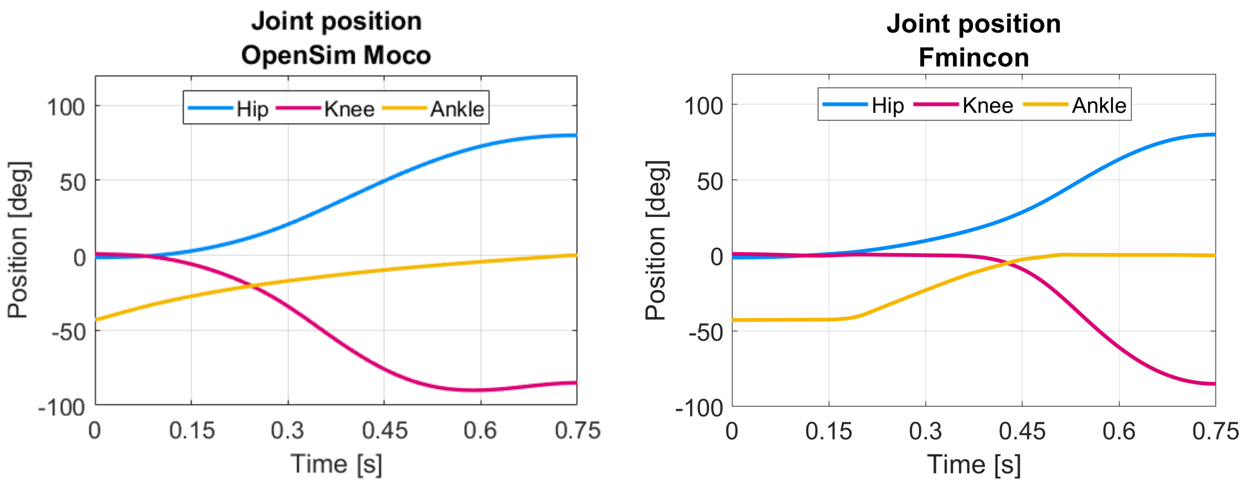
To analyze the results obtained, we considered the trajectory of the movement, as well as the torques and the muscle activations required to perform the motion. We analyzed if the time values and graph shapes obtained for each magnitude were coherent and physically possible for the movement proposed.

We also made considerations about the complexity in to prepare and execute the simulations, the time spent on each execution and the advantages and disadvantages of using OpenSim Moco against the Fmincon a used by Lee and Umberger (2016).

3. RESULTS AND DISCUSSION

In this section, the results obtained with the resolution of the proposed OCP, using OpenSim Moco and Fmincon, are presented and discussed. Next, a comparison is made between the software in terms of flexibility, ease of use, speed of execution, degree of optimization and computational cost.

Observing the Fig. 2 it is possible to notice that the movement was accomplished as expected, with all joints starting and ending in the desired positions, within the stipulated time. Comparing the results from Fig. 2a and b we can observe that with OpenSim Moco the model performed the movement more smoothly, in a continuous way, whereas with Fmincon from MATLAB, the movement was not as smooth, especially for the knee joint, which started the movement at $t = 0.37$ s and the ankle joint, which started the movement only at $t = 0.18$ s in an almost non-linear manner.

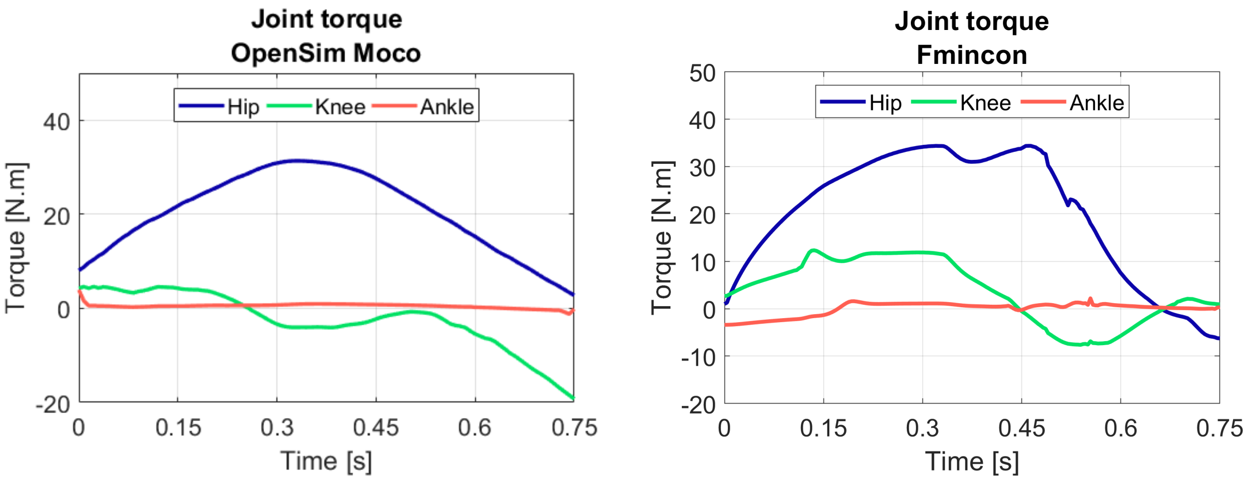


(a) (b)

Figure 2. Angular position of each joint obtained with OpenSim Moco (a) and Fmincon (b)

The Fig. 3a presents the torques obtained with Moco, as expected, the hip joint required a greater amount of torque, as it must move the thigh, shin and foot, that is, a greater mass than the other joints. On the other hand, the ankle was the joint that required the least torque, due to the low mass it moved. The torque profile presented by the knee is related to its equilibrium while the hip is moved, together with the desired movement of the knee itself. Comparing with the results obtained with Fmincon (Figure 3b), there are no big disparities, and with Moco the torques were applied more smoothly, which is evident when analyzing the positions over time (Fig. 2).

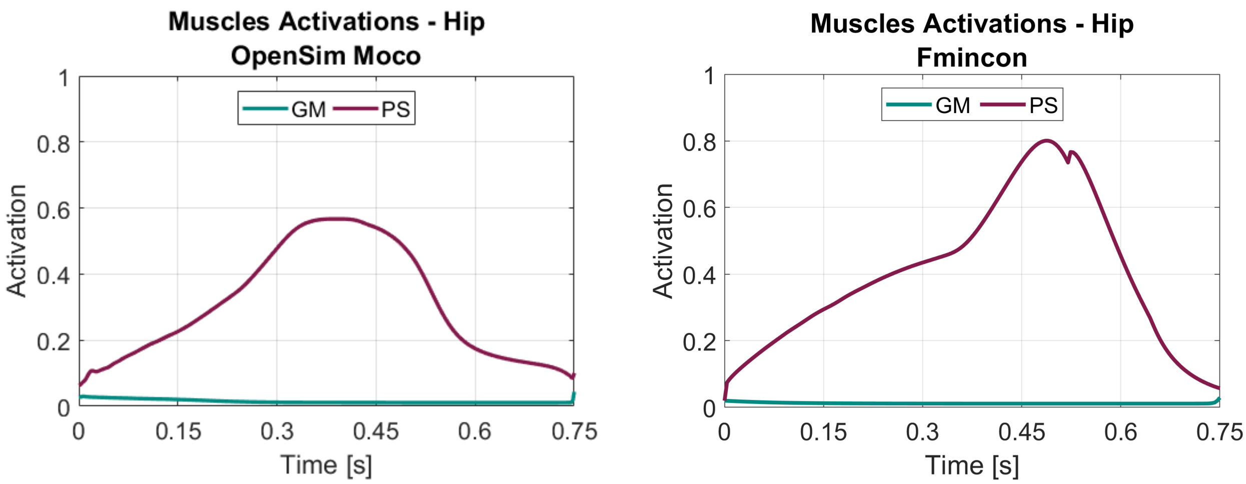
[1] https://opensim-org.github.io/opensim-moco-site/



(a) Torques from OpenSim Moco (b) Torques from Lee and Umberger (2016)

Figure 3. Torque of each joint

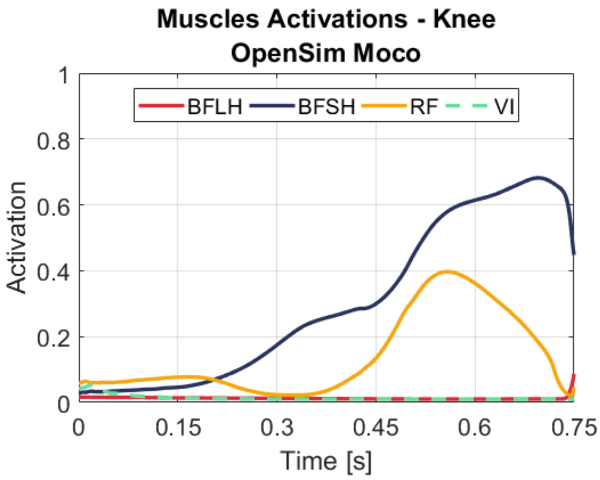
The activations of the muscles whose main joint moved is the hip are presented in the Figure 4. Both for the results obtained with Moco and for the ones obtained with Fmincon, it is possible to notice that the Psoas (PS) muscle was more activated, which was expected, as it is responsible for the flexion of the hip and it was exactly this movement that the joint performed , as can be confirmed by looking at the graphs in Figure 2. As there was no hip extension at any time, the gluteus maximus (GM) muscle remained little activated all the time. The simulation and optimization performed with Moco produced lower activation levels than those obtained by the process using Fmincon.



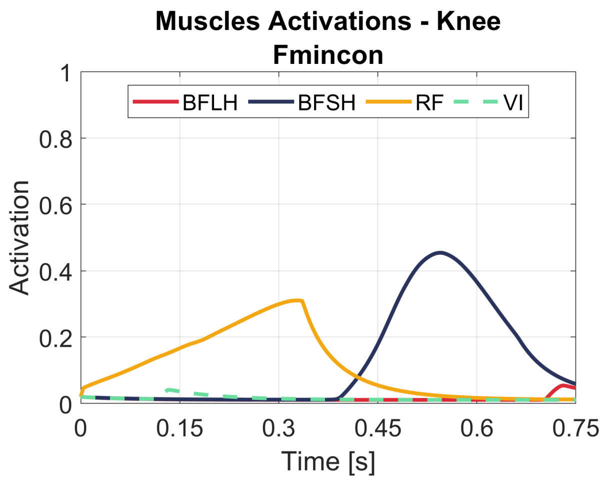
(a) Activations from OpenSim Moco (b) Activations from Lee and Umberger (2016)

Figure 4. Activations of the hip muscles

Observing the Figure 5, it is possible to notice that the biceps femoris long head (BFLH) and the vastus intermedius (VI) were little activated. While the biceps femoris short head (BFSH) and the rectus femoris (RF) were reasonably activate. Although the knee does not undergo extension during the movement, the rectus femoris muscle was activated to act as a brake and also to ensure the stability of the joint that is affected by the movement of the hip, as well as to help in the hip flexion. To move the joint according to the desired movement, the biceps femoris short head was activated. As this muscle is short, more force was required to produce the torque needed for the movement, resulting in a higher activation level than that of the biceps femoris long head. Furthermore, a high activation in the BFLH muscle would negatively impact the hip flexion movement, so that, from an optimal point of view, it is more interesting to have a greater participation of BFSH than BFLH for the given movement. In this simulation, the activations obtained by Lee and Umberger (2016) using Fmincon were at lower levels than those obtained by us through Moco, in addition, the shape of the graphs differ from each other, but they are coherent, considering the movements performed.



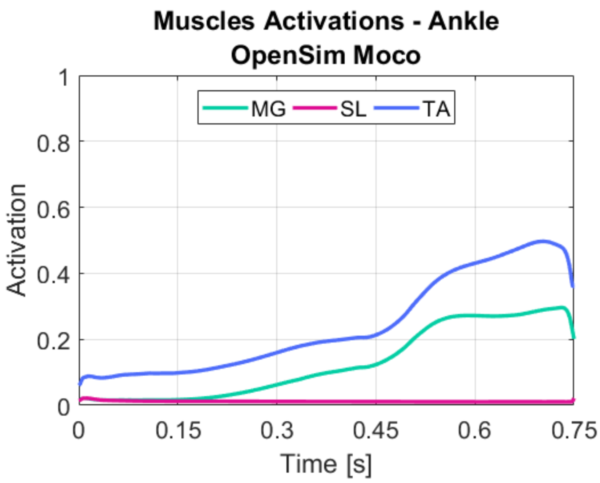
(a) Activations from OpenSim Moco



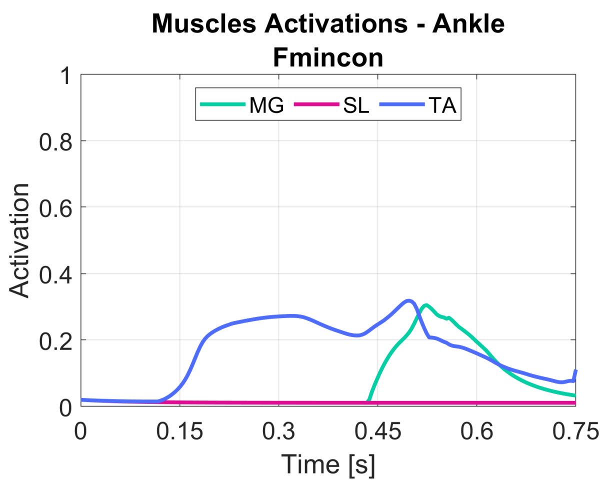
(b) Activations from Lee and Umberger (2016)

Figure 5. Activations of the knee muscles

Regarding the muscles that actuate the ankle, the tibialis anterior (TA) was activated to promote dorsiflexion, the medial gastrocnemius (MG) was also activated both to serve as a brake, controlling movement, and to aid in knee flexion. Little activation was identified for the soleus (SL), as its function would only be to promote plantarflexion, which did not happen during the movement. Again, the activations obtained with Fmincon were smaller than those obtained with OpenSim Moco.



(a) Activations from OpenSim Moco



(b) Activations from Lee and Umberger (2016)

Figure 6. Activations of the ankle muscles

Observing the results obtained with OpenSim Moco and comparing them with those from the work of Lee and Umberger (2016) that used Fmincon to solve the same optimal control problem, we can state that both methods lead very well with human neuromusculoskeletal model, in addition to be able to resolve muscle redundancy, producing optimal movements and consistent results. An evaluation of both solvers, considering the aspects of flexibility, ease of use, speed of execution, degree of optimization.

Ease of use: Considering this aspect, the OpenSim Moco presents a great ease of use in relation to Fmincon. The source code written for Moco is intuitive, while the Fmincon source code is more complex, requiring the user to understand both aspects related to solving optimal control problems (e.g. direct collocation method (Mosconi *et al.*, 2023)) and a reasonable mastery of MATLAB tools. Changes to models, constraints and functions for simulations considering different aspects are simple and fast to be done in OpemSim Moco, while using Fmincon it is necessary that major and exhaustive changes are carried out, even for small changes.

Flexibility: Although OpenSim Moco is easier to operate than Fmincon, it is not as flexible: it is simple to change constraints or cost functions, but this can only be done using the functions and constraints available in the software library. In case the user wants to use a cost function or a path constraint that are specific, Moco will not serve him, since it has a limited library of these functions. Fmincon, on the other hand, is not limited in this aspect, giving freedom for the user to use the functions that are necessary.

Speed of execution: The proposed problem was solved in a much shorter time using OpenSim moco (2.5 minutes). With Fmincon the same problem was solved in approximately two hours. Furthermore, one must consider the time for writing the code, which is much smaller for OpenSim Moco: depending on the simulation to be carried out, with Moco only a few lines of code are necessary, while for Fmincon, even for simulations simple, it takes many lines of code.

Degree of optimization: For the presented problem, OpenSim Moco was able to minimize the cost function better than Fmincon: for the first one the value reached for the cost function was 0.31 while for the second one we obtained 1.3. Despite this data demonstrating an advantage of Moco over Fmincon, more tests must be performed in order to verify this hypothesis. Table 1 presents a comparison between the computational aspects related to both solvers applied to the problem presented in this work.

Table 1. Comparison between optimization aspects of OpenSim Moco and Fmincon

	Collocation points	Time consumed [min]	Cost function
OpenSim Moco	200	2.5	0.31
Fmincon	200	120	1.3

Thus, it can be said that OpenSim Moco is useful for scientists who are not so familiar with the mathematics needed to solve optimal control problems, who want ease in writing and maintaining the source code and agility in running the simulations. On the other hand, Fmincon is recommended for scientists who are familiar with solving optimal control problems, need more flexibility and have time to spend with simulations.

4. CONCLUSIONS

In this work we compared the use of OpenSim Moco and Fmincon to solve an optimal control problem applied to a neuromusculoskeletal model of a human leg performing an open kinematic chain movement.

The results obtained proved that both solvers are able to lead well with human neuromusculoskeletal models, providing physically possible solutions on human torque, muscle activations and movement, being useful for the understanding of human movement as well as the development of devices and protocols related to the biomechanical field.

When comparing the usability of the two solvers, the OpenSim Moco proved to be easier to use in addition to being fast and effective in solving the problem, while Fmincon, despite being reasonably more complex and slower, proved to be more flexible.

For future work, we intend use the solvers to run predictive simulations with more complex movements, such as gait free and with obstacles, in addition to using models that can represent a human wearing a lower limbs exoskeleton.

5. ACKNOWLEDGMENTS

This work is supported by Pro-Rectory of Research of University of São Paulo, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, PGPTA, under grant 3457/2014, and São Paulo Research Foundation (FAPESP) under grant 2019/05937-7. This study was approved by the Ethics Committee of the Federal University of São Carlos (Number 26054813.1.0000.5504).

6. REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B. and Diehl, M., 2018. "CasADi: a software framework for nonlinear optimization and optimal control". *Mathematical Programming Computation*, Vol. 11, No. 1, pp. 1–36. doi: 10.1007/s12532-018-0139-4.

Bianco, N.A., Franks, P.W., Hicks, J.L. and Delp, S.L., 2022. "Coupled exoskeleton assistance simplifies control and maintains metabolic benefits: A simulation study". *PLOS ONE*, Vol. 17, No. 1, p. e0261318. doi: 10.1371/journal.pone.0261318.

ChengXin, Y., Benali, A., WeiWei, Y. and Monacelli, E., 2020. "Investigation of Human-Device Interaction via Predictive Simulation". In *Proceedings of the 11th Augmented Human International Conference*. Association for Computing Machinery, New York, NY, USA, AH '20. ISBN 9781450377287. doi:10.1145/3396339.3396386. URL https://doi.org/10.1145/3396339.3396386.

Delp, S.L., Anderson, F.C., Arnold, A.S., Loan, P., Habib, A., John, C.T., Guendelman, E. and Thelen, D.G., 2007. "Opensim: Open-source software to create and analyze dynamic simulations of movement". *IEEE Transactions on Biomedical Engineering*, Vol. 54, No. 11, pp. 1940–1950. doi:10.1109/TBME.2007.901024.

Dembia, C.L., Bianco, N.A., Falisse, A., Hicks, J.L. and Delp, S.L., 2020. "OpenSim moco: Musculoskeletal optimal control". *PLOS Computational Biology*, Vol. 16, No. 12, p. e1008493. doi:10.1371/journal.pcbi.1008493.

Dorn, T.W., Wang, J.M., Hicks, J.L. and Delp, S.L., 2015. "Predictive simulation generates human adaptations during loaded and inclined walking". *PLOS ONE*, Vol. 10, No. 4, p. e0121407. doi:10.1371/journal.pone.0121407.

Groote, F.D., Kinney, A.L., Rao, A.V. and Fregly, B.J., 2016. "Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem". *Annals of Biomedical Engineering*, Vol. 44, No. 10, pp. 2922–2936. doi:10.1007/s10439-016-1591-9.

Koelewijn, A.D., Heinrich, D. and van den Bogert, A.J., 2019. "Metabolic cost calculations of gait using musculoskeletal energy models, a comparison study". *PLOS ONE*, Vol. 14, No. 9, p. e0222037. doi:10.1371/journal.pone.0222037.

Lee, L.F. and Umberger, B.R., 2016. "Generating optimal control simulations of musculoskeletal movement using OpenSim and MATLAB". *PeerJ*, Vol. 4, p. e1638. doi:10.7717/peerj.1638.

MathWorks, 2023. *MATLAB - Optimization Toolbox Userś Guide*. MathWorks.

Mosconi, D., Luizete, C.E. and Siqueira, A.A.G., 2023. *Engenharias: Automação, Robótica, metrologia e Energia*, Científica Digital, Vol. 1, chapter Direct collocation method for solving optimal control problems, p. 18. I edition. ISBN 978-65-5360-278-6. doi:10.37885/978-65-5360-278-6.

Park, S., Caldwell, G.E. and Umberger, B.R., 2022. "A direct collocation framework for optimal control simulation of pedaling using OpenSim". *PLOS ONE*, Vol. 17, No. 2, p. e0264346. doi:10.1371/journal.pone.0264346.

Seth, A., Hicks, J.L., Uchida, T.K., Habib, A., Dembia, C.L., Dunne, J.J., Ong, C.F., DeMers, M.S., Rajagopal, A., Millard, M., Hamner, S.R., Arnold, E.M., Yong, J.R., Lakshmikanth, S.K., Sherman, M.A., Ku, J.P. and Delp, S.L., 2018. "OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement". *PLOS Computational Biology*, Vol. 14, No. 7, p. e1006223. doi:10.1371/journal.pcbi.1006223.

Thelen, D.G., 2003. "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults". *Journal of Biomechanical Engineering*, Vol. 125, No. 1, pp. 70–77. doi:10.1115/1.1531112.

Thelen, D.G., Anderson, F.C. and Delp, S.L., 2003. "Generating dynamic simulations of movement using computed muscle control". *Journal of Biomechanics*, Vol. 36, No. 3, pp. 321–328. doi:10.1016/S0021-9290(02)00432-3.

7. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.