PREDICTIVE SIMULATIONS OF BIOMECHANICAL SYSTEMS USING OPENSIM MOCO AND MATLAB

Denis Mosconi, Mechanical Engineering Department – University of São Paulo, denis.mosconi@usp.br
Melkzedekue Moreira, Mechanical Engineering Department – University of São Paulo, melkzedekue@usp.br
Adriano A. G. Siqueira, Center of Engineering Applied to Healthy – University of São Paulo, siqueira@sc.usp.br

Resumo. *Computer predictive simulations are a powerful resource in the field of biomechanics. Such simulations allow to study biomechanical models under various and different conditions, without the need to collect experimental data. In this work we used the OpenSim Moco to carry out predictive simulations with a Tug of War-base model, in order to asses the effectiveness of the software in perform this type of simulation. The results obtained proved that, despite some limitations, the OpenSim Moco provide solutions in a coherent, flexible and agile manner, being a promising tool for researches in the biomechanical field.*

Keywords: *OpenSim Moco. Predictive Simulation. Biomechanical model.*

1. INTRODUCTION

Computer simulation is a powerful tool that has been widely used in the field of biomechanics. This resource brings agility and flexibility to the research, as it allows numerous simulations using biomechanical computational models, under different conditions in a relatively short time, which is advantageous, especially for experimental studies (Jansen and McPhee, 2020). Simulations can be divided in two groups: data-tracking and predictive.

In the tracking case, the objective is to minimize the difference between the behavior of a computational model and a target set of data experimentally collected, such as ground reaction forces (GRF) and joint kinematics (Lee and Umberger, 2016).The data-tracking approach can be used to support the development of human-exoskeletons interaction models (Mosconi et al., 2020), development of interaction controls in robotic rehabilitation (Peña et al., 2019), determination of human joint torques (Nunes et al., 2018) and to perform forward dynamics simulations of human walking (Thelen and Anderson, 2006). Despite being widely used in the field of biomechanics, tracking simulations are limited by the need for data collection, in addition to being restricted to the conditions imposed by such data.

Predictive simulations are performed based on a optimality criterion (e.g. minimal effort, minimal joint load) that must be satisfied while the model complete a task (e.g to move from a pose to another). Thus, the states and controls involved are not determined by following a reference, which allows predicting responses to the most varied conditions. Some applications of predictive simulations are prediction of human gait (Ackermann and van den Bogert, 2010), musculoskeletal movements (Lee and Umberger, 2016), solve the muscle redundancy problem (Groote et al., 2016) and simulation of Olympic cycling stand start (Jansen and McPhee, 2020). The challenge imposed by this type of simulation is the computational cost involved, but despite this, such approach allows to answer "*what-if*" types of questions, as well as, to consider a wide range of different conditions, not limited to a set of experimental data, that can establish cause-effect relationships (Lee and Umberger, 2016).

The optimality criterion of the predictive simulation is satisfied by solving an optimal control problem. Consequently, the states and controls related to the model and the task it must accomplish are optimal. A manner of solve the optimal control problem formulated is through the direct collocation method that transcribes this continuous problem into a finite-dimensional nonlinear programming (NLP) problem that can be solved by well know techniques (Gill et al., 2008; MathWorks, 2021; Wächter and Biegler, 2005; Betts, 2010; Byrd et al., 2006).

Recently Dembia et al. (2020) introduced the *OpenSim Moco*, an easy-to-use, extensible and customizable software toolkit that uses the direct collocation method to solve optimal control problems with biomechanical models provided by OpenSim[1]. According to the developers, with this resources the biomechanists are free both from implementing equations of motion and direct collocation optimization, needing only to set up the costs and constraints from a library provided by the software and thus run predictive simulations in a quick and easy way, focusing exclusively on the use of optimal control resources and not on their development.

The objective of this work was to perform predictive simulations using OpenSim Moco and variations of the simple and well-accepted biomechanical model *Tug of War* from OpenSim, in order to evaluate the speed of execution of the simulations, the coherence of the obtained solutions, the flexibility and complexity of the software and the possibility of extending its application to more complex biomechanical models, such as human-exoskeleton interaction models, which

[1] http://opensim.stanford.edu

are very useful for research in robotic neurorehabilitation. Furthermore, it was intended to describe the advantages and limitations of using such software for to carry out predictive simulations.

This paper is organized as follows. In Section 2.we introduce the optimal control problem, describe the OpenSim Moco and models used and present how the analysis was performed. In Section 3.we present the solutions obtained with the predictive simulations and lastly, conclusions are shown in Section 4.

2. METHODOLOGY

In order to asses the capabilities of the software OpenSim Moco, as described in the previous section, a simple and well accepted biomechanical model and a well-defined movement were chosen to be simulated in this work.

The model in question is the *Tug of War* provided by OpenSim, such model consists of a massive block translating on a frictionless surface under the force action of two opposing muscle-tendon actuators attached to fixed supports (Fig. 1). The block is a cube with a 0.1 m edge and 20 kg mass. The distance between the fixed supports is 0.7 m and when the block is centered, muscle-tendon lengths are 0.3 m. Variations of this model were built in order to promote simulations with different models.

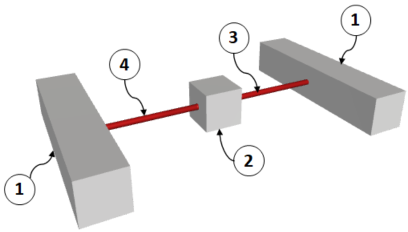


Figura 1. Biomechanical model of Tug of War provided by OpenSim. (1) are the fixed supports, (2) is the block, (3) is the muscle 1 and (4) is the muscle 2.

The simulated movement consists of the block starting from rest at position -0.08 m, being translated to position 0.08 m and finally returning to position -0.08 m where it should remain.This movement was chosen because it is relatively simple and well known, and it can even be determined analytically for dynamic systems of low complexity Kelly (2017).

Choosing simple and well-known models and motion allows us to turn our attention exclusively to analysis of the software and solution methods, as mentioned above, without worrying about the models and motions in question.

2.1 Optimal Control Problem Formulation

The optimal control problem was formulated as a trajectory optimization, that is, to find the optimal states and controls that best describe the movement performed by the block as it moves from the given initial to final points.

The objective function to be minimized is the sum of the absolute values of the controls related to the muscles and actuators of the model (Eq. 1).

$$\frac{1}{d}\int_{t_i}^{t_f}\sum_{c\in C} w_c|u_c(t)|^{\rho}dt \tag{1}$$

Where d is the displacement of the system, C is the set of control signals, w_c is the weight for control c (unitary in this work), $u_c(t)$ is the control signal c and ρ is the exponent (2 in this work).

The objective function must be minimized subject to the system multibody dynamics (Eq. 2), the kinematic constrains (Eq. 3), the boundary constrains (Eq. 4), the path constrains (Eq. 5) and the initial and final states and controls (Eq. 6).

$$M(q,p)\ddot{q} + G(q,p)^T\lambda = f_{app}(t,y,u,p) - f_{inertial}(q,\ddot{q},p) \tag{2}$$

$$0 = \phi(q,p) \tag{3}$$

$$V_{L,k} \leq V(t_0,t_f,y_0,y_f,u_0,u_f,\lambda_0,\lambda_f,p) \leq V_{U,k} \tag{4}$$

$$g_L \leq g(t,y,u,\lambda,p) \tag{5}$$

$$\begin{gathered} y_{0,L} \leq y_0 \leq y_{0,U} \quad y_{f,L} \leq y_f \leq y_{f,U} \\ u_{0,L} \leq u_0 \leq u_{0,U} \quad u_{f,L} \leq u_f \leq u_{f,U} \end{gathered} \tag{6}$$

Where $M(q,p)$ is the mass matrix, q and $\ddot{q}$ are the joint position and acceleration, respectively, $G(q,p)$ is the Jacobian matrix whose transpose converts the Lagrange multipliers λ into generalized forces along the system's degrees of freedom, $f_{app}(t,y,u,p)$ are the applied forces from muscles and actuators, $f_{inertial}(q,\ddot{q},p)$ are the centripetal and Coriolis forces, y, u and p are the states, controls and parameters, respectively.

2.2 OpenSim Moco

OpenSim Moco[2] (OpenSim Musculoskeletal Optimal COntrol) was developed by Dembia et al. (2020) and is a software that allows to perform both tracking and predictive simulation. For this, it solve a optimal control problem using direct collocation method.

The software was designed to free the researchers from implement the equations of motion of the models and the optimization methods, so that they can focus exclusively on the execution of the simulation, being concerned solely with the research object in question.

There Moco has a library with various goals functions and constrains pre-developed, but the user can feel free to create their own goal function and constrain. To perform the simulation, the user determines the model, the goal functions and optionally introduces some experimental data, then the OpenSim Moco executes the optimization and provides the solution.

2.3 Tug of War-based models

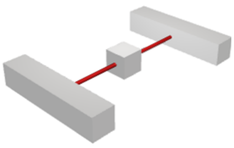
As mentioned above, the models used in this work were based on the Tug of War model provided by OpenSim. Below, each of the models used is presented and described.

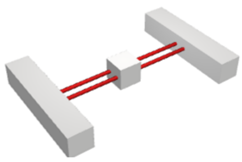
Model 1: This model was obtained by replacing the 6 DOF free joint from the original Tug of War by a 1 DOF slider joint. In this work both muscles have optimal fiber lengths of 0.25 m, peak isometric forces of 1000 N, tendon slack lengths of 0.05 m, and pennation angles of 0°. This model was used by Lee and Umberger (2016) and can be seen in Fig. 2a (the red lines are the muscles).

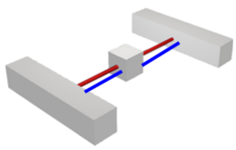
Model 2: In this model (Fig. 2b), two muscles were added on each side of the block from Model 1. The objective in use such model is to verify how well OpenSim Moco solves the muscle redundancy problem.

Model 3: This is also a variation of model 1, in which a single coordinate actuator has been added, capable of providing a generalized force along the axis of motion of the slider joint. The model can be seen in Fig. 2c (the red lines are the muscles and the blue lines represent the coordinate actuator). The purpose in use this model is to asses how the Moco distributes the work among various types of actuators.

Model 4: In this model (Fig. 2d) all the muscles were removed and only the coordinate actuator was kept. With this model we compare the effectiveness of OpenSim Moco in performing inverse dynamics. The results obtained were compared with an inverse dynamics performed using the *Inverse Dynamics Tool* from OpenSim.


(a) Model 1


(b) Model 2


(c) Model 3

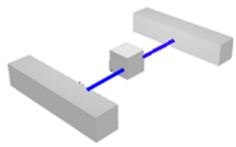

(d) Model 4

Figura 2. Biomechanical models based on the Tug of War. The red lines are the muscles and the blue lines represents the coordinate actuator.

2.4 Analysis

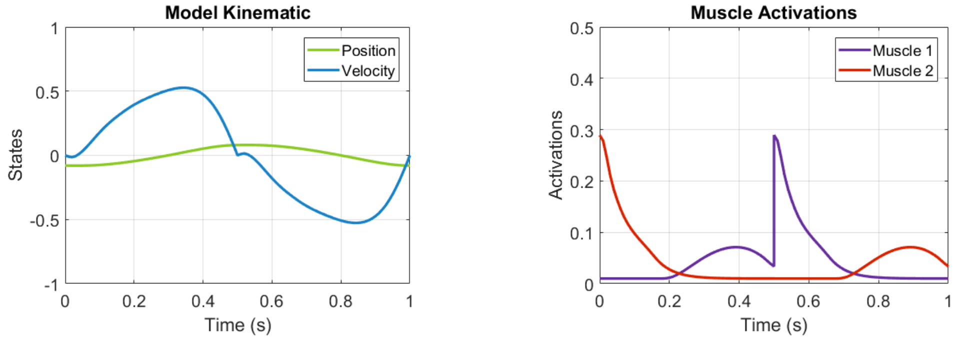
In order to asses the coherence of the obtained solutions and if they are optimal, we analyzed both the movement performed and the control signals determined. The time took to perform each simulation was also measured and the complexities involved in prepare and execute the simulations was also analyzed, in order to verify if the Moco is agile and easy-to-use as described by the developers.

The simulations were performed on a computer with Intel®Core™i7-10510U 2.30 GHz processor, 8.00 GB of RAM, 2.00 GB dedicated video card, 512 GB SSD PCIe 3.0 x2 NVMe (M.2 2280) and Windows 10 Pro 64 bits.

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

3. RESULTS

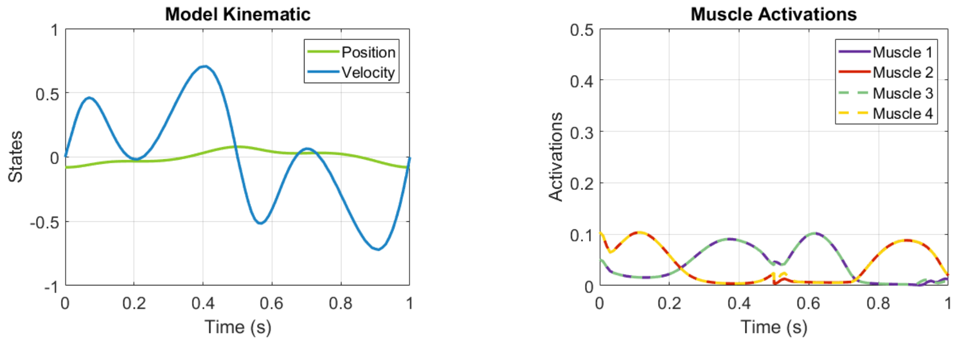
Analyzing the Fig. 3a it is possible to notice that the with the Model 1, the OpenSim Moco determined the optimal movement. However, there is a non-linearity in the velocity, at a moment of 0.5 s, which may be undesirable if this variable is used in a practical application, such as an impedance control. About the muscle activations Fig. 3b, in a first moment the muscle 2 starts activated in order to keep the block in the -0.08 m position, then this activation is reduced and muscle 1 exerts a passive force that brings the block to the 0.08 m position. Note that from 0.2 s, muscle 1 starts to act actively, since at that moment the passive force is not enough to keep moving the block. Then, from 0.5 s the movement starts again in the opposite direction, with muscle 1 starting activated to keep the block in the 0.08 m position, followed by a reduction in this activation and muscle 2 performing passive force until 0.7 s, when it starts to exert active force to end the movement with the block at 0.08 m. When compared to the work of Lee and Umberger (2016), the results are approximately the same, but the time to perform the simulation are different: with Moco the simulation took 6 seconds to be finished and with Fmincon (that Lee and Umberger (2016) used) 17 minutes (1020 seconds).



(a) Kinematic (b) Muscle Activations

Figura 3. Kinematic (a) and muscle activations (b) obtained with the Model 1.

When the amount of muscles was doubled (Model 2) the activations of each muscle were reduced (Fig. 4b), as expected, as the force is now distributed among more actuators.Despite the consistent result regarding the reduction in the level of muscle activation, it is noted that there is an increase in oscillations, both in activations and in velocity (Fig. 4a), which is undesirable for interaction controls involving speed and activations. The running movement also deviated slightly from the optimal one, however the starting, intermediate and ending positions were reached by the block. The time took in this simulation was 3.5 minutes.

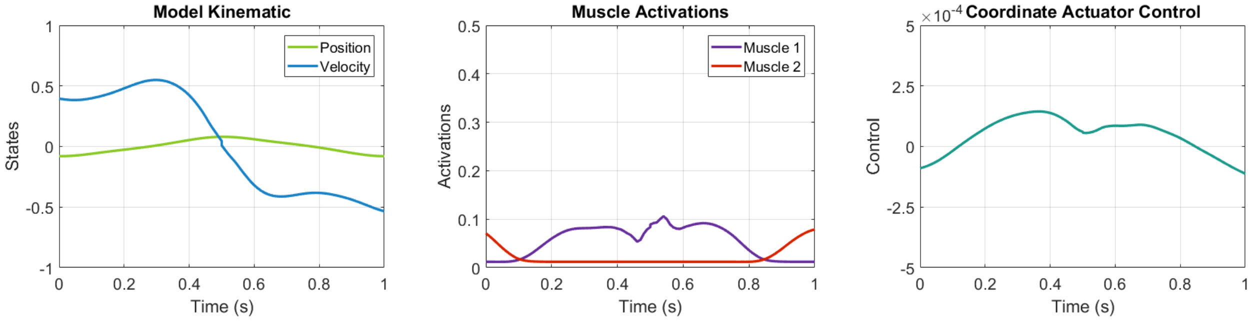


(a) Kinematic (b) Muscle Activations

Figura 4. Kinematic (a) and muscle activations (b) obtained with the Model 2.

When the coordinate actuator was used (Model 3) the muscle activations were reduced since the actuator since the actuator behaved as both an accelerator and a brake. This made the block move with an optimal trajectory, with a velocity with few oscillations and with better continuity. Its is possible to affirm that until now the Moco was able to lead with various actuators (both in number and in type) and solve redundancy problems. This simulation took 9.77 seconds to be accomplished.

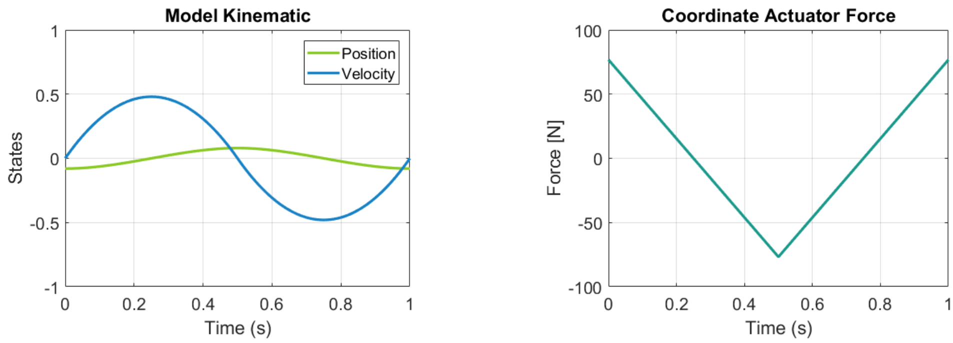
An inverse dynamics (ID) was performed using the Model 4, that has only one coordinate actuator, without muscles. As the dynamics of such actuator is linear, it is possible to notice that both the position and velocity of the block are optimal (Fig. 6a), without non-linearities in the velocity. The force of the actuator (Fig. 6b) also has a optimal trajectory



(a) Kinematic (b) Muscle Activations (c) Actuator control

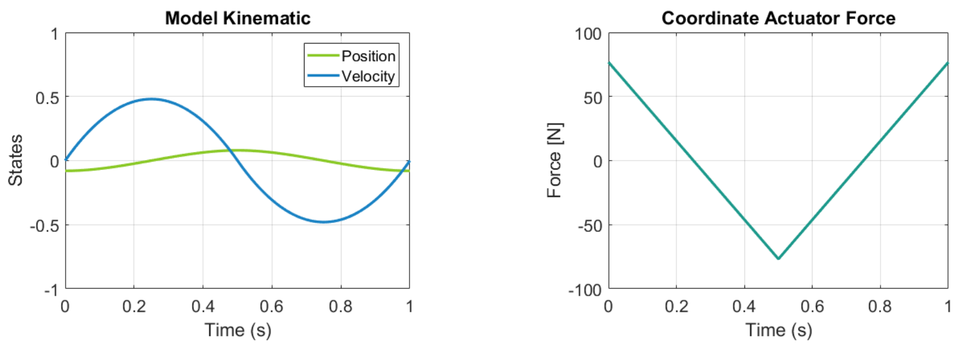
Figura 5. Kinematic(a), muscle activations (b) and actuator control (c) obtained with the Model 3.

and is coherent with the movement performed. Such simulation took 0.64 seconds to be accomplished. When compared with the inverse dynamics performed using the *Inverse Dynamics Tool*, the results obtained are close, with the ID with Moco having a maximum force of 73 N and the ID with Inverse Dynamics Tool having a maximum force of 76 N (This demonstrates that Moco attempt to solve the problem by minimizing the effort, as expected.). The Inverse Dynamics Tool took 0.012 seconds to perform the ID.



(a) Kinematic (b) Actuator Control

Figura 6. Kinematic (a) and actuator control (b) from inverse dynamics performed using OpenSim Moco and Model 4.



(a) Kinematic (b) Actuator Control

Figura 7. Kinematic (a) and actuator control (b) from inverse dynamics performed using Inverse Dynamics Tool and Model 4.

Based on the obtained results, it can be inferred that OpenSim Moco produces relatively fast simulations, when compared to other softwares (e.g. Fmincon), with coherent results. Furthermore, the program is flexible and easy-to-use, as it only took a change in the code line to switch between the models used. However, although writing the code for simulation/optimization is simple and fast (each simulation executed in this work did not have more than 10 lines of code) the biggest limitation of OpenSim Moco lies in the fact that there is little information available about the functions contained in the software library, as well as its syntax rules, which may discourage from using the software.

4. CONCLUSIONS

With the results obtained it is possible to conclude that, despite its limitations, OpenSim Moco is a promising tool to run predictive simulations in the field of biomechanics.

For future works we expected to perform simulations with more complex models, like human musculoskeletal and human-exoskeleton interaction models.

5. REFERENCES

Conor Jansen and John McPhee. Predictive dynamic simulation of olympic track cycling standing start using direct collocation optimal control. *Multibody System Dynamics*, 49(1):53–70, feb 2020. doi: 10.1007/s11044-020-09723-3.

Leng-Feng Lee and Brian R. Umberger. Generating optimal control simulations of musculoskeletal movement using OpenSim and MATLAB. *PeerJ*, 4:e1638, jan 2016. doi: 10.7717/peerj.1638.

Denis Mosconi, Polyana F. Nunes, Icaro Ostan, and Adriano A. G. Siqueira. Design and validation of a human-exoskeleton model for evaluating interaction controls applied to rehabilitation robotics. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, nov 2020. doi: 10.1109/biorob49111.2020.9224436.

G. G. Peña, L. J. Consoni, W. M. dos Santos, and A. A. G. Siqueira. Feasibility of an optimal emg-driven adaptive impedance control applied to an active knee orthosis. *Robotics and Autonomous Systems*, (112):98–108, 2019. doi: 10.1016/j.robot.2018.11.011.

Polyana F. Nunes, Wilian M. dos Santos, and Adriano A. G. Siqueira. Control strategy based on kinetic motor primitives for lower limbs exoskeletons. In *10th IFAC Symposium on Biological and Medical Systems*. International Federation of Automatic Control, 2018.

Darryl G. Thelen and Frank C. Anderson. Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of Biomechanics*, 39(6):1107–1115, jan 2006. doi: 10.1016/j.jbiomech.2005.02.010.

Marko Ackermann and Antonie J. van den Bogert. Optimality principles for model-based prediction of human gait. *Journal of Biomechanics*, 43(6):1055–1060, 2010. ISSN 0021-9290. doi: https://doi.org/10.1016/j.jbiomech.2009.12.012. URL `https://www.sciencedirect.com/science/article/pii/S0021929009007210`.

Friedl De Groote, Allison L. Kinney, Anil V. Rao, and Benjamin J. Fregly. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. *Annals of Biomedical Engineering*, 44(10): 2922–2936, mar 2016. doi: 10.1007/s10439-016-1591-9.

Philip E. Gill, Walter Murray, and Michael A. Saunders. User's guide for SNOPT version 7: Software for large-scale nonlinear programing. 2008. URL `https://web.stanford.edu/group/SOL/guides/sndoc7.pdf`.

MathWorks. MATLAB optimization toolbox. 2021. URL `https://www.mathworks.com/products/optimization.html`.

Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, apr 2005. doi: 10.1007/s10107-004-0559-y.

John T. Betts. *Practical methods for optimal control and estimation using nonlinear programing*. SIAM, Philadelphia, PA, 2nd edition, 2010. ISBN 978-0-898716-88-7.

Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. *Knitro: An Integrated Package for Nonlinear Optimization*, pages 35–59. Springer US, Boston, MA, 2006. ISBN 978-0-387-30065-8.

Christopher L. Dembia, Nicholas A. Bianco, Antoine Falisse, Jennifer L. Hicks, and Scott L. Delp. OpenSim moco: Musculoskeletal optimal control. *PLOS Computational Biology*, 16(12):e1008493, dec 2020. doi: 10.1371/journal.pcbi.1008493.

Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4): 849–904, jan 2017. doi: 10.1137/16m1062569.

6. 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 2013/14756-0 and 2019/05937-7.

7. RESPONSIBILITY NOTICE

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