

Development of a Computational Tool to Estimate Muscle Activations from Musculoskeletal Simulations

Research Topic Introduction

Musculoskeletal simulations can be used to relate the body's motion with muscle forces across a broad spectrum of conditions. Such simulations also allow us to gain insight on the activation profiles of specific muscle groups, which broadens our understanding of the body's neural control strategies during movement. This research aims to investigate whether a computer-generated musculoskeletal simulation can accurately identify individual muscle activations. A Static Optimization (SO) algorithm provides a useful approach but has not yet been implemented as part of a widely available scripting language. [1] Therefore, using MATLAB we will apply the OpenSim API—a program that runs biomechanical calculations such as inverse kinematics (which calculates joint angles) or dynamics (which calculates joint moments)—to build a pipeline that can automate SO calculations for numerous data sets. Our results will be compared against data generated from EMG sensors (which empirically measure electrical muscle activations) as a validation strategy. Thus, we will provide a tool in MATLAB for users to seamlessly calculate muscle activations, which can be used in future biomechanics studies.

Objectives for Semester Project

With a timeline illustrated in Figure 1, we plan to design this tool in five primary steps: OpenSim RRA/SO Testing, RRA Implementation, SO Implementation, Pipeline Development & Scaling, and Data Validation. First, we will test the Residual Reduction Algorithm (RRA) and the Static Optimization (SO) for one data set (trial zero) in native OpenSim software. Using those results, we will be able to tune the RRA and SO controllers that drive musculoskeletal simulations and implement them into MATLAB. After building a working pipeline, we will scale up the program by looping through multiple datasets or by parallelizing the algorithm, so that we can generate results from numerous trials. Throughout the development process, muscle activations will be regularly compared to EMG data (accurate muscle activation data) and an error reduction method will be used to sharpen the accuracy of the algorithms. After the pipeline has been tested for reliability, larger data samples can be calculated using the tool. Figure 2 illustrates the general workflow throughout this semester-long project.

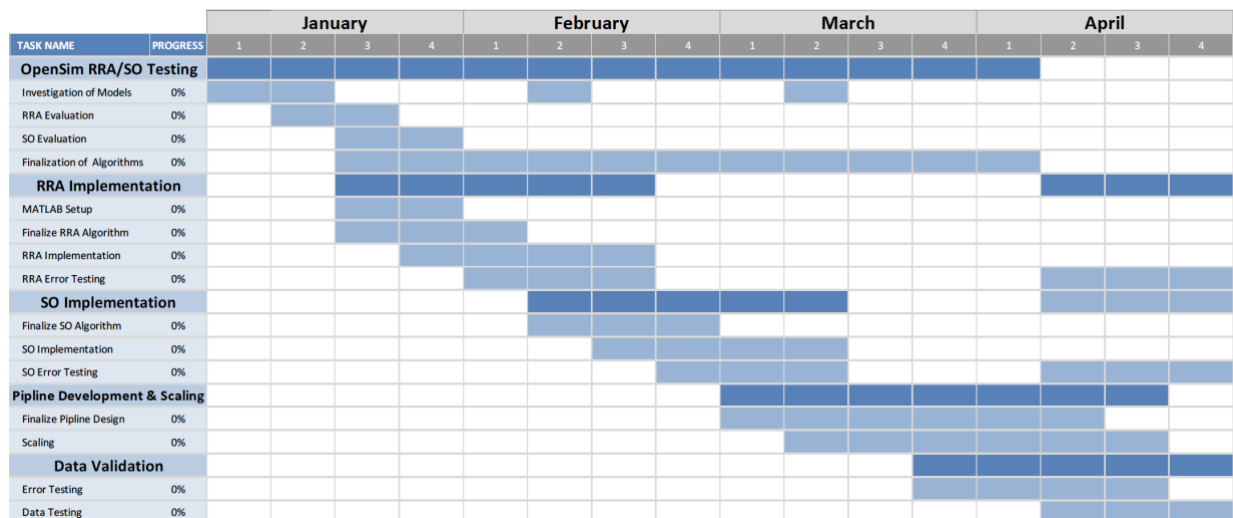


Figure 1: Proposed Project Timeline

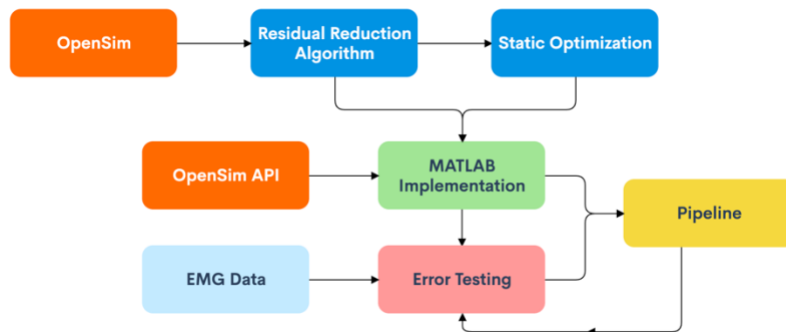


Figure 2: Project Flowchart



Figure 3: OpenSim Musculoskeletal Model

Related Work and Background Information

Previous studies have shown that Static Optimization is the best method to estimate muscle forces for human movement when compared to other commonly used methods (Computed Muscle Control and Neuromusculoskeletal Tracking) because the optimization proves to be more accurate, robust, and efficient. [1] This estimation technique breaks down joint moments to calculate individual muscle activations at specified time points without using EMG (electromyography) data. [2] Experimental data can be preprocessed using a Residual Reduction Algorithm (RRA), which uses external ground forces to balance the torque and force sums from centripetal, coriolis, gravitational, and inertial reactions. [3] This calibrates the coordinate system which can be used in OpenSim to formulate musculoskeletal models (Figure 3) and create dynamic simulations for movement. OpenSim is an open-sourced program, which allows users to analyze and simulate data. [4] OpenSim incorporates key algorithms, such as the Hill-type model of a muscle which uses five muscle parameters (tendon slack length, muscle fiber length, isometric muscle force, pennation angle, muscle fiber velocity) to connect musculotendon and external forces to motion; it uses these calculations to improve upon its own computations and results. [5] MATLAB is an extensively used programming platform that can integrate OpenSim models so that data sets can be modified or stored according to a user's needs.

Methods and Techniques

We will be using the OpenSim version 4 with a full-body musculoskeletal model developed by Rajagopal et al. 2015. [3, 6] The pipeline development work will all be done in MATLAB (ver. R2021a) with the OpenSim API (ver. 4.3), which will be used to incorporate specific algorithms to generate an end-to-end script. Using the API, motion capture and ground reaction force data will be inputted, and muscle activations from Static Optimization will be outputted. Further, to improve upon the reliability of our model we will pre-process our data by using a Residual Reduction/Elimination Algorithm (RRA). This will successfully sort out any inconsistencies between muscle coordinates, measured kinematics, and ground reaction forces and moments. [2] We have previously collected EMG data (2000 Hz, Delsys sensors) from 8 superficial muscles. Root Means Squared Error (RMSE) and R^2 will be used to quantify error.

Progress

Throughout this semester, I have become familiar with concepts such as inverse kinematics and dynamics and how they relate to an individual's movement. I have also helped to collect data for a study measuring gait, which laid the groundwork for my background knowledge in biomechanics. The data that we will use for this project has previously been collected and is fully ready to be processed. The data set consists of seven participants each of which walked unperturbed at 1.25 m/s and 2 m/s, and ran at 2 m/s and 3.25 m/s, in addition to 40 trials where each participant was slipped (rapidly accelerated) on a treadmill.

Mentors

For this project, I am being mentored by Pawel Golyski, a 5th year BioEngineering PhD student exploring the biomechanics associated with the use of lower limb prosthetics and assistive devices. Pawel has 7 years of experience performing biomechanical analyses with human participants and has previously mentored a PURA student.

Bibliography

- [1] Lin, Y. C., Dorn, T. W., Schache, A. G., & Pandy, M. G. (2012). *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of engineering in medicine*, 226(2), 103–112.
- [2] Pizzolato, C., Lloyd, D. G., Sartori, M., Ceseracciu, E., Besier, T. F., Fregly, B. J., & Reggiani, M. (2015). CEINMS: A toolbox to investigate the influence of different neural control solutions on the prediction of muscle excitation and joint moments during dynamic motor tasks. *Journal of biomechanics*, 48(14), 3929–3936.
- [3] Thelen, D. G., & Anderson, F. C. (2006). Using computed muscle control to generate forward dynamic simulations of human walking from experimental data. *Journal of biomechanics*, 39(6), 1107–1115.
- [4] Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., & Thelen, D. G. (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on bio-medical engineering*, 54(11), 1940–1950.
- [5] De Groote, F., Van Campen, A., Jonkers, I., & De Schutter, J. (2010). Sensitivity of dynamic simulations of gait and dynamometer experiments to hill muscle model parameters of knee flexors and extensors. *Journal of biomechanics*, 43(10), 1876–1883.
- [6] 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., Delp, S.L. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Computational Biology*, 14(7). (2018)