Reinforcement learning estimates muscle activations

Gheorghe Lisca[1,2,3], Cristian Axenie[3], Thomas Grauschopf[2,3] & Veit Senner[1]

[1]Technical University of Munich, TUM School of Engineering and Design, Munich, Germany

[2]Technische Hochschule Ingolstadt, Faculty of Computer Science, Ingolstadt, Germany

[3]Audi Konfuzius-Institut Ingolstadt Laboratory, Ingolstadt, Germany

Abstract

A digital twin of the human neuromuscular control can substantially improve the prediction of injury risks and the evaluation of the readiness to return to sport. Reinforcement learning (RL) algorithms already learn physical quantities unmeasurable in biomechanics, and hence can contribute to the development of the digital twin. Our preliminary results confirm the potential of RL algorithms to estimate the muscle activations of an athlete's moves.

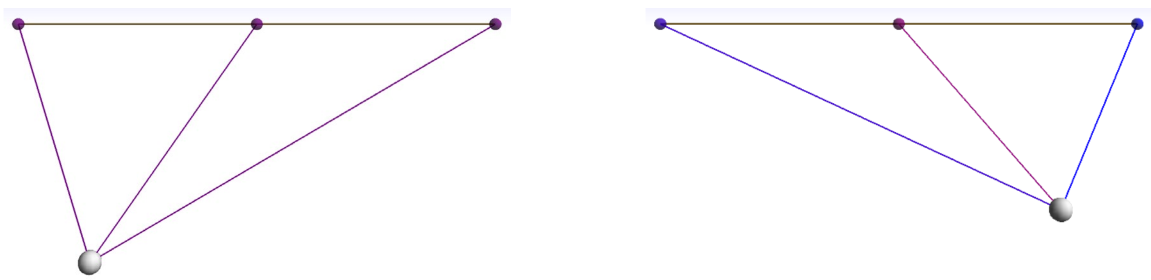
Keywords: neuromuscular control, neuromechanical simulation, reinforcement learning

Introduction

A detailed assessment of the human neuromuscular control can drastically improve the prediction of injury risks for a healthy athlete and the evaluation of the readiness to return to sports for an injured one. Furthermore, such a detailed assessment will bring us one step closer to a detailed digital twin (Barricelli et al. (2020)) of the human neuromuscular system and significantly improve the personalization and efficiency of neuromuscular training. Neuromechanical simulators (Seth et al. (2018)) are already capable of estimating the muscle activations necessary to reproduce a captured movement of an athlete on the athlete's musculoskeletal model. Unfortunately, trajectory optimization techniques, on which they rely, constrain their usage for simple movements and with simplified musculoskeletal models. These constraints limit the neuromechanical simulators from accurately modelling the neuromuscular system of an athlete. However, the recently developed RL algorithms show great potential for overcoming these limitations (Song et al. (2021)). They are already capable of reproducing complex captured movements on torque actuated skeletal models within physics simulators (Peng et al. (2022)). We envision to leverage their learning power, and employ them to reproduce the muscle activations necessary for the generation of the captured movements of an athlete on the athlete's personalized musculoskeletal model within a neuromechanical simulator. Since this is an ambitious endeavour, before approaching it at its real scale, it is essential to validate, at small scale, that RL algorithms are able to learn plausible muscle activations. In this context, the next two sections of this paper present the methods and results. The last section opens a discussion on the results, and points towards our future experiments on using RL algorithms for predictive simulations.

Methods

To validate that the RL algorithms estimate plausible muscle activations, we designed our experiment such that the *learned muscle activations* resulted are directly comparable with the *optimized muscle activations* of state-of-the-art Moco (Dembia et al. (2021)) optimizer.



(a) (b)

Fig. 1 The point mass in OpenSim's visualizer. (a) The point mass in its initial position; (b) The point mass in the desired position.

Neuromechanical simulators enable the research of the connections between the brain and the body while dynamically interacting with the world, and encompass computational models of different fidelity for each of them.

A ***mass suspended by three muscles*** depicted in (Fig. 1) is one of the first experiments performed with Moco to validate that the muscle activations which it optimizes are plausible. We reproduced the same simulation setup to validate that the RL algorithms learn muscle activations which resemble the ones optimized by Moco, using optimal control theory. The simulation setup consists of a point mass of 1Kg, mobile only on a vertical plane (two degrees of freedom: vertical and horizontal), suspended by three muscles, and under the influence of gravity. We want to find the activations which will command the three muscles to move the point mass from its initial position (Fig. 1 (a)) into its desired one (Fig. 1 (b)) with minimum energy, and within the defined time interval (0.4 Seconds).

The ***dynamic optimizer*** of Moco leverages the advantages of the direct collocation method used in trajectory optimization techniques, and automatically generates a nonlinear problem, which it solves using the IPOPT solver (Wächter et al. (2006)). For this simulation setup, Moco receives the initial and desired positions of the point mass, and the time interval, and outputs the *optimized muscle activations*.

Reinforcement Learning has its roots in the theory of animal learning and its core component is the learning from the interaction with the environment. During the last decade, the RL algorithms obtained multiple spectacular results mostly by leveraging the breakthroughs of Deep Learning (DL) algorithms, which became very successful at training very complex Artificial Neural Networks (ANNs). In contrast to the dynamic optimization algorithms which optimize for muscle activations relative to the provided costs and constraints, the RL algorithms, by repeated trial-and-error, learn to generate improved muscle activations. This generative capability is particularly valuable for our experiments.

The behaviour cloning (BC) (Ho. et al. (2016)) is one of the simplest RL algorithms which for two sets of examples, trains an ANN to map the set of contexts to the set of actions. The cartesian trajectory of the point mass obtained from the forward simulation of Moco's optimized activations, represents the set of contexts, and the optimized activations represent the actions. BC runs multiple forward simulations of the muscle activations which the ANN generates. It selects the ones which best resemble the optimized muscle activations, and uses them to retrain the ANN. After a few iterations, the ANN becomes increasingly better, and generates muscle activations which mirror with precision the optimized ones.

Results

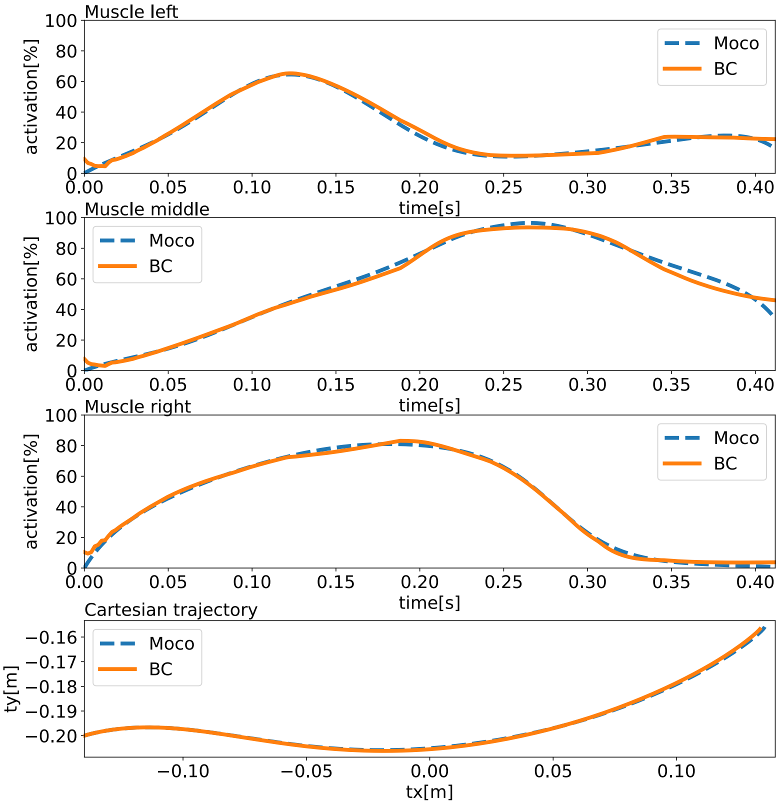


Fig. 2 The three topmost plots depict the optimized and learned muscle activations. The bottom most plot depicts the cartesian trajectories which the point mass follows when forward simulating the optimized and learned muscle activations.

The first three plots of (Fig. 2) present the activations of the three muscles over time. With blue we depict Moco's optimized muscle activations and with orange the BC' learned muscle activations. The fourth plot presents the two cartesian trajectories of the point mass. With blue we depict the trajectory obtained from forward simulating the optimized muscle activations and with orange is the one obtained from forward simulating the learned muscle activations. The first three plots share the same horizontal axis - the time axis. The horizontal axis of the fourth plot is aligned, and has the same scale with horizontal axis of the first three plots. It represents the horizontal cartesian coordinate of the position of the point mass. These preliminary results validate that RL can estimate plausible muscle activations because the *learned muscle activations* mirror very well the optimized ones, and, as consequence, the two cartesian trajectories are also very similar.
We run our simulation setup in OpenSim, which simulates the three muscles using the De Groote et al. (2016) implementation of the Hill's muscle model. The ANN is a multilayer perceptron with four sequential layers, each composed of 64 neurons with *hyperbolic tangent* as activation function.

Discussion

The generative capabilities of BC are limited because the ANN which it trains is unable to predict the muscle activations for a new cartesian position of the point mass which is not provided in the training sets. This is mainly due to the weak generalization capability of the

BC algorithm and the deterministic nature of the ANN. From this perspective, BC is very similar to the supervised learning algorithms. Nevertheless, Proximal Policy Optimization (PPO) (Schulman, et al. (2017)) is a better alternative. It trains a stochastic ANN, and hence has significantly increased generalization and generative powers. Our future reports will present the performance of PPO on learning the muscle activations, which the musculoskeletal model of an athlete's lower body needs in order to reproduce athlete's gait in OpenSim.

The learning power of the RL algorithms is still only minimally used. In addition to their capability of estimating physical quantities unmeasurable in biomechanics (Song et al. (2021)), RL algorithms can test complex models of the motor control (Merel et al. (2019).

Conflict of interest We declare no conflicts of interest.

Funding This work was funded by the Audi-Konfuzius Institut Ingolstadt.

References

Barricelli, B. R., Casiraghi, E., Gliozzo, J., Petrini, A., & Valtolina, S. (2020). Human digital twin for fitness management. IEEE Access, 8, 26637-26664.

De Groote, F., Kinney, A. L., Rao, A. V., & Fregly, B. J. (2016). Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. Annals of Biomedical Engineering.

Dembia, C. L., Bianco, N. A., Falisse, A., Hicks, J. L., & Delp, S. L. (2021, 12). Opensim moco: Musculoskeletal optimal control. PLOS Computational Biology, 16(12), 1-21.

Ho, J., Gupta, J. K., & Ermon, S. (2016). Model-free imitation learning with policy optimization. In Proceedings of the 33rd international conference on international conference on machine learning - volume 48 (p. 2760–2769). JMLR.org.

Merel, J., Botvinick, M., & Wayne, G. (2019). Hierarchical motor control in mammals and machines. Nature communications, 10(1), 5489

Peng, X. B., Guo, Y., Halper, L., Levine, S., & Fidler, S. (2022, jul). Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. ACM Trans. Graph., 41(4).

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv.

Seth, A., Hicks, J. L., Uchida, T. K., Habib, A., Dembia, C. L., Dunne, J. J., . . . Delp, S. L. (2018, 07). Opensim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. PLOS Computational Biology, 14(7), 1-20.

Song, S., Kidzi ´nski, , Peng, X. B., Ong, C., Hicks, J., Levine, S., . . . Delp, S. L. (2021, 08). Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. Journal of NeuroEngineering and Rehabilitation, 18(1).

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line search algorithm for large-scale nonlinear programming. Mathematical Programming, 106, 25-57.