https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa2-806704

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 system 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

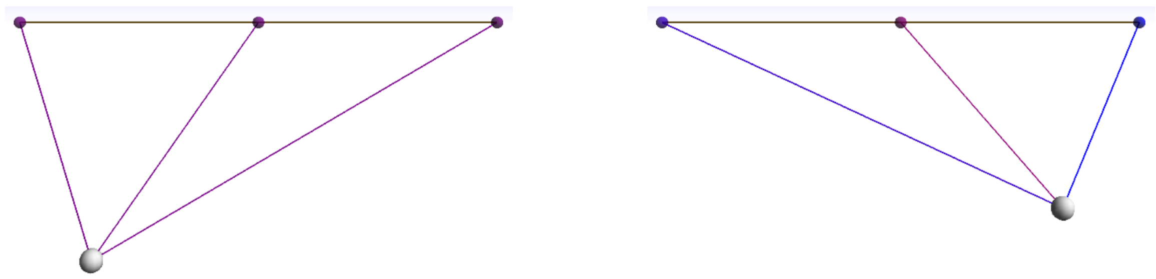
Introduction

A detailed assessment of the human neuromuscular system (NMS) 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 an accurate digital twin (Barricelli Barbara Rita et al., 2020) of the human NMS and significantly improve the personalization and efficiency of neuromuscular training. Neuromechanical simulators (Seth et al., 2018) already estimate the muscle activations of an athlete and reproduce a captured movement 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 NMS 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 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 endeavor, it is essential to validate at small scale that RL algorithms learn plausible muscle activations.

Methods

We designed our experiment that the learned muscle activations are directly comparable with the ones optimized by the state-of-the-art Moco optimizer (Dembia et al., 2021).

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



(a) (b)

Fig. 1 (a) The point mass in its initial position; (b) The point mass in its final position.

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 1 kg which moves only in a vertical plane, is suspended by three muscles, and is 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. 1a) to its desired one (Fig. 1b) with minimum energy and within the defined time interval (0.4 seconds). The simulation setup runs in OpenSim, which simulates the three muscles using the De Groote implementation of the Hill's muscle model.

The dynamic optimizer of Moco leverages the advantages of the direct collocation method used in trajectory optimization techniques, automatically generates a nonlinear problem which it solves using the IPOPT solver (Wächter & Biegler, 2006). For this simulation setup, Moco receives the initial and desired positions of the point mass, 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) ones which became very successful at training highly 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.

Behavioral cloning (BC) (Ho et al., 2016) is one of the simplest RL which trains an ANN to map the set of context to the one of actions provided as training examples. 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 represents the one of 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. The ANN is a multilayer perceptron with 4 sequential layers, each composed of 64 neurons with hyperbolic tangent as the activation function.

Results

The first three plots of Fig. 2 present the activations of the three muscles over time. The dashed lines depict Moco's optimized muscle activations, and the continuous ones depict the BC's learned ones. The fourth plot presents the two Cartesian trajectories of the point mass obtained from forward simulating the optimized and learned muscle activations.

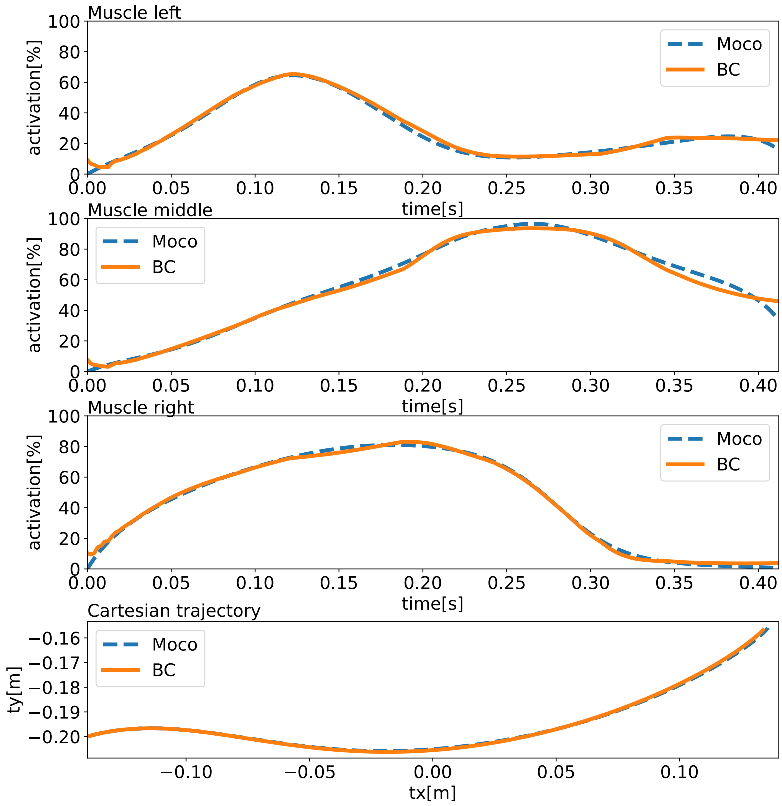


Fig. 2 The activations of the three muscles and the Cartesian trajectory of the point mass.

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 the 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 algorithms estimate plausible muscle activations because the learned muscle activations mirror the optimized ones very well and the two Cartesian trajectories are very similar.

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, RL algorithms can test complex models of the human 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, Ingolstadt, Germany.

References

Barricelli Barbara Rita, Casiraghi Elena, Gliozzo Jessica, Petrini Alessandro, & Valtolina Stefano (2020). Human Digital Twin for Fitness Management. *IEEE Access*, *8*, 26637–26664. https://doi.org/10.1109/ACCESS.2020.2971576

Dembia, C. L., Bianco, N. A., Falisse, A., Hicks, J. L., & Delp, S. L. (2021). OpenSim Moco: Musculoskeletal optimal control. *PLOS Computational Biology*, *16*(12), 1–21. https://doi.org/10.1371/journal.pcbi.1008493

Ho, J., Gupta, J. K., & Ermon, S. (2016). Model-Free Imitation Learning with Policy Optimization. Advance online publication. https://doi.org/10.48550/ARXIV.1605.08478

Merel, J., Botvinick, M., & Wayne, G. (2019). Hierarchical motor control in mammals and machines. *Nature Communications*, *10*(1), 5489. https://doi.org/10.1038/s41467-019-13239-6

Peng, X. B., Guo, Y., Halper, L., Levine, S., & Fidler, S. (2022). ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Trans. Graph.*, *41*(4). https://doi.org/10.1145/3528223.3530110

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., 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. (2018). OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLOS Computational Biology*, *14*(7), 1–20. https://doi.org/10.1371/journal.pcbi.1006223

Song, S., Kidziński, Ł., Peng, X. B., Ong, C., Hicks, J., Levine, S., Atkeson, C. G., & Delp, S. L. (2021). Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *Journal of NeuroEngineering and Rehabilitation*, *18*(1), 126. https://doi.org/10.1186/s12984-021-00919-y

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*(1), 25–57. https://doi.org/10.1007/s10107-004-0559-y