Rehabilitation of Musculoskeletal Models Using Deep Reinforcement Learning

Queralt Madorell[1], Albert Peiret[1], Josep M. Font-Llagunes[1]

[1] Dept. of Mechanical Engineering and Biomedical Engineering Research Centre
Universitat Politècnica de Catalunya, Diagonal 647, 08028, Barcelona, Spain
email1@address

EXTENDED ABSTRACT

1 Introduction

Neural rehabilitation is a long and complex process that patients undergo after suffering a nervous system injury, such as stroke. These kinds of injuries usually result in brain cells death and partial loss of mobility and coordination. During rehabilitation, the patient performs a series of movements and physical exercises that promote neural plasticity, the brain's mechanism to regenerate and make new pathways that substitute the damaged connections. Unfortunately, full recovery is almost impossible.

The rehabilitation process is tailored to the patient based on the physician's expertise, and it evolves with the patient's needs and recovery. However, few computational models for rehabilitation have been developed. For instance, Lee et al. [1] trained a musculoskeletal model of a healthy subject using deep reinforcement learning, and then a prosthetic leg was added to simulate an injury. Results showed how the artificial neural network that controlled muscle contraction was able to adapt and learn to move with the prosthetic leg.

Here we show how deep reinforcement learning can be used to control a musculoskeletal model. The algorithm is able to learn new and stable motions by maximizing the so-called reward function. The nervous system is modelled with an artificial neural network, and the deep deterministic policy gradient (DDPG) algorithm is used to train the model in a simulated environment.

2 Methods

The musculoskeletal model is a multibody systems with bones represented by rigid bodies and joints actuated by muscles. Kinematic constraints represent the joints and muscle contraction dynamics is modelled with a Hill-type muscle model. The neural excitation of each muscle controls muscle fiber contraction, which results in a force being applied to the bones. Motion control of musculoskeletal models presents some challenges, such as muscle redundancy, which is why a nonlinear optimization is required to solve the control problem.

Continuous control in deep reinforcement learning can be achieved with the DDPG algorithm, which is an actor-critic method. The actor network learns the optimal policy, while the critic network learns the action–value function. The policy is the function that calculates the control input (i.e., muscle neural excitation) in terms of the system state (i.e., joint angles, velocities, and accelerations, as well as muscle activation). The action–value function assesses how well the actor network performs with the action taken at a given time.

The algorithm learns the actions that maximize the reward function, which is why selecting a meaningful and effective reward function is paramount in reinforcement learning. The reward can be seen as the reciprocal of the cost. In biomechanics, the metabolic cost associated with muscle contraction can be represented by muscle fiber activation. Here, the reward function is expressed as

$$r = -\sum_{\text{joints}} (\theta_i - \theta_i^*)^2 - \sum_{\text{muscles}} w_a a_i^2 \tag{1}$$

where θ_i^* is the target value of the joint angle θ_i, and w_a is a weight associated with the muscle activation $a_i \in [0,1]$.

3 Results

A neural controller for a musculoskeletal model of the arm was trained using the DDPG algorithm to reach a target position with the hand. The 2-DoF model of the arm had 2 segments (upper-arm and forearm), and 8 muscles. The left side of Figure 1 shows the position of the model's hand on the sagittal plane as well as its joint angles used to shape the reward function. A total of 300.000 steps with random target positions of the ball were simulated to train the neural controller model using the DDPG algorithm. These steps were divided in episodes of 200 steps (0.01 seconds/step), which were the allotted time for the model to reach the ball. Once this termination condition was met, the ball's position was randomly reseted.

Once the model was trained, a scenario was designed to check the achieved results. The target configuration was set at 54° for θ_1^* and at 107° for θ_2^*, which was considered to require sufficient range of movement. The results are shown in the right side of Figure 1, where the absolute distance to the target point is shown.

These results show a little offset present in the X and Y coordinates, but overall the model has been successfully trained. It is able to reach the target in any position of its reach and it maintains the hand still in the target point for the entire duration of the episode.

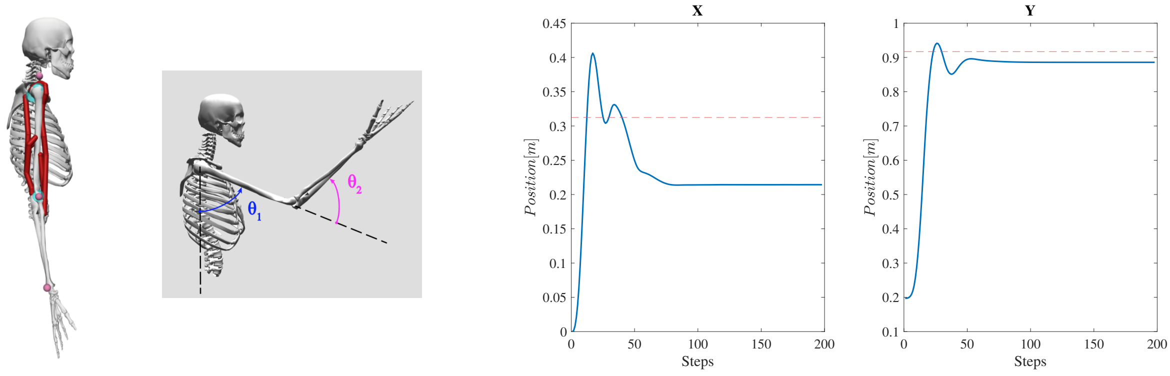

Figure 1: Motion generated using deep reinforcement using a target position of the hand.

4 Conclusions

The present results are promising and show the potential that reinforcement learning has in finding generalized solutions to biomechanical problems. The subsequent line of work that is currently being considered is the use of this reinforcement learning-based framework as a basis for testing different rehabilitation devices. This could allow the modelling of neuromuscular conditions and the assessment of the most suitable rehabilitation devices a treatments for a given case.

References

[1] Lee, S., Park, M., Lee, K. and Lee, J., 2019. "Scalable muscle-actuated human simulation and control," *ACM Transactions On Graphics (TOG)*, 38(4), pp.1-13.

[2] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. "Continuous control with deep reinforcement learning," 4th International Conference on Learning Representations, *ICLR 2016 - Conference Track Proceedings*, 2016.