

FES Control of a Human Arm Using Reinforcement Learning*

Philip S. Thomas,¹ Michael S. Branicky,¹ Antonie van den Bogert,² Kathleen Jagodnik²

1. Electrical Engineering and Computer Science Dept., Case Western Reserve University (CWRU)

2. Biomedical Engineering Depts. at both CWRU and Lerner Research Institute, Cleveland Clinic

I. INTRODUCTION

PEOPLE with spinal cord injury (SCI) are often unable to move their limbs, though most of their nerves and muscles may be intact. Functional Electrical Stimulation (FES) can activate these muscles to restore movement (Lynch & Popovic, 2008; Sujith, 2008; Ragnarsson, 2008; Sheffler & Chae, 2007; Peckham & Knutson, 2005).

Closed-loop control has been applied to FES tasks and can significantly improve performance compared to feed-forward control, and compensate for disturbances (Crago et al., 1996). In practice, such closed-loop controllers have to be manually tuned to each subject to overcome differences in dynamics from simulation, due to, e.g., muscle spasticity and atrophy. Closed-loop controllers are also unable to adapt to muscle fatigue during trials, which is frequent in FES patients.

Reinforcement learning (RL) techniques (Sutton & Barto, 1998) can be used to create controllers that adapt to changes in system dynamics. Within FES, RL has been tested in simulation to control standing up (Davoodi & Andrews, 1998), but this did not require generalization or a command input. RL has also been shown to control arm movements (Izawa et al., 2004), but learning required too many episodes for clinical applications.

Herein, we show the feasibility of using RL for FES control of upper extremities. While other closed-loop controllers (e.g., PD and PID) can partially compensate for changing dynamics, the RL controller outperforms them after training. In addition, we have shown that the actor-critic architecture can perform well, adapting to changing dynamics in a simulated human arm within 70 to 200 two-second episodes.

II. METHODS

Model. A computational model (Fig. 1) was used to test controllers in simulation. The arm moved in a horizontal plane without friction, had two joints (shoulder and elbow) and was driven by six muscles. Each muscle was modeled by a three-element Hill model and simulated using two differential equations, one for activation and one for contraction (McLean et al., 2003). Consequently, force is indirectly controlled via muscle dynamics. The internal muscle states (active state and contractile element length) were hidden and not available to the controller. We used a time step of 20ms.

* This work was supported in part by NIH Grant R21HD049662 and Predoctoral Fellowship F31HD049326 (Jagodnik). The authors also acknowledge and thank Dr. Robert Kirsch for his helpful input.

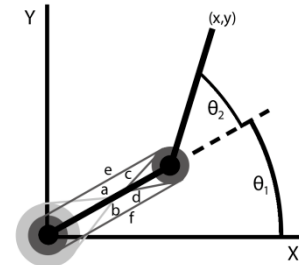


Fig. 1. Two-joint, six-muscle biomechanical arm model used.

RL Controller. We used the actor-critic architecture (Sutton & Barto, 1998), adapted for continuous time and space (Doya, 2000). The critic was implemented using an artificial neural network (ANN) with twenty neurons in its hidden layer and one neuron in its output layer, while the actor had ten neurons in its hidden layer and six in its output layer. For both, the neurons in the output layers used the identity threshold function, while the neurons in the hidden layers used the sigmoid threshold function, S . The actor-critic uses a 6x1 state vector x , given by

$$x(t) = [\bar{\theta}(t), \dot{\bar{\theta}}(t), \bar{\theta}_{\text{Goal}}(t)]^T. \quad (5)$$

At each time step, the 6x1 action vector of muscle stimulations $u(t)$ was computed using

$$u(t) = S(A(x(t); w) + \sigma \cdot n(t)), \quad (5)$$

where $A(x(t); w)$ is the actor ANN with weight vector w , σ is a noise scaling constant, and $n(t)$ is the 6x1 noise vector. The instantaneous reward function we use is

$$r(t) = -10^{-7} \sum_i F_i^2 - |\bar{\theta} - \bar{\theta}_{\text{Goal}}|^2, \quad (8)$$

where F_i is the muscle force of the i^{th} muscle, in Newtons.

Pre-Training. Before beginning unsupervised learning using the equations above, the actor-critic was pre-trained using a numerically optimized PD controller (Jagodnik & van den Bogert, 2007) as a supervisor. To do this, the actions for 550,000 training pairs and 170,000 testing pairs, each consisting of the state and corresponding action generated by the PD controller, were run through the inverse sigmoid giving training pairs for the actor ANN, $A(\bar{x}(t); w)$ from Eqn. 5. The actor ANN was then trained using the error backpropagation algorithm with a learning rate of .001 (Russell & Norvig, 1995). After 2,000 epochs, the actor converged to a policy qualitatively similar to the PD controller's policy. The critic ANN was then trained with the actor's policy fixed and noise removed from its actions. It was trained for 100,000 two second episodes with $\eta_c=1$, and $\kappa=1$.

For each episode, the start and goal were randomly selected movements with the sum of the squared difference in joint angles (in radians) between the initial and goal configurations being greater than .6.

Evaluation. To evaluate the actor-critic's performance, we use the average total reward over 256 fixed episodes involving large motions over the state space. For comparison throughout, the PD controller's evaluation is $-.18$, and the actor, after PD pre-training, had an evaluation of $-.21$.

Tests. Two tests were devised to judge the actor-critic's learning and adaptive capabilities. The first test was inspired by PD controller human trials in which the subject had spasticity of the biceps brachii, causing it to exert a constant low level of torque on both joints. This *Baseline Biceps Test* (BBT) involved adding 20% of the maximum stimulation to the stimulation requested by the controller in order to simulate this subject's condition. The second test, the *Fatigued Triceps Test* (FTT), simulates the effects of a muscle being severely weakened. In this test, the triceps stimulation used is 20% of the requested triceps stimulation. The actor-critic's performance on the two tests after pre-training, but before any further training, is shown in Fig. 2.

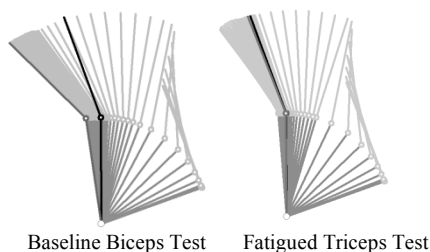


Fig. 2. Initial actor ANN's performance on a particular motion. The black state is the goal, and the medium grey one is the final state after 2s of simulation; the light grey states are snapshots taken every 20ms. The initial condition is the clockwise-most trace. In the BBT, the final state is the counterclockwise-most; in the FTT, it partially obscures the goal state.

III. RESULTS

The actor-critic's ability to improve the policy hinges on all of its learning parameters being properly set. For all tests we used $\Delta t = .02s$, and $\tau = 1s$, while η_A , η_C , τ_n , κ , and σ were varied. These learning parameters were optimized for the BBT, and their generalizability was tested using the FTT.

The parameters were optimized using Random-Restart Hill-Climbing search (Russell & Norvig, 1995), with the gradient sampled at 90% and 110% of the current value for each parameter. Of the 4,460 learning parameter sets examined, those in Table 1 were selected for further inspection due to their consistently good evaluations, as well as their different characteristics with respect to exploratory noise (not addressed here due to lack of space).

Fig. 3 shows the steady state moving closer to the goal configuration over time, as the actor-critic controllers learn on the BBT. The learning parameter sets' ability to adapt to changing dynamics was then tested using the FTT (Fig. 4).

Parameter set A steadily improved to the point where the arm does not overshoot the goal when starting clockwise of it after just 70 episodes (Fig. 4, left). Parameter set B learns slower: after 400 episodes it has reduced the overshoot (Fig. 4, right).

Table 1: Two of the best parameter sets found from optimization. Means and standard deviations were calculated with $N=30$.

Parameter Names	η_A	η_C	τ_n	κ	σ	Mean Evaluation	Std. Dev.
A	.001	.0001	.55	.55	74.5	-.267	.01
B	99.5	34.4	.25	71.5	7991	-.286	.09

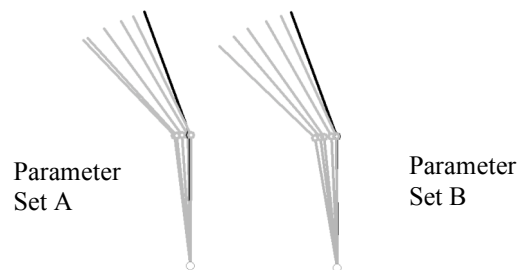


Fig. 3. Final states (grey) after training on the BBT for 1, 10, 50, 100, and 200 episodes (left to right), where the black state is the goal.

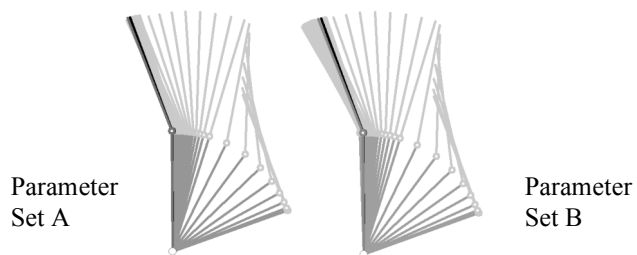


Fig. 4. Repeat of simulations from Fig. 2 after training on the FTT.

REFERENCES

- [1] Crago PE, Lan N, Veltink PH, Abbas JJ, Kantor C (1996) New control strategies for neuroprosthetic systems. *J Rehab Res Devel*, 33(2):158-72.
- [2] Davoodi R, Andrews JB (1998) Computer simulation of FES standing up in paraplegia: A self-adaptive fuzzy controller with reinforcement learning. *IEEE Trans Rehab Eng* 6(2):151-61.
- [3] Doya, K. (2000). Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219-45.
- [4] Izawa J, Toshiyuki K, Koji I (2004) Biological arm motion through reinforcement learning. *Biological Cybernetics*, 91(1):10-22.
- [5] Jagodnik KM, van den Bogert AJ (2007) A proportional derivative FES controller for planar arm movement. *12th Ann. Conf Int FES Soc*, Phila.
- [6] Lynch LC, Popovic RM (2008) Functional Electrical Stimulation: Closed-loop control of induced muscle contractions. *IEEE Control Systems Mag*, 28(2):40-50.
- [7] McLean SG, Su A, van den Bogert AJ (2003) Development and validation of a 3-D model to predict knee joint loading during dynamic movement. *J Biomech Eng*, 125(6):864-74.
- [8] Peckham PH, Knutson JS (2005) Functional electrical stimulation for neuromuscular applications. *Annu Rev Biomed Eng*, 7:327-60.
- [9] Ragnarsson KT (2008) Functional electrical stimulation after spinal cord injury: Current use, therapeutic effects and future directions. *Spinal Cord*, 46(4):255-74.
- [10] Russell S, Norvig P (1995) *Artificial Intelligence*, Prentice Hall. 2nd Ed.
- [11] Sheffler LR, Chae J (2007) Neuromuscular electrical stimulation in neurorehabilitation. *Muscle Nerve*, 35(5):562-90.
- [12] Sujith OK (2008) Functional electrical stimulation in neurological disorders. *Eur J Neurol*, 15(5):437-44.
- [13] Sutton R, Barto A (1998). *Reinforcement Learning*. MIT Press.