1-1-2008

Creating a Reinforcement Learning Controller for Functional Electrical Stimulation of a Human Arm

Philip S. Thomas
Case Western Reserve University

Michael Branicky

Antonie van den Bogert
Cleveland State University, a.vandenbogert@csuohio.edu

Kathleen Jagodnik

Published in final edited form as:
Yale Workshop Adapt Learn Syst. 2008 ; 049326: 1–6.

Creating a Reinforcement Learning Controller for Functional Electrical Stimulation of a Human Arm*

Philip S. Thomas[1], **Michael Branicky**[1], **Antonie van den Bogert**[2,3], and **Kathleen Jagodnik**[2,3]

[1] Department of Electrical Engineering and Computer Science, Case Western Reserve University

[2] Department of Biomedical Engineering, Case Western Reserve University

[3] Department of Biomedical Engineering, Lerner Research Institute, Cleveland Clinic

Abstract

Clinical tests have shown that the dynamics of a human arm, controlled using Functional Electrical Stimulation (FES), can vary significantly between and during trials. In this paper, we study the application of Reinforcement Learning to create a controller that can adapt to these changing dynamics of a human arm. Development and tests were done in simulation using a two-dimensional arm model and Hill-based muscle dynamics. An actor-critic architecture is used with artificial neural networks for both the actor and the critic. We begin by training it using a Proportional Derivative (PD) controller as a supervisor. We then make clinically relevant changes to the dynamics of the arm and test the actor-critic's ability to adapt without supervision in a reasonable number of episodes.

Index Terms

Functional Electrical Stimulation; Motor Control; Reinforcement Learning; Actor-Critic; Function Approximation

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. For background information on FES refer to (Sujith, 2008; Ragnarsson, 2008; Sheffler and Chae, 2007; Peckham and Knutson, 2005).

Open-loop control has been applied to FES systems including hand grasp (Peckham et al., 2001), rowing (Wheeler et al. 2002), and gait (Kobetic and Marsolais, 1994; Braz et al., 2007). The drawbacks to open-loop (feed-forward) control are that detailed information about the system's properties is required to produce accurate movements, and that poor movements can result if the properties of the system change (Crago et al., 1996).

Closed-loop control, which involves the use of sensors for feedback, has been applied to FES tasks such as hand grasp (Crago et al., 1991), knee joint position control (Chang et al., 1997), and standing up (Ferrarin et al., 2002). This form of control has the advantages that it can significantly improve performance as compared to feed-forward control, and it can compensate for disturbances (Crago et al., 1996). However, challenges related to using the

*This work was supported in part by NIH Grant R21HD049662 and Predoctoral Fellowship F31HD049326 (Jagodnik).

required sensors have largely prevented feedback control from being used in a clinical setting (Jaeger, 1992).

Other, more complex controllers, such as those combining feed-forward and feedback control (Stroeve, 1996) or adaptive feed-forward control (Abbas and Triolo, 1997) have been largely tested only in simulation or in simple human systems.

In practice, closed-loop controllers have been manually tuned to each subject to overcome differences in dynamics from simulation. These differences in dynamics can be significant due to muscle spasticity and atrophy. Closed-loop controllers are also unable to adapt to muscle fatigue during trials, which is frequent because muscle atrophy can create a higher proportion of fast-twitch muscle fibers which fatigue faster than slow-twitch fibers. Fatigue is also exacerbated because FES has a high stimulation frequency compared to a healthy central nervous system (Lynch and Popovic, 2008).

Reinforcement learning (RL) techniques (Sutton and Barto, 1998) can be used to create controllers that adapt to changes in system dynamics, such as those due to spasticity, atrophy, and fatigue, and can find non-obvious and efficient strategies. Within FES, RL has been tested in simulation to control a standing up movement (Davoodi and 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. In this paper we show the feasibility of using reinforcement learning for FES control of upper extremities as an improvement over previous closed-loop controllers that are unable to adapt to changing system dynamics.

The rest of the paper is organized as follows. We begin by considering static linear controllers in Section II. In Section III, we present the actor-critic framework used, the results of which are given in Section IV and discussed in Section V. Section VI contains overall conclusions and future work.

II. Static Linear Controllers

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. Two of the four muscles act across both joints. 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, muscle force was not directly controlled but indirectly via muscle dynamics. The internal muscle states (active state and contractile element length) were hidden and not available to the controller.

Jagodnik and van den Bogert (2007) have designed a Proportional Derivative (PD) controller for planar control of the arm of a paralyzed subject. The gains for the PD controller were tuned to minimize joint angle error and muscle forces for a two-dimensional arm simulation using a Hill-based muscle model (Schultz et al., 1991) with a time step of 20ms.

During human trials, Jagodnik and van den Bogert (2007) found that the PD controller's gain matrix often required retuning to account for changing dynamics in the subject's arm. The subject's arm differed significantly from the ideal arm used in simulation because it had baseline biceps stimulation due to spasticity. Results from simulation, which will be given later, support the claim that PD and PID controllers do not perform well with changing dynamics.

The output equation for the PD and PID controllers is

$$u = Gs, \tag{1}$$

where u is a 6×1 vector of muscle stimulations, s is the state vector, and G is a 6×4 gain matrix for the PD controller and a 6×6 gain matrix for the PID controller. For the PD controller, the state vector, s, is given by

$$s = [\vec{\theta}(t) - \vec{\theta}_{\text{Goal}}(t),\ \dot{\vec{\theta}}(t)]^{\text{T}} \tag{2}$$

for the PD controller, and

$$s = [\vec{\theta}(t) - \vec{\theta}_{\text{Goal}}(t),\ \dot{\vec{\theta}}(t),\ \int \vec{\theta}(\tau) - \vec{\theta}_{\text{Goal}}(\tau) d\tau]^{\text{T}} \tag{3}$$

for the PID controller, where $\vec{\theta}(t)$ is a vector of the shoulder and elbow joint angles, and $\vec{\theta}_{\text{Goal}}(t)$ contains the target joint angles. The integral error term was approximated using backward rectangular approximation.

We implemented a Proportional Integral Derivative (PID) controller to determine whether a more sophisticated closed-loop architecture could better cope with the changing dynamics of the arm. The gains were tuned using the Random-Restart Hill Climbing (RRHC) minimization algorithm (Russell and Norvig, 1995) using the same evaluation criteria as Jagodnik and van den Bogert (2007). For the random restarts, the proportional and derivative gains were taken from the PD controller, and the integral gains chosen randomly between −1 and 1. The gradient was sampled in steps 5% of each current gain value, with sign changes allowed as each weight approaches 0.

To test the PID's ability to adapt to changing dynamics, the arm model was modified to include a baseline biceps stimulation. The biceps muscle was given the PID's instructed stimulation to the biceps muscle plus an additional 20% (not to exceed 100%). This simulated the spasticity that was observed during human trials of the PD controller. When using the PID controller during a two-second episode with an initial state of shoulder joint angle θ_1=20°, elbow joint angle θ_2=90°, and a goal state of θ_1=90°, θ_2=20°, the arm overshoots the goal state.

Unlike the PD and PID controllers, an RL controller (described in the next section) could learn to not overshoot the goal position given unexpected muscle spasticity.

III. Reinforcement Learning Methods

We chose to use the actor-critic architecture (Sutton and Barto, 1998) because of its ability to reduce the dimensionality of the problem by half, as opposed to other temporal difference (TD) learning architectures. Because we are working in continuous time and space, we selected the continuous actor-critic (Doya, 2000), which is reviewed in this section.

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

$$S(z)=\frac{1}{1+e^{-z}}. \tag{4}$$

The actor-critic uses a 6×1 state vector x, given by

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

At each time step, the 6×1 action vector of muscle stimulations $u(t)$ was computed using

$$u(t)=S\left(A(x(t);w)+\sigma\cdot n(t)\right), \tag{6}$$

where $A(x(t); w)$ is the actor ANN with weight vector w, σ is a noise scaling constant, and $n(t)$ is the 6×1 noise vector given by

$$\dot{n}(t)=\frac{-n(t)+N(t)}{\tau_n}, \tag{7}$$

where $N(t)$ is normal Gaussian noise and τ_n is another noise scaling constant. The noise is initialized to 0: $n(0) = 0$.

The resulting TD error was computed using a backward Euler approximation given by

$$\delta(t)=r(t)+\frac{1}{\Delta t}\left[\left(1-\frac{\Delta t}{\tau}\right)V(t)-V(t-\Delta t)\right], \tag{8}$$

where Δt is the discrete time step for learning updates, τ is the time constant for discounting future rewards, $V(t)$ is the critic's estimate of the value of the state at time t and $r(t)$ is the instantaneous reward given by

$$r(t)=-10^{-7}\sum_i F_i^2-\left|\vec{\theta}-\vec{\theta}_{\text{Goal}}\right|^2, \tag{9}$$

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

The weights for the critic ANN were then updated using

$$\dot{w}_i=\eta_c\delta(t)e_i(t), \tag{10}$$

where η_C is the learning rate and $e_i(t)$ is the eligibility trace for the corresponding weight, given by

$$\dot{e}_i(t) = -\frac{1}{\kappa} e_i(t) + \frac{1}{\kappa} \frac{\partial V(x(t);w)}{\partial w_i}, \tag{11}$$

where κ is a time constant. Finally, each weight in the actor ANN is updated using

$$\dot{w}_i = \eta_A \delta(t) n(t) \cdot \frac{\partial A(\vec{x}(t);w)}{\partial w_i}, \tag{12}$$

where η_A is a learning rate. Note the dot product between the noise and the derivative of the actor ANN with respect to each weight. To ensure stability in both the actor and the critic while allowing for larger learning rates, the magnitude of the TD error, $\delta(t)$, was capped at 10.

Pre-Training

Before beginning unsupervised learning using the equations above, the actor-critic was pre-trained using the PD controller 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, $A(\vec{x}(t); w)$ from Eqn. 6. The actor ANN was then trained using the error backpropagation algorithm with a learning rate of .001 (Russell and Norvig, 1995). After 2,000 epochs, each of which consisted of training once on each of the 550,000 training points, the actor converged to a policy qualitatively similar to the PD controller's policy.

The critic ANN was then trained using the full actor-critic with the previously trained actor. The actor's policy was fixed, and noise removed from its actions while the critic was brought on-policy. It was trained for 100,000 two second episodes with η_C=1, and κ=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. This constraint removed episodes in which the arm does not have to make a significant motion. All future training was done with the same episode duration and constraints.

The actor-critic thus begins with an actor ANN that is a close approximation of the PD controller, and an on-policy critic if there is no discrete change to the arm dynamics. The latter is important because the actor-critic can diverge until the critic is on-policy. When the arm dynamics change, the critic will not be on-policy, but will hopefully reconverge quickly.

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 pre-training on the PD controller, has an evaluation of −.21.

Three tests were devised to judge the actor-critic's learning and adaptive capabilities. The first was a control test, where the dynamics of the arm were not changed, but the actor-critic was allowed to continue to learn.

The second 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. When using the PD controller or the actor-critic trained on it, the steady state of the arm is counterclockwise of the goal state at the point where the controller's requested triceps stimulation balances out the baseline biceps stimulation. The actor-critic's evaluation on the BBT is −.65 immediately after pre-training (i.e., before further learning).

The third 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. Thus, when a controller requests full triceps stimulation, only 20% will be given. Unlike the BBT, this does not change the steady state when using the PD controller, though it does induce overshoot if the initial configuration is clockwise of the goal. This occurs because the biceps are used to pull the arm towards the goal, and the triceps are used to stop it at the goal configuration. With the triceps weakened, the PD controller does not exert enough torque to overcome the arm's angular momentum. The actor-critic's evaluation on the FTT immediately after pre-training is −.22.

The actor-critic's ability to improve the policy hinges on all of its learning parameters being properly set. For all tests we used Δt =.02s, and τ=1s, while η_A, η_C, τ_n, κ, and σ were varied. Thus, the learning rates, exploratory noise, and decay rate of eligibility traces were varied to find those that are most suitable for adapting to changing dynamics in the system. These learning parameters were optimized for the BBT, and their generalizability was tested using the FTT.

The parameters were again optimized using RRHC search (cf. Sec. II), with the gradient sampled at 90% and 110% of the current value for each learning parameter. Each parameter set's learning abilities were measured as the average evaluation after 100, 200, 500, and 1000 random training episodes. Again, only interesting episodes were allowed, in which the squared difference in joint angles between the initial and goal configurations was greater than .6. Random restarts used a logarithmic distribution half the time, and a linear distribution the other half of the time in order to better explore the extremes and full range of the parameter space.

The actor-critic's performance on the three tests after pre-training, but before any further training, is shown in Fig. 2.

IV. Results

Of the 4,460 learning parameter sets examined by the RRHC search, 1,363 had evaluations higher than −.3. However, many of the best learning parameter sets found by the optimization did not have stable evaluations. For example, the best parameter set received an evaluation of −.22 during the optimization, though further tests found their average evaluation was −.33 with a standard deviation of .15 (N=100). The parameter sets 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, which will be addressed later.

Control Test

The first test was the control test, in which the arm model was not modified, and the actor-critic was allowed to further adapt to the standard arm model. Both parameter sets improved objectively upon the pre-trained policy, making faster movements to the goal configuration with less oscillation before reaching stability. Neither achieved the same reward as the PD controller itself (−.18) within 400 episodes of training, as shown in Fig. 3.

It was also observed that after training for thousands of episodes, the actor-critic controller, with the current parameter sets and reward system, becomes unstable. The muscle stimulations become erratic and the arm begins shaking. Eventually, the policy falls apart completely and the arm flails. Increasing the weighting of the muscle forces in the reward (Eqn. 9) was found to decrease and postpone this jitter.

Baseline Biceps Test

Because the learning parameter sets were optimized using the BBT, they both perform well on the BBT, quickly removing overshoot of the goal when the initial configuration is clockwise of the goal configuration, and generating a steady state close to the goal state. Fig. 4 shows the steady state moving closer to the goal configuration over time, as the actor-critic controllers learn.

Fig. 5 shows the actor-critics' policy evaluations after each episode of training on the BBT, when using parameter sets A and B given in Table 1. For reference, an evaluation of −.21 is equivalent to the actor-critic's performance on the unmodified arm model after pre-training on the PD controller.

Fatigued Triceps Test

The learning parameter sets' ability to adapt to changing dynamics was then tested using the FTT. Because the parameters were optimized using the BBT, the FTT serves as a test of their generalizability to other changes in dynamics. Parameter set A did better than parameter set B on this test, steadily improving to the point where the arm does not overshoot the goal when starting clockwise of it, after 70 episodes (Fig. 6, left). Parameter set B learns slower, such that after 400 episodes it has reduced the overshoot, but it is still present (Fig. 6, right).

Fig. 7 shows the actor-critic's policy evaluations after each episode of training on the FTT when using parameter sets A and B given in Table 1. For reference, an evaluation of −.21 is equivalent to the actor-critic's performance on the unmodified arm model after pre-training on the PD controller.

V. Discussion

In order to be practical for subjects with SCI, the learning agent must be able to adapt to the changing dynamics in a reasonable amount of time. On the BBT, the actor-critic adapted to a significant and discrete change in dynamics in fewer than 200 episodes. This change in arm properties was similar to the expected change when adapting to a new subject's arm. On the FTT, the actor-critic adapted to a similar discrete change representing a fatigued arm in fewer than 70 episodes.

Learning parameter sets A and B were chosen because they exemplify how different parameters are capable of learning in the simulated environment. Parameter set A has a massive amount of noise, flopping the arm around during training trials to explore the state and action spaces, while parameter set B exploits current knowledge, with subtle exploratory

noise injected into the policy. In a typical episode in the control test, the average sum of the squared joint angle noise for parameter set A was four orders of magnitude larger than that of parameter set B.

The ability of the RL system to learn equally well with various sets of learning parameters on the simulated arm is encouraging and potentially useful in clinical applications. When used with a human arm, there will be unintentional noise introduced to the system. Parameters ought to be chosen which have just enough noise that the agent can distinguish between the intended exploratory noise and the undetectable noise inherent in real-world experiments. With too much noise, however, the exploratory actions would interfere with the desired movement or even cause injury.

We also performed some experiments using different function approximators to represent the actor and critic. Each function approximator was trained using 550,000 training points, and tested using 170,000 different testing points. The points consist of state and utility pairs, computed in simulation using the PD controller as the actor. Fig. 9 shows the results for learning the critic's utility function. (Policy approximation performance was similar among all function approximators.)

Locally weighted linear regression (LWR) (Schaal et al., 2002) achieved a total squared error one tenth of that achieved by an ANN with 20 neurons in its hidden layer, trained using error backpropagation on the same training set. If converted to a learning algorithm in which one point in the knowledge base is replaced at every 20ms update, the entire knowledge base would be replaced after every three hours of use. Functional link nets (FLNs), using kernel functions derived from the equations of motion, were found to have little improvement over ANNs. *K*-Nearest Neighbor (*K*-NN), though a simple algorithm, was also found to perform better than the ANNs.

The ANN and FLN had 20 neurons in their hidden layers, a learning rate of 10^{-6}, and were trained for 550 epochs. *K*-NN performed best with $K=9$, using a squared-inverse distance weighting metric. LWR performed best using a neighborhood containing the 20 nearest points, using a weighting scale parameter h (Schaal et al., 2002) on the order of 10^{-5}.

VI. Conclusions and Future Work

We have examined reinforcement learning's application to FES control of the upper extremity. In particular, 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. While other closed-loop controllers (e.g., PD and PID) can partially compensate for changing dynamics, the reinforcement learning controller outperforms them after training. We also found that the actor-critic was capable of learning with varying amounts of exploratory noise, which will be necessary when training the actor-critic in a noisy environment.

As this is one of the first attempts known by the authors to apply reinforcement learning techniques to FES, the research area is still open for significant development. Human trials of the actor-critic controller presented in this paper could give further insight into the real world implementation issues.

A similar reinforcement learning agent could be applied to a model of the human arm that allows for full three-dimensional motion. This would bring the field closer to the long-term goal of restoring motor function to people with SCI.

Acknowledgments

The authors thank Dr. Robert Kirsch for his helpful input.

References

1. Abbas JJ, Triolo RJ. Experimental evaluation of an adaptive feedforward controller for use in functional neuromuscular stimulation systems. IEEE Trans Rehab Eng. 1997; 5(1):12–22.
2. Braz GP, Russold M, Smith RM, Davis GM. Electrically-evoked control of the swinging leg after spinal cord injury: Open-loop or motion sensor-assisted control? Australas Phys Eng Sci Med. 2007; 30(4):317–23. [PubMed: 18274072]
3. Chang GC, Luh JJ, Liao GD, Lai JS, Cheng CK, Kuo BL, Kuo TS. A neuro-control system for the knee joint position control with quadriceps stimulation. IEEE Trans Rehabil Eng. 1997; 5(1):2–11. [PubMed: 9086380]
4. Crago PE, Lan N, Veltink PH, Abbas JJ, Kantor C. New control strategies for neuroprosthetic systems. J Rehab Res Devel. 1996; 33(2):158–72.
5. Crago PE, Nakai RJ, Chizeck HJ. Feedback regulation of hand grasp opening and contact force during stimulation of paralyzed muscle. IEEE Trans Biomed Eng. 1991; 38(1):17–28. [PubMed: 2026428]
6. Davoodi R, Andrews JB. Computer simulation of FES standing up in paraplegia: A self-adaptive fuzzy controller with reinforcement Learning. IEEE Trans Rehab Eng. 1998; 6(2):151–61.
7. Doya K. Reinforcement learning in continuous time and space. Neural Computation. 2000; 12(1): 219–45. [PubMed: 10636940]
8. Ferrarin M, Pavan EE, Spadone R, Cardini R, Frigo C. Standing-up exerciser based on functional electrical stimulation and body weight relief. Med Biol Eng Comput. 2002; 40(3):282–9. [PubMed: 12195974]
9. Izawa J, Toshiyuki K, Koji I. Biological arm motion through reinforcement learning. Biological Cybernetics. 2004; 91(1):10–22. [PubMed: 15309543]
10. Jaeger RJ. Lower extremity applications of functional neuromuscular stimulation. Assist Technol. 1992; 4(1):19–30. [PubMed: 10148013]
11. Jagodnik, KM.; van den Bogert, AJ. A proportional derivative FES controller for planar arm movement. 12th Ann. Conf Int FES Soc; Phila. 2007.
12. Kobetic R, Marsolais EB. Synthesis of paraplegic gait with multi-channel functional neuromuscular stimulation. IEEE Trans Rehab Eng. 1994; 2:66–79.
13. Lynch LC, Popovic RM. Functional Electrical Stimulation: Closed-loop control of induced muscle contractions. IEEE Control Systems Mag. 2008; 28(2):40–50.
14. McLean SG, Su A, van den Bogert AJ. Development and validation of a 3-D model to predict knee joint loading during dynamic movement. J Biomech Eng. 2003; 125(6):864–74. [PubMed: 14986412]
15. Peckham PH, Keith MW, Kilgore KL, Grill JH, Wuolle KS, et al. Efficacy of an implanted neuroprosthesis for restoring hand grasp in tetraplegia: A multicenter study. Arch Phys Med Rehabil. 2001; 82:1380–8. [PubMed: 11588741]
16. Peckham PH, Knutson JS. Functional electrical stimulation for neuromuscular applications. Annu Rev Biomed Eng. 2005; 7:327–60. [PubMed: 16004574]
17. Ragnarsson KT. Functional electrical stimulation after spinal cord injury: Current use, therapeutic effects and future directions. Spinal Cord. 2008; 46(4):255–74. [PubMed: 17846639]
18. Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach. 2. Englewood Cliffs, NJ: Prentice Hall; 1995.
19. Schaal S, Atkeson CG, Vijayakumar S. Scalable techniques from nonparametric statistics for real time robot learning. Applied Intelligence. 2002; 17:49–60.
20. Schultz AB, Faulkner JA, Kadhiresan VA. A simple Hill element-nonlinear spring model of muscle contraction biomechanics. J Appl Physiol. 1991; 70(2):803–12. [PubMed: 2022572]
21. Sheffler LR, Chae J. Neuromuscular electrical stimulation in neurorehabilitation. Muscle Nerve. 2007; 35(5):562–90. [PubMed: 17299744]

22. Stroeve S. Learning combined feedback and feedforward control of a musculoskeletal system. Biol Cybern. 1996; 75(1):73–83. [PubMed: 8765656]

23. Sujith OK. Functional electrical stimulation in neurological disorders. Eur J Neurol. 2008; 15(5): 437–44. [PubMed: 18394046]

24. Sutton, R.; Barto, A. Reinforcement Learning. Cambridge: MIT Press; 1998.

25. Wheeler GD, Andrews B, Lederer R, Davoodi R, Natho K, Weiss C, Jeon J, Bhambhani Y, Steadward RD. Functional electrical stimulation-assisted rowing: Increasing cardiovascular fitness through functional electrical stimulation rowing training in persons with spinal cord injury. Arch Med Phys Rehabil. 2002; 83(8):1093–9.

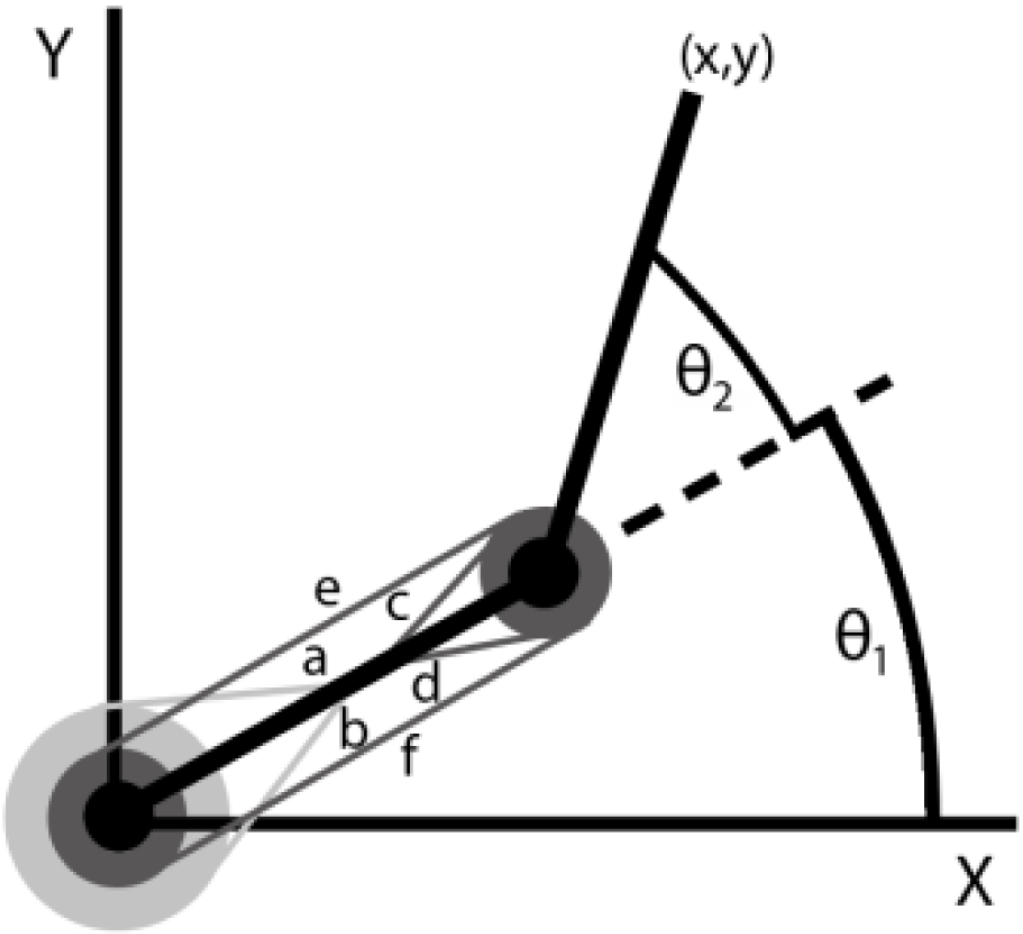

Fig. 1.

Two-joint, six-muscle biomechanical arm model used. Antagonistic muscle pairs are as follows, listed as (flexor, extensor): monoarticular shoulder muscles (a: anterior deltoid, b: posterior deltoid); monoarticular elbow muscles (c: brachialis, d: triceps brachii (short head)); biarticular muscles (e: biceps brachii, f: triceps brachii (long head)).

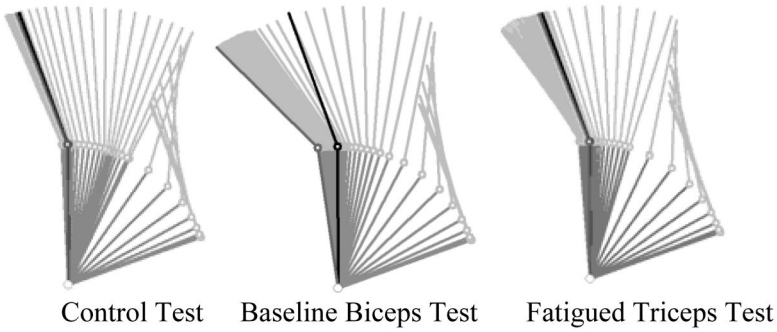

Fig. 2.

Initial actor ANN's performance on a particular motion for the three tests. The black state is the goal state (90°, 20°), the medium grey state is the final state after two seconds of simulation, and the light grey states are snapshots of the arm location taken every 20ms. The initial condition is the clockwise-most trace (20°, 90°). In the BBT, the final state is the counterclockwise-most trace, while in the control test and FTT the final state partially obscures the goal state.

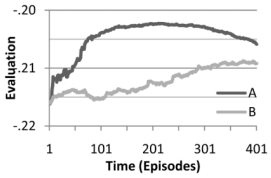

Fig. 3.
Graph of the actor-critic's evaluation over time, in episodes, using learning parameter sets A and B on the control test.

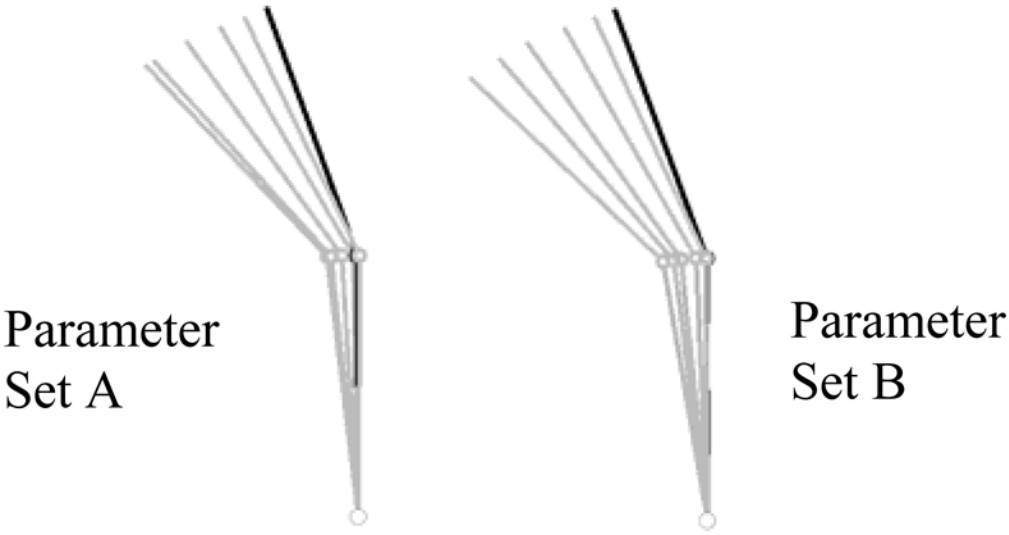

Fig. 4.

Final states (grey) after training for 1, 10, 50, 100, and 200 episodes (left to right), where the black state is the goal. The plot on the left uses learning parameter set A, and the plot on the right uses learning parameter set B.

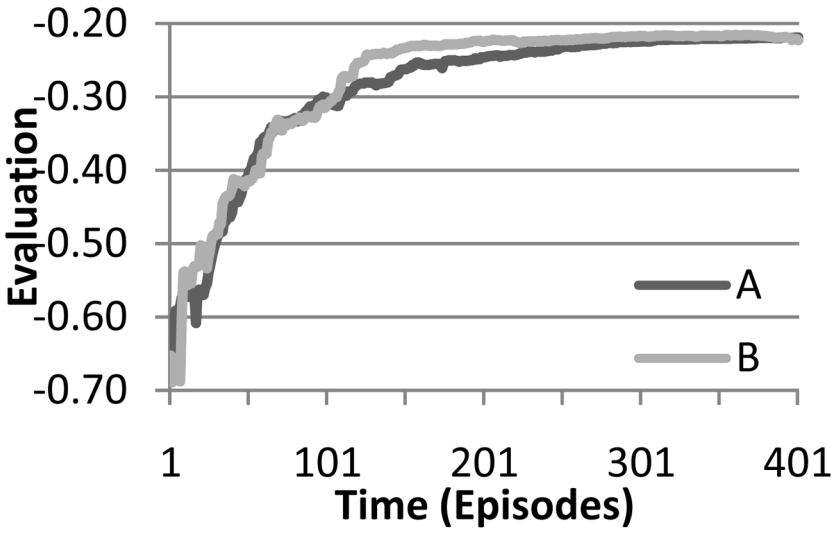

Fig. 5.
Graph of the actor-critic's evaluation over time, in episodes, using learning parameter sets A and B on the BBT.

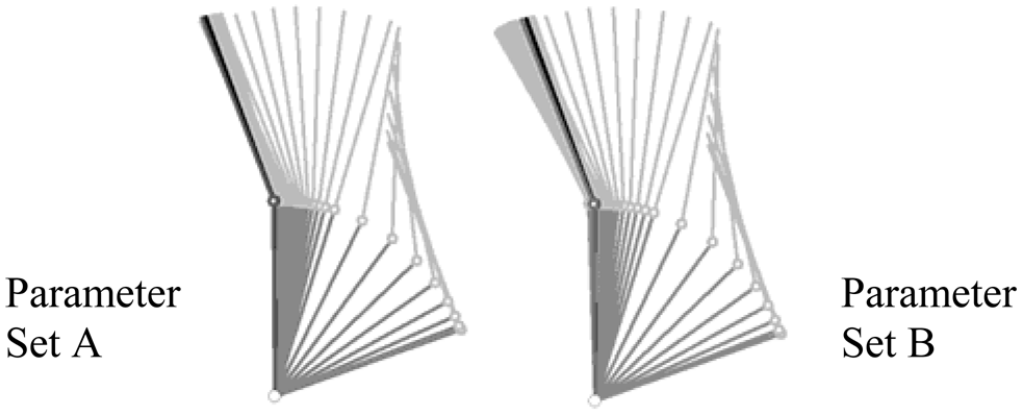

Fig. 6.

Repeat of simulations from Fig. 2 after training. Arm trajectories on FTT using learning parameter set A after 70 training episodes (left), and using learning parameter set B after 400 training episodes (right).

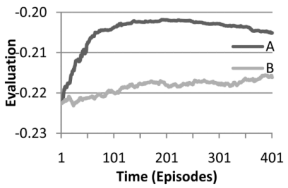

Fig. 7.
Graph of the actor-critic's evaluation over time, in episodes, using paramer sets A and B on the FTT.

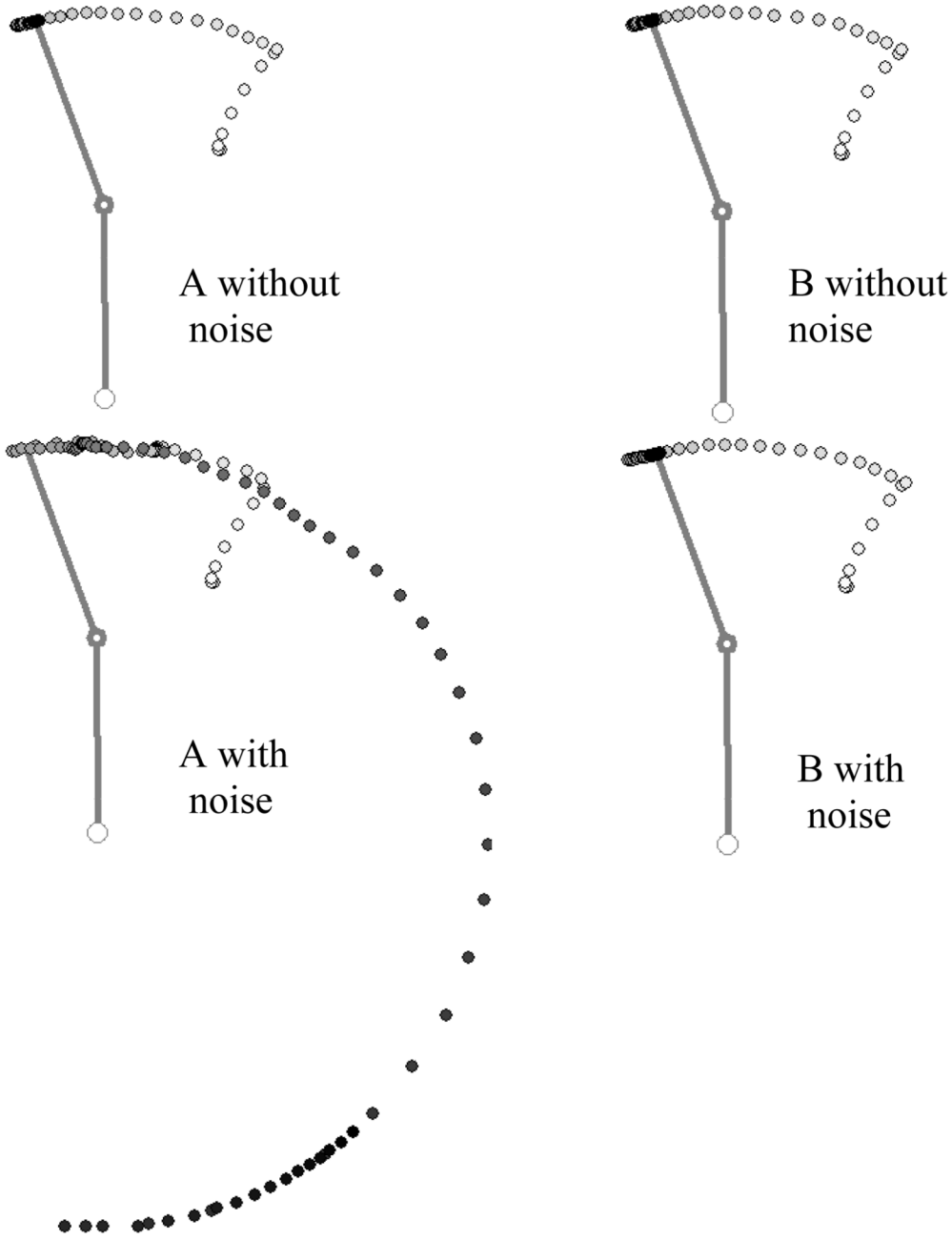

Fig. 8.

Plot of the hand position when using learning parameter set A without noise (top left), B without noise (top right), A with noise (bottom left), B with noise (bottom right). All are attempting the same motion to the grey goal state. Dots, starting white and fading to black, map the endpoint position every 20ms.

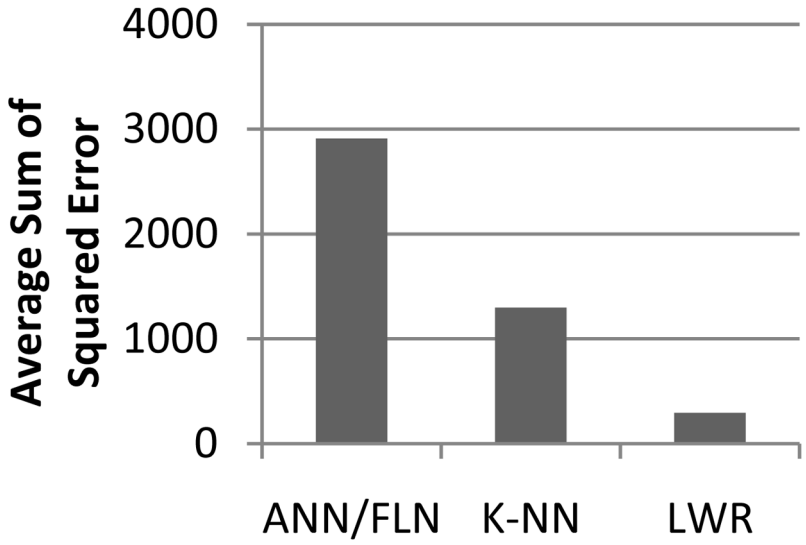

Fig. 9.
Plot of the sum of the squared error in approximating the critic's utility function for an actor-critic with the PD controller as the actor in the simulated arm environment.

Table 1

Two of the best parameter sets found from optimization. Means and standard deviations of the evaluations were calculated with a sample size of 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	2500	71.5	7991	−.286	.09

Yale Workshop Adapt Learn Syst. Author manuscript; available in PMC 2011 November 10.