1-1-2009

Application of the Actor-Critic Architecture to Functional Electrical Stimulation Control of a Human Arm

Philip 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:
Proc Innov Appl Artif Intell Conf. 2009 ; 2009: 165–172.

Application of the Actor-Critic Architecture to Functional Electrical Stimulation Control of a Human Arm

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

Philip Thomas: pst5@case.edu; Michael Branicky: mb@case.edu; Antonie van den Bogert: bogerta@ccf.org; Kathleen Jagodnik: kmj10@case.edu

[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 Foundation 10900 Euclid Ave., Glennan 517B, Cleveland, OH 44106-7071 U.S.A

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 the actor-critic architecture, with neural networks for the both the actor and the critic, as a controller that can adapt to these changing dynamics of a human arm. Development and tests were done in simulation using a planar arm model and Hill-based muscle dynamics. 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. Finally, we devise methods for achieving both rapid learning and long-term stability.

Keywords

Continuous actor-critic; stability; robustness; reinforcement learning; adaptive controller; functional electrical stimulation; human arm; artificial neural network; proportional derivative controller; proportional integral derivative controller; locally weighted regression

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 by activating motor neurons with electrical currents, which are applied via subcutaneous probes. By intelligently selecting the current given to the motor neurons associated with each muscle, individual muscles can be stimulated by various amounts, allowing researchers to control a subject's muscles. 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 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, basic closed-loop controllers have been manually tuned to each subject to overcome significant differences in dynamics from simulation, often due to muscle spasticity and atrophy. Traditional closed-loop controllers, such as those described in the following section, 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 these changes in system dynamics, finding 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 prior work by the authors, RL was used to adapt to changing dynamics in a simulated arm, though the resulting controller used impractical exploration, was not stable, and was not shown to be robust to sensor noise (Thomas et al. 2008).

In this paper, we extend our prior work and try to design a stable and robust controller based on RL that can quickly adapt to various realistic changes in arm dynamics, which would otherwise cause significant loss of performance. The approach chosen was to first train the agent to approximate the PD controller described in the following section, giving it a near optimal policy. Next we found parameters for the RL system that perform well on a specific real-world adaptation problem, the *Baseline Biceps Test* (BBT). The resulting parameters of the optimization were then tested on other relevant adaption and robustness tests: the *Control Test* (CT), *Fatigued Triceps Test* (FTT), and the *Noise Robustness Test* (NRT). In all cases, speed of learning and long term stability were evaluated. Finally, a hybrid RL controller was devised that achieves both rapid initial learning and long-term stability.

Static Linear Controllers

A computational model (Figure 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 and G is a 6×4 gain matrix for the PD controller and a 6×6 gain matrix for the PID controller. The error 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 of 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 by .216 radians for the shoulder angle, and .231 radians on the elbow angle, which equates to an overshoot of 23cm for a typical arm. Unlike the PD and PID controllers, the RL controller described in the next section learns to avoid overshooting the goal position given unexpected muscle spasticity.

Retuning of static linear controllers could restore performance, but would require extensive trial and error experimentation to find the optimal controller. Such a design process would not scale well to systems with more muscles and more joints, especially considering that this must be done on a patient. We therefore decided to consider RL as a method for adaptive control. RL learns online from experience and exploration and allows us to shape the reward signal such that its time integral corresponds with the chosen optimality criterion.

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 as opposed to other temporal difference (TD) learning architectures. For a problem involving an *m*-dimensional state space and an *n*-dimensional action space, state-action based agents, such as Q-learning agents, must compute a function $f: \mathbb{R}^m \times \mathbb{R}^n \rightarrow 1$. In the actor-critic architecture, this problem is reduced to two lower-dimensional problems: learning the value function $f: \mathbb{R}^m \rightarrow 1$ and learning the policy $f: \mathbb{R}^m \rightarrow \mathbb{R}^n$. This explicit representation of the policy also avoids the problem of finding the optimal action given the Q function, which can be difficult when working in continuous space with an infinite number of possible actions. With these considerations, we selected the continuous actor-critic (Doya 2000), reviewed below.

The actor and critic were implemented using artificial neural networks (ANNs) with ten neurons in their hidden layers and one neuron in their output layers. Experiments with varying numbers of neurons had similar results. 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

$$r(t)=W\sum_i u_i^2-|\vec{\theta}-\vec{\theta}_{\text{Goal}}|^2, \tag{9}$$

where u_i is the stimulation of the i^{th} muscle and $W = .016$, a value that was empirically found to generate desirable behavior in which position error and effort were appropriately balanced. This signal is nearly identical to that used to train the PID controller and PD controller (Jagodnik and van den Bogert 2007), except it uses muscle stimulations rather than muscle forces. This change was made because muscle forces are not directly observable in practice.

The weights for the critic ANN were then updated using

$$\dot{w}_i=\eta_C\delta(t)e_i(t)-\eta_C k_C w_i, \tag{10}$$

where η_C is the learning rate, k_C is a weight decay constant, 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 and $0 < \kappa \leq \tau$. Finally, each weight in the actor ANN is updated using

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

where η_A is a learning rate and k_A is a weight decay constant. 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 .5.

Pre-training

Before beginning reinforcement 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 points and 170,000 testing points, each consisting of the state and corresponding action

generated by the PD controller, were run through the inverse sigmoid, generating training points for the actor ANN, $A(\vec{x}(t); w)$ from Equation 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 while the critic was brought on-policy. For each two second 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 did 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. When the arm dynamics change, the critic will not be on-policy, but will reconverge quickly.

Evaluation

To evaluate actor-critic performance, we use a backward Euler approximation of the integral of the reward signal, averaged over 256 fixed episodes involving large motions over the state space. The larger the evaluation, the better, though because all rewards are negative, the evaluations will always be negative. 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. These numbers represent smooth, fast, and efficient movements, as judged from inspecting the movements and muscle forces generated during these tests.

Four tests were devised to judge the actor-critic's learning and adaptive capabilities for medically relevant changes in the system. The first was a control test, where the dynamics of the arm were not changed, allowing the actor-critic to further adapt to the standard arm model.

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 the condition of the subject used in PD controller tests. In the BBT, 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 (long head) 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 is used to pull the arm towards the goal, and the triceps is 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 fourth test, the *Noise Robustness Test* (NRT), adds sensor noise to the model to test the robustness of the controller on the BBT. Standard normal Gaussian noise was added to both the joint angle measurements, $\vec{\theta}(t)$, and the joint angle velocity measurements, $\dot{\vec{\theta}}(t)$, scaled by the constants σ_θ and $\sigma_{\dot{\theta}}$ respectively. Realistic values for these two parameters are $\sigma_\theta < .1$ and $\sigma_{\dot{\theta}} < .3$. For all tests, $\Delta t = .02$s.

The actor-critic's ability to improve the policy on each test hinges on all of its learning parameters being properly set. The six learning parameters, τ, τ_n, κ, σ, η_A and η_C were optimized via RRHC search for the BBT, and their generalizability was tested using the FTT.

The RRHC search sampled the gradient of the performance by evaluating it 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.

Figure 2 shows performance on the three tests after pre-training, but before any further training.

Test 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 values τ=.1s, τ_n=2400, κ=.1, and σ=9000, were found to work best while providing realistic exploration. The learning rates in Table 1 were selected for further evaluation. They are manually tuned parameters similar to those from RRHC, which gave consistently good evaluations.

The slow parameters represent slow and stable on-policy learning, while the fast parameters represent rapid initial learning using the shape of the pre-trained critic. Because we do not want adaptation to slow or stop, the learning rates are not decayed.

Control Test

Using the fast parameters on the control test, the system initially improves its evaluation, before becoming unstable. Qualitatively, the arm movements begin to oscillate around the goal state within the first 1,000 episodes. Using the slow parameters, learning is significantly slower, though stable. Figure 3 show the short and long-term performance of both parameters on the control tests.

Baseline Biceps Test

Because the learning parameter sets were optimized using the BBT, the fast parameters 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. Once again, the fast parameters are unstable in the long-term, while the slow parameters remain stable, as shown in Figure 4.

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. The fast parameters remove the overshoot within 200 episodes.

Performance is consistent with the previous tests, with the fast parameters initially learning rapidly, then diverging, while the slow parameters learn more slowly, but remain stable as shown in Figure 5.

Noise Robustness Test

The system performs well on the NRT, without significant changes to learning speed with noise in the inputs representative of those expected in real world experiments.

Long-Term Stability

In order to be practical for subjects with SCI, the agent must be able to adapt quickly (e.g. using the fast parameters), but remain stable (e.g. using the slow parameters). However, all fast parameter sets found were unstable. Several techniques and modifications to the actor-critic were therefore tested in an attempt to improve the stability of the fast parameters.

In the following subsections, references to fast and slow parameters refer to the set of parameters found with similar behavior, most of which have nonzero η_C.

TD-Error Cap

In the previous tests, the magnitude of the TD-error was capped to .5 for training purposes. By lowering this cap, the system is forced to make smaller updates. This improves stability, but slows down learning. Tests showed that the tradeoff between stability and learning speed was not significantly changed.

Muscle Force Weight

When the system is diverging, it first begins to oscillate at high frequency around the goal state. A possible cause is an improper weighting of the squared muscle stimulation in Equation 9. Changes to this constant were found to influence the magnitude and frequency of the jitter, though its onset was relatively constant, and divergence properties unchanged.

Monitor Critic

Under the assumption that divergence occurs because of error in the value function, a possible solution is to only update the actor when the TD-error over the previous k updates has been less than a manually tuned constant, Δ. Tests showed this system to be relatively stable with TD-errors of magnitude less than .02, suggesting $\Delta \approx .02$. The tradeoff between stability and learning speed was again not significantly changed. For small Δ and large k, the system was stable, though learning was slow, while larger Δ and smaller k learned faster but was unstable.

Weight Decay Term

Previous tests had k_A and k_C both set to zero, resulting in no weight decay term. Trials using the parameters in Table 2 with the TD-error capped at .03 on the control test resulted in a policy more robust to varying dynamics.

After slow but stable training with these parameters on the control test, they achieve a policy with an evaluation of −.192 on the control test, −.22 on the BBT, and −.197 on the FTT.

This result is expected, as weight decay terms are known in machine learning to improve generalization. These parameters use larger muscle forces, similar to a PD controller with larger gains. Though these are mostly desirable traits, the long-term stability of the system remains unchanged.

Local Function Approximator Updates

Tests were run using radial basis functions (RBFs) with Gaussian kernels, and using locally weighted linear regression (LWR) as the critic. Though RBFs are common in literature, the incremental variant of LWR used is novel in its application. LWR was implemented following (Schaal, Atkeson, and Vijayakumar 2002). A fixed number of points was selected, initially in an evenly distributed grid over the domain with reasonable initial values. Rather than adding or removing points, the system was updated according to Equations 11 and 12, with both the position and output of each point treated as a weight. Due to matrix arithmetic properties and certain values having been computed during the approximation step, Equations 11 and 12 can be implemented efficiently.

LWR achieved smaller TD-errors than the ANNs and RBFs, though the stability properties of the system remained unchanged.

Toggling Parameter Sets

The most successful approach was to implement a hybrid controller, switching to the fast parameter set when rapid learning is required, followed by a longer period using the slow parameters to bring the critic back on-policy.

This can be tested by first testing performance on the BBT, and then testing performance in an environment that switches between various tests, requiring constant adaptation. For this, a new test was devised, the *Fatigued Biceps Test* (FBT). This is required because a policy that performs well on the BBT may also perform well on the FTT, because both call for less biceps stimulation and more triceps stimulation. The FBT is identical to the FTT, except that the muscle affected is the biceps brachii.

For these tests, the parameters were switched to the fast parameters whenever the environment switched dynamics. For practical applications, the fast parameters can be used for initial adaptation when the agent is first used on a subject, after which the slow parameters can be used to maintain stability (e.g. Figure 8). At any point, if a subject notices the performance of his or her arm has deteriorated due to muscle fatigue or other changes, the subject could activate a short-term switch to the fast parameters to improve performance (e.g. Figure 7).

Figures 7 and 8 show that this toggling system can learn quickly and remain stable in the long-term. Figure 7 also shows how the system can rapidly converge to a policy with a reasonable evaluation on both the BBT and FBT, while remaining stable.

Conclusion and Future Work

We have examined reinforcement learning's application to FES control of the upper extremity. In particular, we have shown that rapid learning is achievable with the continuous actor-critic architecture, though the system changes too rapidly for the critic to remain on-policy, resulting in long-term divergence. Slowing the learning to a speed with which the critic can keep up, the system becomes impractically slow. These two results can be combined by toggling between the fast and slow parameters to achieve both rapid learning and long-term stability. We have also shown the continuous actor-critic to be robust to noise similar to that expected in the real world application of FES control. This controller achieves

the goal of adapting to realistic changes in arm dynamics within 200 episodes, while remaining stable and robust.

As this is one of the first attempts known by the authors to apply RL techniques to FES, the research area is still open for significant development. These encouraging results have inspired further work in the application of RL to FES control. At the Lerner Research Institute (LRI) of the Cleveland Clinic Foundation, researchers are preparing for human trials of this controller for planar arm movement. These experiments are expected to commence in Summer, 2009.

Researchers at the LRI have also created a detailed three-dimensional musculoskeletal model of a human arm (Chadwick et al. 2009). Pending successful results from the real world application of RL for planar control, this same controller could be applied to the three-dimensional model, and eventually three-dimensional human trials. The primary difficulty in the switch will be the increase in the dimension of the action space, as the three-dimensional model includes over 100 muscles, though this can be overcome by clustering similar muscles into groups that are all given equal stimulation.

This paper has shown that RL is a viable approach for FES control of a human arm, and will hopefully open up a vein of further research in the area, with the long-term goal of restoring natural motor function to people with SCI.

Acknowledgments

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

References

Abbas JJ, Triolo RJ. Experimental Evaluation of an Adaptive Feedforward Controller for use in Functional Neuromuscular Stimulation Systems. IEEE Transactions on Rehabilitation Engineering 1997;5(1):12–22. [PubMed: 9086381]

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? Australasian Physical Engineering Sciences in Medicine 2007;30(4):317–323. [PubMed: 18274072]

Chadwick EK, Blana D, van den Bogert AJ, Kirsch RF. A Real-Time 3D Musculoskeletal Model for Dynamic Simulation of Arm Movements. IEEE Transactions on Biomedical Engineering. 2009 To appear.

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 Transactions on Rehabilitation Engineering 1997;5(1):2–11. [PubMed: 9086380]

Crago PE, Lan N, Veltink PH, Abbas JJ, Kantor C. New Control Strategies for Neuroprosthetic Systems. Journal of Rehabilitation Research and Development 1996;33(2):158–172. [PubMed: 8724171]

Crago PE, Nakai RJ, Chizeck HJ. Feedback Regulation of Hand Grasp Opening and Contact Force During Stimulation of Paralyzed Muscle. IEEE Transactions on Biomedical Engineering 1991;38(1):17–28. [PubMed: 2026428]

Davoodi R, Andrews JB. Computer Simulation of FES Standing Up in Paraplegia: A Self-Adaptive Fuzzy Controller With Reinforcement Learning. IEEE Transactions on Rehabilitation Engineering 1998;6(2):151–161. [PubMed: 9631322]

Doya K. Reinforcement Learning in Continuous Time and Space. Neural Computation 2000;12(1): 219–245. [PubMed: 10636940]

Ferrarin M, Pavan EE, Spadone R, Cardini R, Frigo C. Standing-Up Exerciser Based on Functional Electrical Stimulation and Body Weight Relief. Medical and Biological Engineering and Computing 2002;40(3):282–289. [PubMed: 12195974]

Izawa J, Toshiyuki K, Koji I. Biological Arm Motion Through Reinforcement Learning. Biological Cybernetics 2004;91(1):10–22. [PubMed: 15309543]

Jaeger RJ. Lower Extremity Applications of Functional Neuromuscular Stimulation. Assistive Technology 1992;4(1):19–30. [PubMed: 10148013]

Jagodnik, KM.; van den Bogert, AJ. A Proportional Derivative FES Controller for Planar Arm Movement. 12th Annual Conference International FES Society; Philadelphia. 2007.

Kobetic R, Marsolais EB. Synthesis of Paraplegic Gait With Multi-Channel Functional Neuromuscular Stimulation. IEEE Transactions on Rehabilitation Engineering 1994;2:66–79.

Lynch LC, Popovic RM. Functional Electrical Stimulation: Closed-Loop Control of Induced Muscle Contractions. IEEE Control Systems Magazine 2008;28(2):40–50.

McLean SG, Su A, van den Bogert AJ. Development and Validation of a 3-D Model to Predict Knee Joint Loading During Dynamic Movement. Journal of Biomechanical Engineering 2003;125(6): 864–874. [PubMed: 14986412]

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. Archives of Physical Medicine and Rehabilitation 2001;82:1380–8. [PubMed: 11588741]

Peckham PH, Knutson JS. Functional Electrical Stimulation for Neuromuscular Applications. Annual Review of Biomedical Engineering 2005;7:327–360.

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]

Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach. 2. Englewood Cliffs, NJ: Prentice Hall; 1995.

Schaal S, Atkeson CG, Vijayakumar S. Scalable Techniques From Nonparametric Statistics for Real Time Robot Learning. Applied Intelligence 2002;17:49–60.

Schultz AB, Faulkner JA, Kadhiresan VA. A Simple Hill Element-Nonlinear Spring Model of Muscle Contraction Biomechanics. Journal of Applied Physiology 1991;70(2):803–812. [PubMed: 2022572]

Sheffler LR, Chae J. Neuromuscular Electrical Stimulation in Neurorehabilitation. Muscle Nerve 2007;35(5):562–590. [PubMed: 17299744]

Stroeve S. Learning Combined Feedback and Feedforward Control of a Musculoskeletal System. Biological Cybernetics 1996;75(1):73–83. [PubMed: 8765656]

Sujith OK. Functional Electrical Stimulation in Neurological Disorders. European Journal of Neurology 2008;15(5):437–444. [PubMed: 18394046]

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

Thomas, PS.; Branicky, M.; van den Bogert, AJ.; Jagodnik, KM. Creating a Reinforcement Learning Controller for Functional Electrical Stimulation of a Human Arm. Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems; New Haven, CT. 1–6 June 2008; 2008.

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. Archives of Physical Medicine and Rehabilitation 2002;83(8):1093–1099. [PubMed: 12161830]

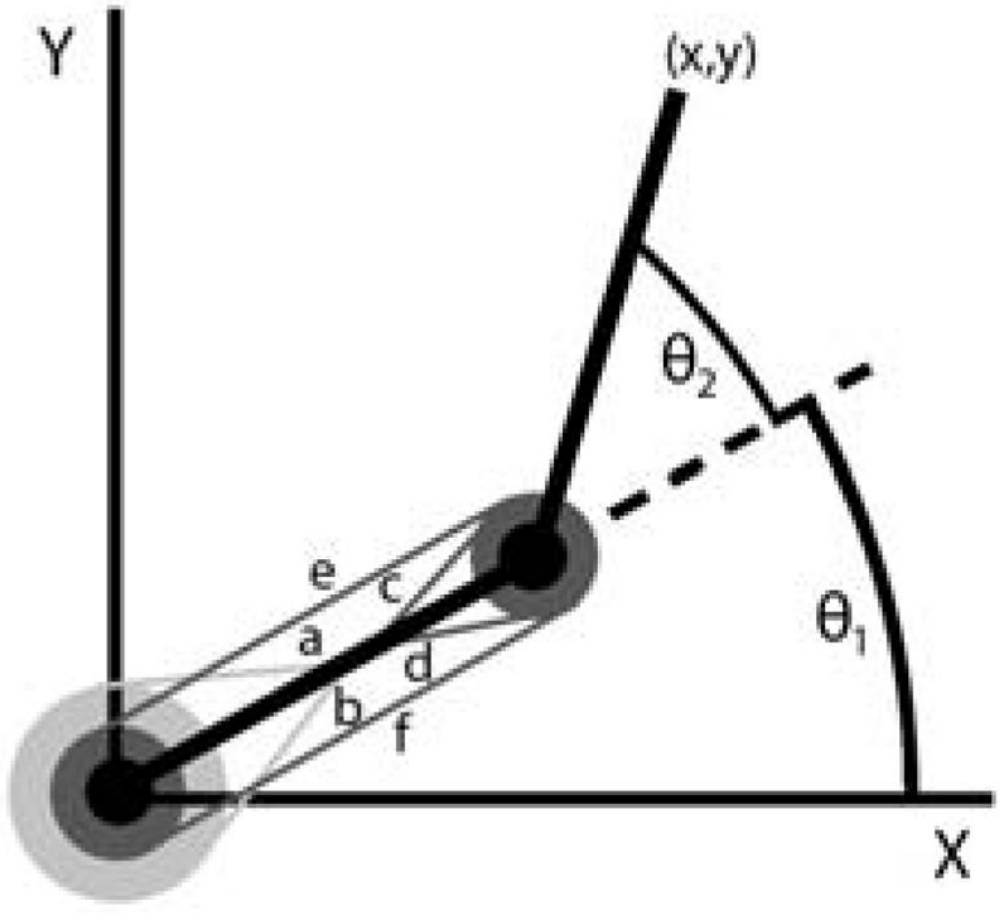

Figure 1.
Two-joint, six-muscle biomechanical arm model used. Antagonistic muscle pairs are as follows, listed as (flexor, extensor): monoarticular shoulder muscles (a: anterior deltoid, b: posterior deltoid); monoarticular elbow muscles (c: brachialis, d: triceps brachii (short head)); biarticular muscles (e: biceps brachii, f: triceps brachii (long head)).

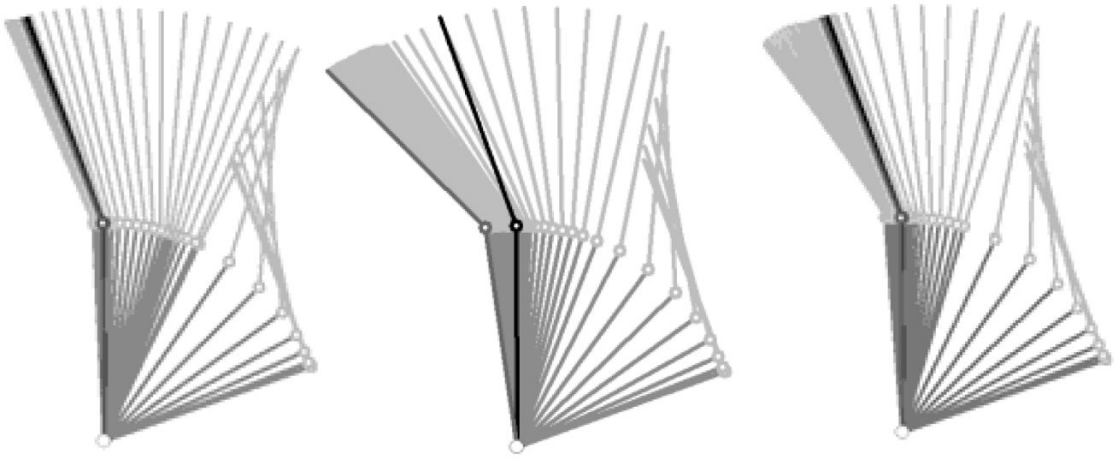

Figure 2.
Initial actor ANN's performance on a particular motion for the three tests. The black state is the goal state (90°, 20°), the medium grey state is the final state after two seconds of simulation, and the light grey states are snapshots of the arm location taken every 20ms. The initial condition 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.

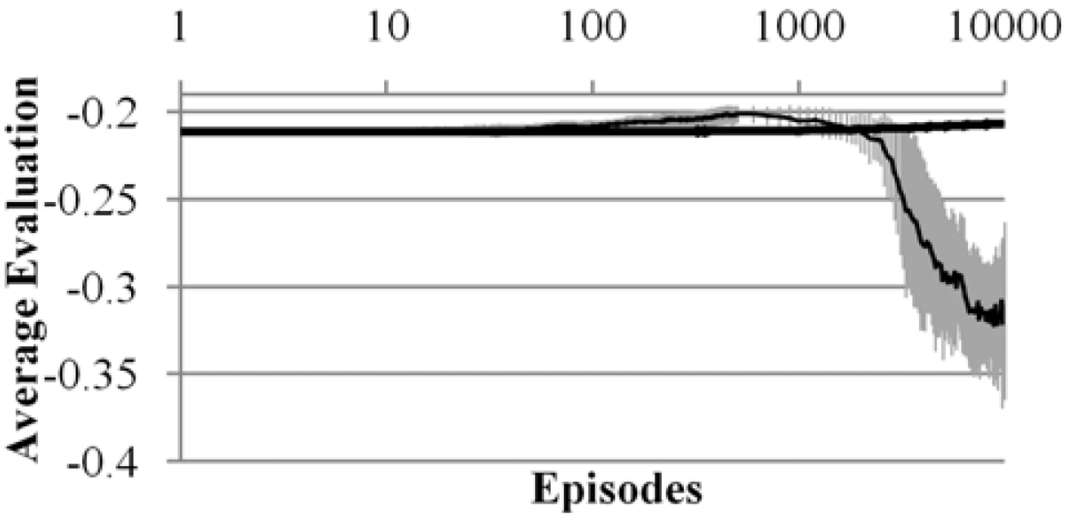

Figure 3.
The actor-critic's average evaluation on the control test with standard devation error bars (*N*=10). Evaluations represent those just prior to the x^{th} episode. For this and the next two figures, the thick line represents the slow parameters (finishes higher in all plots), while the thin line represents the fast parameters (finishes lower in all plots).

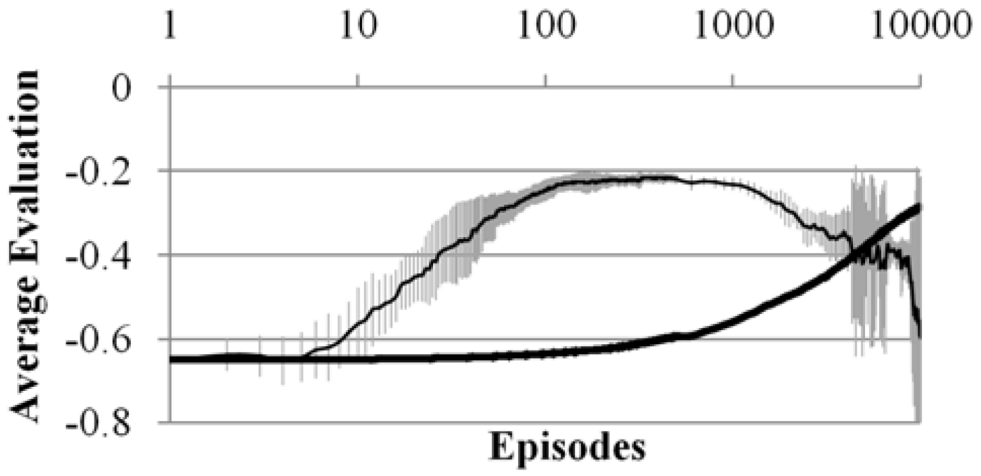

Figure 4.
The actor-critic's evaluation on the BBT.

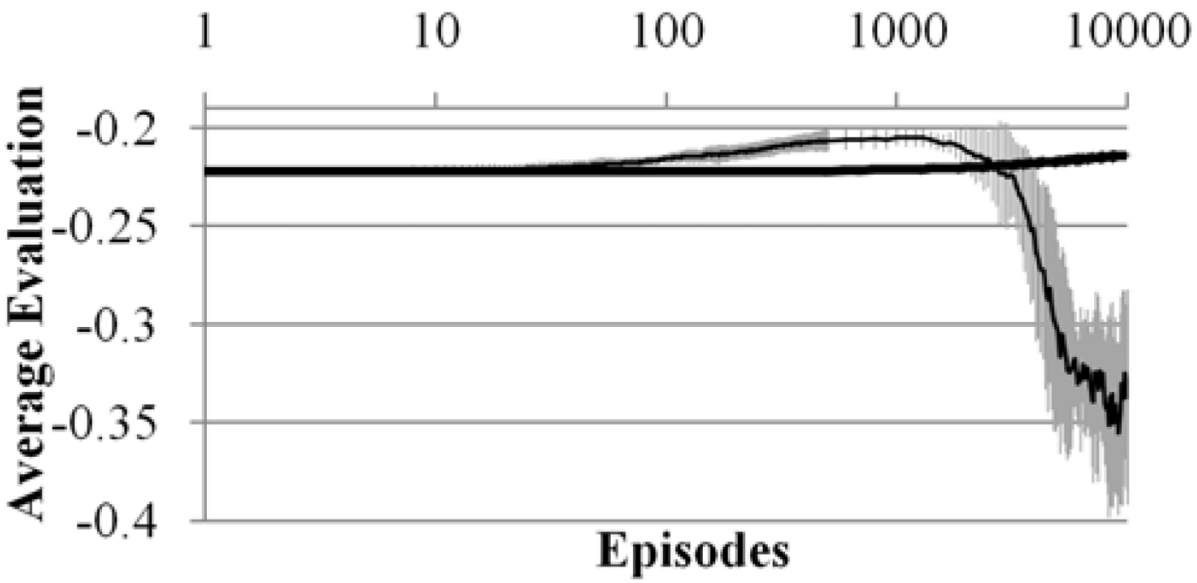

Figure 5.
The actor-critic's average evaluation on the FTT.

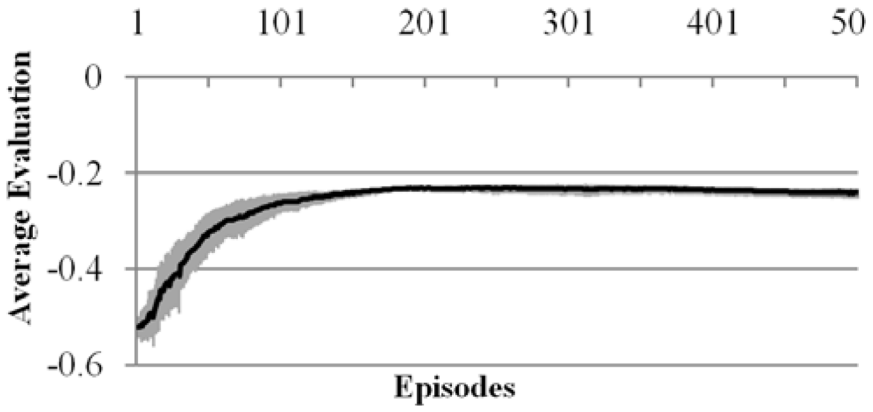

Figure 6.
The actor-critic's average evaluation on the NRT with σ_θ= .1, $\sigma_{\dot{\theta}}$ = .3, and N=10.

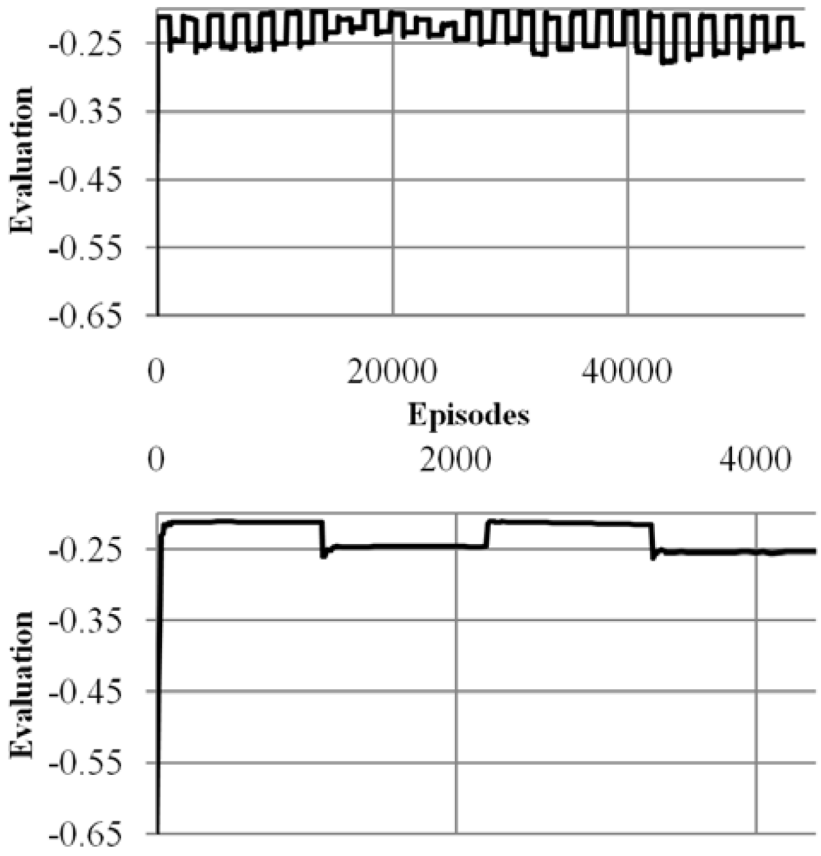

Figure 7.
The actor-critic's evaluation, where the environment starts as the BBT, then switches to the FBT after 1,100 episodes, then back to the BBT after 2,200 episodes, etc. The parameters also switch to the fast parameters for the first 100 episodes on each test to the slow parameters for the remaining 1000 episodes on each test. The top plot shows the long-term performance while the bottom shows the short-term performance.

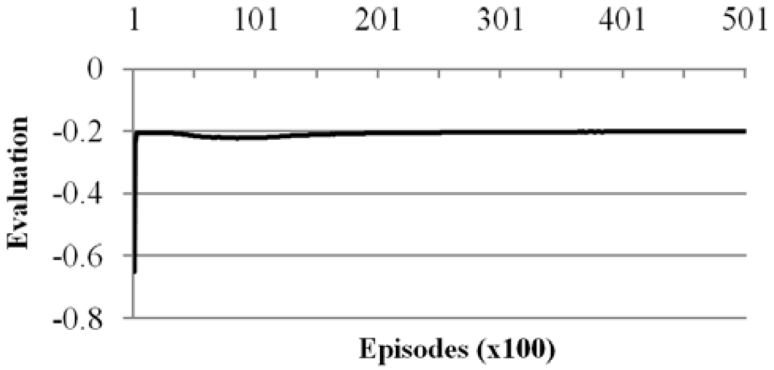

Figure 8.
The actor-critic's evaluation over 50,000 episodes on the BBT using the fast parameters for the first 200 episodes, and the slow parameters thereafter.

Table 1

Parameter sets representative of those found by RRHC.

Parameter Names	η_A	η_C	k_A	k_C
Slow	10	.344	0	0
Fast	70	0	0	0

Table 2

Two of the best parameter sets found from optimization after manual tuning.

η_A	η_C	k_A	k_C
70	.344	2E-7	2E-6