To appear: Proc. of the 15th IEEE-RAS-EMBS International Conference on Rehabilitation Robotics (ICORR 2017), July 17-20, 2017, QEII Centre, London, UK, pp. 1-8. ©2017 IEEE

Learning from Demonstration: Teaching a Myoelectric Prosthesis with an Intact Limb via Reinforcement Learning

Gautham Vasan and Patrick M. Pilarski

Abstract— **Prosthetic arms should restore and extend the capabilities of someone with an amputation. They should move naturally and be able to perform elegant, coordinated movements that approximate those of a biological arm. Despite these objectives, the control of modern-day prostheses is often non-intuitive and taxing. Existing devices and control approaches do not yet give users the ability to effect highly synergistic movements during their daily-life control of a prosthetic device. As a step towards improving the control of prosthetic arms and hands, we introduce an intuitive approach to training a prosthetic control system that helps a user achieve hard-to-engineer control behaviours. Specifically, we present an actor-critic reinforcement learning method that for the first time promises to allow someone with an amputation to use their non-amputated arm to teach their prosthetic arm how to move through a wide range of coordinated motions and grasp patterns. We evaluate our method during the myoelectric control of a multi-joint robot arm by non-amputee users, and demonstrate that by using our approach a user can train their arm to perform simultaneous gestures and movements in all three degrees of freedom in the robot's hand and wrist based only on information sampled from the robot and the user's above-elbow myoelectric signals. Our results indicate that this learning-from-demonstration paradigm may be well suited to use by both patients and clinicians with minimal technical knowledge, as it allows a user to personalize the control of his or her prosthesis without having to know the underlying mechanics of the prosthetic limb. These preliminary results also suggest that our approach may extend in a straightforward way to next-generation prostheses with precise finger and wrist control, such that these devices may someday allow users to perform fluid and intuitive movements like playing the piano, catching a ball, and comfortably shaking hands.**

I. INTRODUCTION

Humans often exploit the dynamics of their complex musculoskeletal system in ingenious ways to generate efficient and coordinated movement. When the central nervous system (CNS) produces voluntary movement, various muscles, each comprising thousands of motor units, are simultaneously activated and coordinated. Computationally, this is a daunting task since the CNS needs to handle the large number of degrees of freedom (DoF) that must be continually adjusted and controlled (i.e., the *degrees-of-freedom problem* [1]). However, according to Bernstein [2], humans do not control elementary degrees of freedom, but instead use *muscle synergies*—the coordinated activation of a group of muscles—to handle their degrees-of-freedom problem. Recent findings of d'Avella et al. [3] suggest that the CNS encodes a set of muscle synergies, and that it combines them in a task-dependent fashion in order to generate the muscle contractions that lead to desired movements. While modern hand prostheses now include a limited number of predefined synergistic grasping patterns, synergistic actuation of the kind described by d'Avella et al. is largely missing from most if not all commercial prosthetic devices.

Since the 1960s, the most common way of controlling powered prostheses has been through surface electromyography (sEMG), termed *myoelectric control*, which involves measuring the electrical manifestation of muscle contraction. Despite significant technological advancements, a large proportion of amputees stop using myoelectric prostheses due to non-intuitive control, lack of sufficient feedback, and insufficient functionality [4]. Even though sophisticated upper extremity prostheses like the Modular Prosthetic Limb (MPL) are capable of effectuating almost all of the movements as a human arm and hand and with more than 100 sensors in the hand and upper arm (26 DoF and 17 degrees of control) [5], they can be useful only if robust systems of control are available.

The most widely used approach to myoelectric control is still direct proportional control [6]. In direct control, the magnitude of muscle contraction is used to move a degree of control (DoC, involving one or more prosthetic joints) of the prosthesis using a proportional mapping [7]. This allows the selection of control muscles based on physiological functions but has the disadvantage of typically requiring two control muscles for each prosthetic DoC [7]. In order to control additional motions of the prostheses, various switching methods sequentially transition between different DoCs (c.f., [8], [9]). These simplistic methods provide reliable control, but lack the functionality to smoothly operate multiple DoCs.

More recently, pattern recognition methods have started to see commercialization and clinical use [10], [11]. Pattern Recognition methods use classification [10] and regression [12] techniques to translate EMG signals into usable control commands for a multi-function prosthetic limb. Myoelectric classification for prosthetic control is not only possible but also highly accurate, even with a large number of functions (> 10) [13] [14]. However, while natural movements are continuous and require simultaneous, coordinated articulations of the multiple DoFs, pattern classification provides only a discrete approximation of the continuous parameter space. Current methods can typically generate reliable activation

This research was undertaken, in part, thanks to funding from the Canada Research Chairs program, the Canada Foundation for Innovation, and was supported by the Alberta Machine Intelligence Institute (Amii), Alberta Innovates – Technology Futures (AITF), the Government of Alberta, and the Natural Sciences and Engineering Research Council (NSERC).

G. Vasan and P.M. Pilarski are with the Department of Computing Science and the Department of Medicine, University of Alberta, Edmonton, AB T6G 2E1, Canada. Please direct correspondence to: vasan@ualberta.ca; pilarski@ualberta.ca

in only one class. Additionally, proportional control is not directly obtained from the classification, but instead requires additional processing [15].

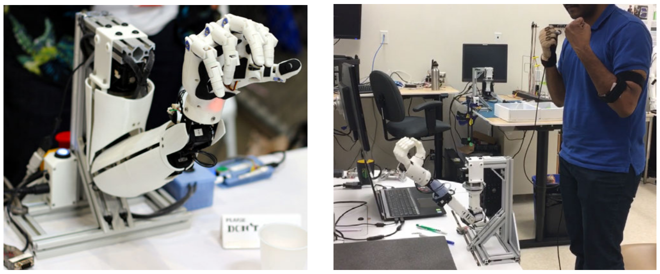
Few alternative methods employ ideas drawn from motor skill learning and brain plasticity to extend direct control principles to multiple DoFs. Pistohl et al. and Ison et al. showed that users adapt to controls within a single session regardless of their initial intuitiveness or relationship with kinematics and develop muscle synergies associated with enhanced control of the myoelectric interface [16], [17]. Even though this approach promises improved control for prosthetic users, it relies completely on the human user to adapt to his/her prosthetic device rather than vice versa. Great utility may arise from a bidirectional partnership between the prosthetic device and its human user—while the user improves his or her ability to communicate their intentions to the prosthesis, the prosthesis would learn to anticipate and adapt to that specific needs of the user and improve its own ability to satisfy them [18].

The aforementioned approaches are all trying to address a fundamental issue—how to overcome the significant mismatch between the number of functions available in a modern powered prosthesis and the number of functions an amputee can attend to at any moment. With this goal in mind, in the present work we develop a method that could allow someone with an amputation to use their non-amputated arm to teach their prosthetic arm how to move in a natural and coordinated way. Such a paradigm could well exploit the muscle synergies already learned by the user. Consider cases where an amputee has a desired movement goal, e.g., “add sugar to my coffee,” “button up my shirt,” or “shake hands with an acquaintance.” In these more complicated examples, it may be difficult for a user to frame their objectives in terms of device control parameters or existing device gestures, but they may be able to execute these motions skillfully with their remaining biological limb.

One approach that has been shown to reduce barriers for humans specifying a complex control policy (i.e., a desired behavior) is *learning from demonstration* (LfD) [19]. In LfD, a policy that map states to actions is learned from the examples or demonstrations provided by the teacher. The examples are defined as a sequence of state-action pairs or trajectories that are recorded during the teacher’s demonstration of the recorded behavior [19]. By formulating prosthetic limb training as a LfD task, we present a new scenario wherein an amputee could teach their prosthesis how to move by showing desired movements via the movement of his or her non-amputated limb.

II. METHODS

A myoelectric prosthesis can be thought of as a wearable robot that responds to sEMG control signals. A myoelectric user is faced with the task of interacting with a robot to accomplish everyday tasks. It is reasonable to expect that most people with amputations may not be robotics experts, but they could have ideas of what their prosthesis should do, and therefore what types of synergies their prosthetic control algorithms should give rise to. A natural and practical extension of having this knowledge is to use it to develop their own desired control algorithms. However, unlike the usual practice of directly engineering a control approach, we suggest that desired behaviours could be learned by the prosthesis from demonstrations provided by the user.



(a) The Bento Arm

(b) Experimental Setup

Fig. 1: Experimental setup which includes the Bento Arm, Delsys Trigno Wireless Lab and CyberTouch II. The Bento Arm as used in our trials had 5 active DoFs including shoulder rotation, elbow flexion/extension, wrist pronation/supination, wrist flexion/extension and hand open/close.

In the present work, we specifically address the common case of a user with a unilateral, transhumeral amputation—someone missing their hand, forearm, and elbow. In this setting, the user has one biological limb, and one robotic limb that they wish to train to appropriately respond to the commands being generated by the muscle tissue in the user’s residual limb. We refer to the arm generating the control signals as the user’s *control arm*. For someone with an amputation, these control signals would come from the residual limb that is attached to their robotic prosthesis, where EMG signals from residual biceps and triceps may be already used in the direct control of their robotic elbow. We term the arm providing the training movements the *training arm*, or the contralateral, intact biological limb.

By asking the user to perform the *same motion* with both arms (or visualize performing, in the case of an amputated control limb, and as in pattern recognition training [10]), we suggest that the motion of the training limb can provide training information for creating a prosthetic policy that maps the state of the control limb (e.g., gross robot limb position and control-limb EMG signals) to motor commands for the remaining joints of the prosthetic hands and wrist not controllable by the user. The robotic prosthesis can then use its learned, state-conditional policy to “fill in the gaps” for the user during ongoing, post-training use.

For this study, we first explore prosthetic LfD with able-bodied participants. In the case of these able-bodied subjects, the control arm is defined as the arm providing the control signals to a robot limb, where control channels are sampled in the same locations as they would be for someone with an amputation. The training arm is their contralateral limb, and represents what would be the user’s non-amputated arm.

Robotic Arm: Our experiments were done via an open-source robot platform known as the Bento Arm, as shown

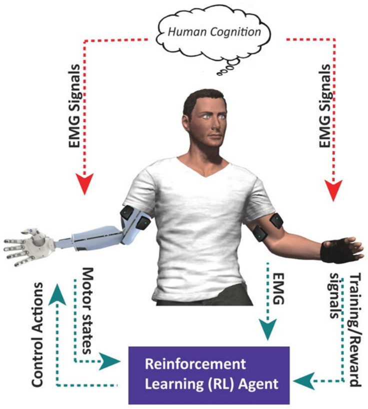


Fig. 2: Schematic showing the flow of information through the experimental setup during the training period.

in Fig. 1. The Bento Arm is a myoelectric training tool to assess and train upper-limb amputees in how to use their muscle signals prior to being fit with a myoelectric prostheses [20]. Although designed to be donned via a socket, for repeatability in this experiment the Bento Arm was rigidly fixed to a desk directly in front of the able-bodied subject (Fig. 1b), such that its arm position was aligned with the control-delivering arm of the subject. The myoelectric control system received the angular position and velocity of the following joints from the Bento Arm: elbow $\langle \theta_e, \dot{\theta}_e \rangle$, wrist flexion/extension $\langle \theta_{wf}, \dot{\theta}_{wf} \rangle$, wrist rotation $\langle \theta_{wp}, \dot{\theta}_{wp} \rangle$, and aperture angle of the gripper hand $\langle \theta_h, \dot{\theta}_h \rangle$.

EMG Data Acquisition: We used a 16-Channel Delsys Trigno Wireless Lab (Delsys, Inc.) to record EMG signals from our subjects. As shown in Fig. 1b, able-bodied subjects were fitted with four Delsys Trigno units that provided sEMG signals and inertial measurements from accelerometers, magnetometers and gyroscopes. The Trigno units were placed on the control arm of each subject as follows: two units on the biceps and two units on the triceps. We placed one additional inertial measurement unit (IMU) on the training arm's wrist to measure the desired wrist rotation angle (θ^*_{wp}).

Motion Capture Glove: The desired joint angle configurations for wrist flexion/extension (θ^*_{wf}) and hand open/close (θ^*_h) were defined by the subject using a CyberTouch II system (CyberGlove Systems LLC) worn on the hand of their training arm. The CyberTouch (shown in Fig. 1b) uses resistive bend-sensing technology to accurately transform hand and finger motions into real-time digital joint-angle data (18 high-accuracy joint-angle measurements). When this data was coupled with that from the single IMU on the training wrist, the subject was able to use the movement of their training arm to precisely specify their desired pose for the robot arm's hand and wrist.

Phase I: Recording training data

In this phase, subjects were instructed to execute a repetitive sequence of simple reaching and grasping movements that were mirrored by both their control and training arms (for someone with an amputation, this would correspond to trying to perform identical movements using their non-amputated arm and the prosthetic arm). The training arm demonstrated the desired movement and grasp pattern to the prosthetic arm. During training, the elbow of the Bento Arm was actuated via proportional myoelectric control from the subject's control arm, while to wrist and hand of the Bento Arm were actuated via direct teleoperation—i.e., the Bento Arm copied the training arm's movements as reflected to the contralateral side. As shown in Fig. 2 and described above, we recorded desired angles from the subject's training arm (wrist and finger joints) using the motion capture glove and inertial measurement system.

For our experiment, we chose a simple, repeatable movement as the desired behavior—a bicep-curl motion involving the smooth alternation of 1) supinated hand-closed wrist flexion during elbow flexion, and 2) hand-open pronation with wrist extension during elbow extension. The position of the wrist and hand was correlated to the angular position of the elbow joint, such that any given elbow position could be uniquely mapped by a policy into a higher dimensional combination of joint motions. Using the specific approach described previously by Pilarski et al. [21]–[23], mean-absolute-value signals recorded from antagonistic muscle groups (biceps/triceps) were mapped to joint velocity commands in order to control the elbow joint of the Bento Arm—i.e., the user controlled the elbow joint of the Bento Arm using EMG signals from their control arm using direct proportional control. The subject used their training arm to demonstrate the desired behavior for three joints: wrist flexion/extension, wrist rotation and opening/closing the hand using the motion capture glove (the CyberTouch system) and IMU signals. The CyberTouch was worn on the intact limb and used only during this phase. We recorded all the real-valued data signals we received from the Bento Arm, Delsys Trigno system and the CyberTouch II while the user repeatedly demonstrated the desired behavior to the prosthetic arm. In this phase, no machine learning takes place. We effectively asked the subjects to teleoperate the Bento Arm while demonstrating the desired behavior since seeing the motor outcomes on the Bento Arm was found to help subjects visualize what the prosthetic arm would actually do. We recorded data for $\geq 5mins$ for each user. All subjects $(n = 3)$ gave informed consent in accordance with the studys authorization by the University of Alberta Health Research Ethics Board.

Mapping contralateral training hand demonstrations to robot hand joint angles: We fixed our frame of reference (3-dimensional euclidean space) relative to the wrist such that every hand movement could be represented as series of rotations along the x, y and z axis (i.e, roll, pitch and yaw respectively). The CyberTouch provided wrist pitch (i.e., flexion/extension) and yaw (i.e., radial/ulnar deviation) angles of the intact limb. We placed an additional Trigno unit on the wrist to capture pronation/suppination (i.e., roll) of the wrist. While the right wrist rotates in a clock-wise direction, the left wrist rotates in an anti-clockwise direction. However, the pitch and yaw rotation axes are similar for both arms

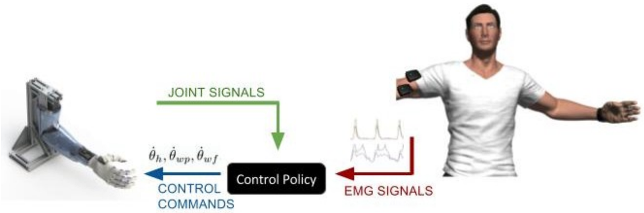


Fig. 3: Schematic showing the flow of information during deployment. The controller generates new robot joint velocity commands using its learned policy, EMG data from the user, and sensor readings from the robot limb.

(relative to our frame of reference). Hence, we used wrist pitch angle of the training arm as target for the Bento Arm. In the case of wrist rotation, the difference between 2π and the joint angle was used as the desired actuator angle. Yaw was not present on the robot and thus not used in this study.

Phase II: Learning a robot control policy

The user was not involved in this phase. A reinforcement learning agent, described below, was tasked with learning and maintaining a control policy for the three target actuators on the robot arm: wrist flexion/extension, wrist rotation and opening/closing the hand. The learned control policy should pick actions such that the actuators instantaneous position matches the joint configuration demonstrated by the subject using the motion capture glove. The agent used the recorded data to learn on its own through trial and error. In essence, the robotic arm trained itself to track the desired trajectory via joint velocity modulation (further details about robot learning are given in Sec. III). In practice, learning could be conducted once a limb is doffed, e.g., overnight or periods of non-use.

Phase III: Testing the learned control policy

During testing and deployment, a subject used his or her EMG signals from the control limb to move the prosthetic arm, again via conventional direct myoelectric control (as shown in Fig. 3). The subjects were asked to freely actuate the elbow joint of the Bento arm using conventional EMG-based linear-proportional control, and the system would use the learned control policies to move the remaining target joints (i.e., in response to user control choices the system would now effect the synergies learned during the training phase). The non-amputated arm was free to perform any movement. The controller selected appropriate velocity commands at each timestep depending on the EMG signals from the user and sensor readings from the Bento Arm (as shown in Fig. 3). For safety during initial user testing, all joint velocities were bounded by $-2 \leq \dot{\theta}_j \leq 2\ radians/sec.$

III. Learning a control policy in real-time using Actor-Critic Reinforcement Learning

In order to learn from the demonstrations of the training (non-amputated) arm, we applied actor-critic reinforcement learning (ACRL) as our primary mechanism for LfD policy development. As shown in previous work by Pilarski et al. [21], [23] and others, ACRL is a flexible, online learning framework that can be easily adapted to different application domains and the needs of individual amputees. In particular, reinforcement learning (RL) enables a robot to autonomously discover optimal behavior through trial-and-error interactions with its environment. Instead of explicitly detailing the solution to a problem, in RL the user provides feedback about the performance of the robot in terms of a scalar reward signal. The goal of any RL agent is to maximize the expected cumulative reward (also known as the *return*) [24].

ACRL methods in particular are well-suited for the LfD task in the present work since they are model-free, parameter-based incremental learning algorithms which allow fast computation (millisecond updates even over large state spaces) [23]. In the field of robotics, one of the earliest successes of ACRL was shown by Benbrahim et al. for biped locomotion [25]. Peters and Schaal have since applied Natural Actor Critic methods to teach a 7-DoF anthropomorphic robot arm to hit a ball as far as possible [26].

The RL agent chooses control actions denoted a based on the learned policy. The actions, in this case, were real-valued signals which indicate the desired joint velocity. At each time step, the continuous actions a_{wf}, a_{wp}, a_h were taken according to each joint's respective parameterized policy. The actions were drawn from a normal distribution with a probability density function defined as $\mathcal{N}(s,a) = \frac{1}{\sqrt{2\pi\sigma^2(s)}} \exp\Big(-\frac{(a-\mu(s))^2}{2\sigma^2(s)}\Big)$. The parameters of the normal distribution were functions of the system's learned weight vectors w_μ and w_σ as given by $\dot{\theta} \approx a \leftarrow \mathcal{N}\{\mu,\ \sigma\}$. The actions selected by the RL agent were allowed to persist for $\sim 75ms$ to give the robot enough time to execute the control commands and better explore the world. Learning updates occurred every $\sim 40ms$ of the training period.

In our policy parameterization, the scalars $\mu = {u_\mu}^T x(s)$ and $\sigma = \exp({u_\sigma}^T x(s) + \log(\sigma_c))$ were defined as a linear combination of the parameters of the policy and the feature vector of the state $x(s)$. Actor weights w_μ and w_σ were updated based on the compatible features for normal distribution [27]. We used accumulating eligibility traces for both the critic (e_v) and the actor $(e_\mu$ and $e_\sigma)$ [27].

The ACRL agent, implemented as described in Pilarski et al. [21], was given control of three continuous angular velocities $\dot{\theta}_{wf}, \dot{\theta}_{wp}$ and $\dot{\theta}_h$, where they denote the angular velocities of wrist flexion/extension, wrist pronation/suppination and the gripper hand. Raw EMG signals s were rectified and averaged as $\bar{s} = (1-\tau)\bar{s} + \tau s$, with a time constant $\tau = 0.037$. Differential EMG was later computed for antagonistic muscle pairs (biceps/triceps), $\bar{s}_1 = \bar{s}_{BI} - \bar{s}_{TRI}$ to control the robot's elbow joint. The following signals were used to construct the state approximation vector for each joint j controlled by the learning system $x(s)$: $\langle \bar{s_1}, \theta_e, \dot{\theta}_e, \theta_j \rangle$; where $\theta_e, \dot{\theta}_e, \theta_j$ denote elbow joint angle, elbow joint velocity and current angle of the joint controlled by the robot.

We used tile coding [24] to construct the state approximation vector $x(s)$ used in learning. Our state representation consisted of 25 incrementally offset tilings $(width = 1)$ for better generalization. Each tiling had two resolution levels $N_R = [4, 8]$, along with a single baseline unit. This resulted in a binary feature vector of length 108,801 hashed down to a memory size of 2048, with m = 51 active features per step. The learning parameters were set as follows: $\sigma_c = 1$, $\alpha_v = 0.1/m$, $\alpha_\mu = 0.02/m$, $\alpha_\sigma = 0.25\alpha_\mu$, $\gamma = 0.96$ and $\lambda = 0.7$. Weight vectors e_v, v, e_μ, w_μ, e_σ, w_σ were initialized to zero and σ bounded by $\sigma \geq 0.01$.

The ACRL systems was trained incrementally using repeated cycles of the training data earlier recorded in Phase 1, as described in Sec. 2. Total training time was held constant at 45 min after which the learned control policy was tested on a different data set for accuracy. The control learner received negative rewards r on each step proportional to the difference between the target and current joint angles: $r_j = -|{\theta_j}^* - \theta_j|$, in radians. Each controlled joint had its own ACRL learner with its own reward function r_j.

Performance of the learning system was measured based on its ability to achieve desired joint angles. All learning algorithms were run on a Lenovo Flex-3 Laptop with Intel Core i7-6500U @2.50GHz x 4 and 8GB RAM. We used the Robot Operating System (ROS) on Ubuntu 16.04 to send and receive information and commands from the Bento Arm, CyberTouch II and the Delsys Trigno Wireless Lab.

IV. RESULTS

In our experiments, the actuator targets ${\theta_j}^*$ were demonstrated by the user on a moment to moment basis during training. To serve as a baseline performance measure for post-training ACRL policies, we used a reactive control approach [21] as an offline equivalent to direct teleoperation. Since the set of joints targets are not known until the user demonstrates them, a simple baseline teleoperation policy would be to observe the desired joint angle ${\theta_j}^*$ and take an action a_j that moves current actuator angle θ_j towards the target angle as quickly as possible. This control approach assumes perfect knowledge of desired joint angles, (i.e, the states and targets are fully observable), so is only applicable as a training-data baseline.

Algorithm 1 Actor Critic Reinforcement Learning

```
procedure ACRL    ▷ Learn a control policy for joint $j$
    Initialize: $s$, $x(s)$ and weights $e_v$, $v$, $e_\mu$, $w_\mu$, $e_\sigma$, $w_\sigma$
    for each step do
        $a_j \leftarrow \mathcal{N}(\mu, \sigma^2)$
        Take action $a$ in state $s$ and observe the next state
            $s'$ and reward $r_j$
        $x(s') \leftarrow tilecode(s')$    ▷ Tile Coding [24]
        $\delta \leftarrow r_j + \gamma v^T x(s') - v^T x(s)$
        $e_v \leftarrow \gamma\lambda e_v + x(s)$
        $v \leftarrow v + \alpha_v \delta e_v$
        $e_\mu \leftarrow \gamma\lambda e_\mu + (a - \mu)x(s)$
        $w_\mu \leftarrow w_\mu + \alpha_\mu \delta e_\mu$
        $e_\sigma \leftarrow \gamma\lambda e_\sigma + ((a - \mu)^2 - \sigma^2)x(s)$
        $w_\sigma \leftarrow w_\sigma + \alpha_\sigma \delta e_\sigma$
        $x(s) \leftarrow x(s')$
```

Figure 4 shows the quartile analysis of mean absolute angular error accumulated over two minutes of learning/testing for five independent runs for each subject. The ACRL learner showed continuous improvement in performance over learning. Importantly, performance at the end of the learning phase was consistent with performance during actual user control in the testing phase for all three subjects. The negative of the mean absolute angular error is the average reward received during the evaluation period.

Figure 5 shows examples of the joint control trajectories achieved by the ACRL learner during the early and late stages of learning, and during testing. As shown in Fig. 5, the joint angles remained within the target regions for the majority of the evaluation period during both testing and training scenarios. The trajectories achieved by the ACRL learner are compared with direct teleoperation (reactive control as described above). After 20 min of learning the ACRL learner started to achieve the desired joint trajectories, though the controller did visibly overshoot/oscillate around the desired trajectory. The controller started to tightly track the desired trajectory following 30 min of learning. However, as seen in Fig. 5, we observe occasional spikes in the joint angles whenever there is a sudden transition in the desired position. These spikes likely correspond to the fact that less time is spent in training for transitional motion.

V. DISCUSSION

A. *Comparison of performance between subjects*

Among the 3 subjects, Subject 1 had the most familiarity with the experimental setup. The rest of the subjects had minimal/no experience with the system. As can be seen from Fig. 4, Subject 1 was able to obtain slightly better control performance especially in terms of the variability of the ACRL system's selected control actions. Our observations suggest that improved performance could be achieved with more practice and familiarity with the system.

B. *Would this approach work for amputee subjects?*

An amputee subjects' capacity to generate control commands at will is often a major constraint in developing robust control algorithms for prostheses. Though there are significant differences in the EMG signals obtained from the residual limb of amputees and a healthy biological limb, as our method relies only on the ability of the subject to perform single-joint myoelectric control, with straightforward extensions to multi-joint pattern recognition, any user capable of using current clinical myoelectric control solutions could potentially benefit from the ACRL LfD approach presented here. Prior work by our group has also shown that temporal-difference-learning-based methods as used here operate in similar ways between subjects with and without amputations, especially when primary EMG control is performed via direct proportional mappings [8], [9]. In our experiments, the RL agent was able to learn an exact sequence of movements without knowing the underlying mechanics of the system or the relationship between EMG signals and desired motion.

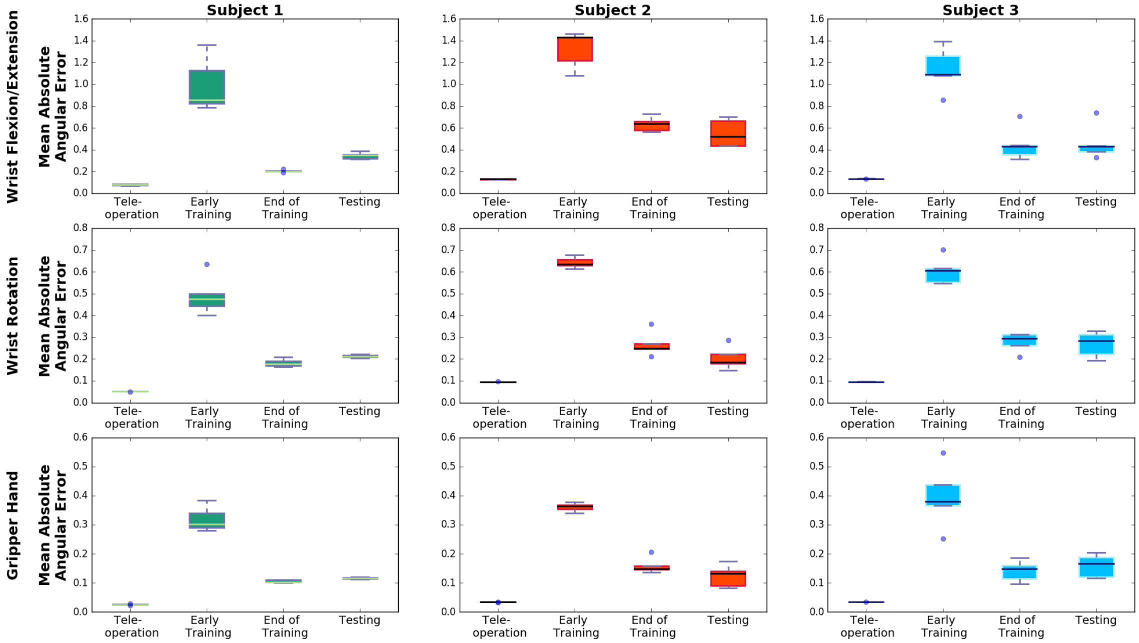


Fig. 4: Comparison of mean absolute angular error accumulated over the course of ~ 45min of learning. Quartile analysis of median values shown over 2 min of learning and testing as compared to direct reactive control for 5 independent runs for each subject. These plots are reflective of the performance of the ACRL learner on this particular task.

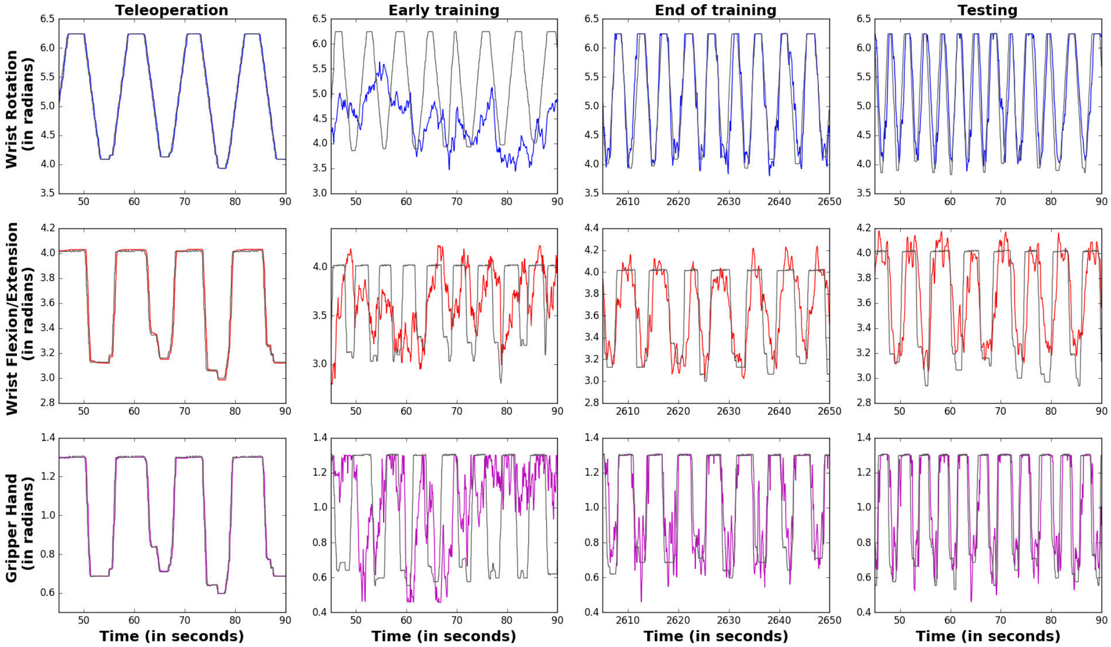


Fig. 5: Comparison of target (grey line) and achieved (colored lines) actuator trajectories over training and testing periods. This plot shows the joint trajectories achieved by the ACRL learner for Subject 1 during training and testing as compared to the offline teleoperation baseline (reactive control).

In part because of this generality, and the ability of RL approaches to optimize to case-by-base prosthetic situations [23], [28], we expect our method to also transfer well to real-world use by amputees with transradial and shoulder disarticulation amputations, assuming reliable physiological control signals can be recorded from the user (i.e., users that could be prescribed myoelectric prostheses). Finally, while we have here considered the case of a unilateral amputation, we could also imagine an occupational therapist using our approach to demonstrate single-arm or bimanual training movements during occupational therapy for bilateral amputation and myoelectric control.

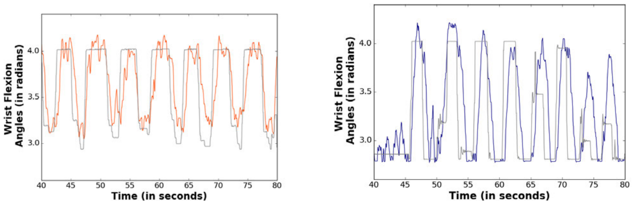
C. *Extensions to context-dependent motion*

When any muscle or muscle group is activated, the resulting movement is dependent on the context. The relationship between muscle excitation and movement is variable and this variability is context conditioned [1]. In order to achieve situation-dependent movement based on muscle excitation, the control system should be given the relevant contextual information and meta-data about the user, the robotic limb and its environment. Modern day prostheses could receive a huge density of data about the user, their physiological and psychological needs and their environment. For example, camera data or even additional sensors on the socket of a prosthesis can readily provide enough contextual information to allow an ACRL system to produce varied motor synergies in response to similar EMG signals from the user—e.g., a system can use additional sensor and state information to help manage the user's degree-of-freedom problem, generating synergies that artfully align to different situations in the user's daily life. It is therefore important that efficient ways of structuring prosthetic data are developed to better represent context to a machine learning prosthetic control system without facing the curse of dimensionality [29].

Our approach is able to produce context-dependent motion if presented with contextually relevant representations. While we have shown results for coarse movements involving three DoFs, there is no algorithmic barrier to adding additional DoFs or DoCs such that a dexterous manipulator could be capable of finer movements and sophisticated multi-actuator grasp synergies. Using an intact limb to train a prosthetic hand for context-specific manipulation with multiple digits is the subject of ongoing work.

D. *Transferability of results*

In order to test the effectiveness of the learned control policy over prolonged use, the control policy learned from the initial training period was stored for future use. Another testing session was conducted after a week to evaluate the performance of the learned control policy on this new data. As can be seen in Fig. 6, the control policy managed to pick actions that achieve the desired trajectory, but it does visibly overshoot in a few regions. It can be seen from Fig. 6b that the actual trajectory overshoots considerably around 70 to 80 seconds. Though the system does capture the intent of the user, it doesn't encapsulate the finer movements of the user. We attribute this deterioration in performance in part to differences in EMG gains and sensor positions between visits. We believe that training the system with a larger dataset (training over multiple sessions, with and without muscle fatigue) could possibly alleviate some of these issues.



(a) Testing on Day 1

(b) Testing after a week

Fig. 6: Comparison of target (grey line, θ^*_{wf}) and learned (coloured lines, θ_{wf}) angles for control on different days.

E. *Why reinforcement learning?*

Bernstein assumed that a large part of the development of motor control involved learning, and that such learning was accomplished through an active search involving gradient extrapolation by probabilistic sampling so that each attempt is informed by previously acquired information about how and where the next step must be taken [2](p. 161). RL has been identified as a promising approach to learn from incremental experience and discover successful decision-making policies through the pursuit of reward.

By shaping these rewards, we can engineer behavior. Though we use a position-based similarity measure in our experiments, it is possible to use other non-trivial measures such as patient satisfaction (e.g., face-valuing methods [30]), torque or velocity matching, minimal power consumption, and others. We could imagine using a combination of these measures as an ACRL reward function to satisfy numerous goals. Mathewson et al. have further explored control learning methods using human-generated rewards and the robustness of these learning methods with stochastic reward signals [28]. Optimizing multiple objectives is often challenging using other approaches like supervised learning.

F. *Extending to applications beyond upper-limb prostheses*

While our approach was tested on the myoelectric control domain, it is widely applicable to other human-computer interaction tasks where the degrees of freedom problem exists. Researchers are currently looking at developing a robotic limbs attached to the human body (also known as supernumerary limbs) that can assist the human user during laborious tasks which cause discomfort and fatigue or when involved in tasks in dangerous environments. Asada et al. developed supernumerary limbs that assist a human user while installing ceiling panels in an airplane [31]. In our work we studied the scenario where an intact limb teaches a prosthetic limb different movement patterns. A user could also teach supernumerary limbs a desired behavior in the same way. Our approach is equally applicable to other domains like lower-limb gait training, lower-limb prostheses, powered orthotics, exoskeletons, and functional electrical stimulation systems.

VI. CONCLUSIONS

This work presented an ARCL LfD framework that will potentially allow an amputee to use their non-amputated arm to teach their prosthetic arm how to move in a natural and coordinated fashion. To our knowledge, this study is the first demonstration of the training an upper-limb myoelectric prosthesis with a user's contralateral limb. We show that an ACRL learner can observe patterns of movement provided by a user and use these demonstrations in learning so as to generate accurate hand and wrist synergies during testing and free-form control by a user. Though our experiments were limited to motions involving three DoFs, our approach could be easily extended to incorporate more DoFs and finer motions. Ideally, we imagine someone with an amputation could use a LfD approach to continue to train a powered prostheses at home on an ongoing basis. While our approach is designed for upper-limb prosthetic control, we expect that it can be easily extended to other human-computer interaction tasks where the degrees of freedom problem exists. In the long run, we expect these methods to improve the quality of life for people with amputations by providing them better ways of communicating their intentions and goals to their myoelectric prosthesis.

ACKNOWLEDGEMENT

We thank Craig Sherstan, Kory Mathewson, Michael Rory Dawson, Jaden Travnik, Vivek Veeriah and Sai Vemula for many helpful discussions.

REFERENCES

[1] M. T. Turvey, H. L. Fitch and B. Tuller, "The Bernstein perspective, I: The problems of degrees of freedom and context-conditioned variability," in *Human motor behaviour: An introduction,* Ed. J.A.S. Kelso. Hillsdale, NJ: Lawrence Erlbaum Associates, 1982.

[2] N. Bernstein. *The coordination and regulation of movements.* Oxford, England: Pergamon Press, 1967

[3] A. dAvella, A. Portone, L. Fernandez, and F. Lacquaniti, "Control of fast-reaching movements by muscle synergy combinations," *J. Neurosci.,* vol. 25, no. 30, pp. 7791–7810, 2006.

[4] B. Peerdeman, D. Boere, H. Witteveen, et al., "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *J. Rehab. Res. Dev*, vol. 48, no. 6, 2011, pp 719–738.

[5] M. M. Bridges, M. P. Para, and M. J. Mashner, "Control system architecture for the Modular Prosthetic Limb," *Johns Hopkins APL Tech. Dig.*, vol. 30, no. 3, pp. 217–222, 2011.

[6] P. M. Pilarski and J. S. Hebert, "Upper and lower limb robotic prostheses," in *Robotic Assistive Technologies: Principles and Practice*, Eds. P. Encarnacao and A. M. Cook, pp. 99–144. Boca Raton, FL: CRC Press, 2017. ISBN: 978-1-4987-4572-7.

[7] R. Merletti and P. Parker, "Control of powered upper limb prostheses," *Electromyography: Physiology, Engineering, and Non-Invasive Applications*, vol. 1, Wiley-IEEE Press, 2004, pp. 453–475.

[8] A. L. Edwards, M. R. Dawson, J. S. Hebert, et al., "Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching," *Prosthetics & Orthotics International*, vol. 40, no. 5, pp. 573–581, 2016.

[9] A. L. Edwards, "Adaptive and autonomous switching: Shared control of powered prosthetic arms using reinforcement learning," M.Sc. Thesis, University of Alberta, 2016.

[10] E. Scheme and K. B. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *J. Rehab. Res. Dev.,* vol. 48, no. 6, pp. 643–660, 2011.

[11] C. Castellini, P. Artemiadis, M. Wininger, et al., "Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography," *Frontiers in Neurorobotics*, vol. 8, no. 22, 2014.

[12] J. Hahne, F. BieBmann, N. Jiang, et al., "Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control," in *IEEE Trans. Neural Syst. Rehab. Eng.,* vol. 22, no. 2, pp. 269–279, 2014.

[13] A. Gijsberts and B. Caputo, "Exploiting accelerometers to improve movement classification for prosthetics," in *Proc. IEEE Int. Conf. on Rehab. Robotics (ICORR)*, Seattle, WA, June 24–26, 2013, pp. 1–5.

[14] M. Atzori, H. Mller and M. Baechler, "Recognition of hand movements in a trans-radial amputated subject by sEMG," in *Proc. IEEE Int. Conf. on Rehab. Robotics (ICORR)*, Seattle, WA, June 24–26, 2013, pp. 1–5.

[15] N. Jiang, S. Dosen, K. R. Muller and D. Farina, "Myoelectric control of artificial limbs—Is there a need to change focus? [In the Spotlight]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 152–150, 2012.

[16] T. Pistohl, C. Cipriani, A. Jackson, and K. Nazarpour, "Abstract and proportional myoelectric control for multi-fingered hand prostheses," *Annals Biomed. Eng.,* vol. 41, no. 12, pp. 2687–2698, 2013.

[17] M. Ison, I. Vujaklija, B. Whitsell, D. Farina and P. Artemiadis, "High-density electromyography and motor skill learning for robust long-term control of a 7-DoF robot arm," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 24, no. 4, pp. 424–433, 2016.

[18] P. M. Pilarski, R. S. Sutton, and K. W. Mathewson, "Prosthetic devices as goal-seeking agents, in *Second Workshop on Present and Future of Non-Invasive Peripheral-Nervous-System Machine Interfaces: Progress in Restoring the Human Functions (PNS-MI),* Singapore, Aug. 11, 2015. 4 pages.

[19] B. Argall, S. Chernova, M. Veloso and B. Browning , "A survey of robot learning from demonstration," in *Robotics and Autonomous Systems*, vol. 67, pp. 469–483, 2009.

[20] M. R. Dawson, C. Sherstan, J. P. Carey, et al., "Development of the Bento Arm: An improved robotic arm for myoelectric training and research, in *Proc. of MEC'14: Myoelectric Controls Symposium*, Fredericton, New Brunswick, August 18–22, 2014, pp. 60–64.

[21] P. M. Pilarski, T. B. Dick, and R. S. Sutton, "Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints," in *Proc. IEEE Int. Conf. on Rehab. Robotics (ICORR)*, Seattle, USA, June 24–26, 2013. 8 pages

[22] P. M. Pilarski, M. R. Dawson, T. Degris, et al., "Adaptive artificial limbs: A real-time approach to prediction and anticipation," in *IEEE Robotics & Automation Magazine*, vol. 20, no. 1, pp 53–64, 2013.

[23] P. M. Pilarski, M. R. Dawson, T. Degris, et al., "Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning," in *Proc. IEEE Int. Conf. on Rehab. Robotics (ICORR)*, June 29–July 1, Zurich, Switzerland, pp. 134–140, 2011

[24] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction.* MIT Press Cambridge, Massachusetts; London, England, 1998.

[25] H. Benbrahim and J. A. Franklin, "Biped dynamic walking using reinforcement learning," *Robot. Auton. Syst.,* vol. 22, pp. 283–302, 1997.

[26] J. Peters and S. Schaal, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7–9, pp. 1180–1190, 2008.

[27] T. Degris, P. M. Pilarski, and R. S. Sutton, "Model-free reinforcement learning with continuous action in practice," in *Proc. of the 2012 American Control Conference (ACC)*, June 27–29, 2012, Montreal, Canada, pp. 2177–2182, 2012.

[28] K. W. Mathewson and P. M. Pilarski, "Simultaneous control and human feedback in the training of a robotic agent with actor-critic reinforcement learning," *2016 IJCAI Workshop on Interactive Machine Learning*, New York, July 9th, 2016.

[29] J. Travnik and P. M. Pilarski, "Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective Kanerva coding," *Proc. IEEE Int. Conf. on Rehab. Robotics (ICORR)*, in press, 2017.

[30] V. Veeriah, P. M. Pilarski, and R. S. Sutton, "Face valuing: Training user interfaces with facial expressions and reinforcement learning," in *IJCAI Workshop on Interactive Machine Learning*, New York, July 9th, 2016.

[31] B. L. Bonilla and H. H. Asada, "A robot on the shoulder: Coordinated human-wearable robot control using Coloured Petri Nets and Partial Least Squares predictions," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA),* Hong Kong, 2014, pp. 119–125.