To appear: *Proc. of the 6th IEEE RAS/EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob2016)*, June 26-29, 2016, Singapore. 

# Machine Learning and Unlearning to Autonomously Switch Between the Functions of a Myoelectric Arm

Ann L. Edwards[1,2,3], Jacqueline S. Hebert[2,4], and Patrick M. Pilarski[2,3,*]

***Abstract*— Powered prosthetic arms with numerous controllable degrees of freedom (DOFs) can be challenging to operate. A common control method for powered prosthetic arms, and other human-machine interfaces, involves switching through a static list of DOFs. However, switching between controllable functions often entails significant time and cognitive effort on the part of the user when performing tasks. One way to decrease the number of switching interactions required of a user is to shift greater autonomy to the prosthetic device, thereby sharing the burden of control between the human and the machine. Our previous work with adaptive switching showed that it is possible to reduce the number of user-initiated switches in a given task by continually optimizing and changing the order in which DOFs are presented to the user during switching. In this paper, we combine adaptive switching with a new machine learning control method, termed *autonomous switching*, to further decrease the number of manual switching interactions required of a user. Autonomous switching uses predictions, learned in real time through the use of general value functions, to switch automatically between DOFs for the user. We collected results from a subject performing a simple manipulation task with a myoelectric robot arm. As a first contribution of this paper, we describe our autonomous switching approach and demonstrate that it is able to both learn and subsequently unlearn to switch autonomously during ongoing use, a key requirement for maintaining human-centered shared control. As a second contribution, we show that autonomous switching decreases the time spent switching and number of user-initiated switches compared to conventional control. As a final contribution, we show that the addition of feedback to the user can significantly improve the performance of autonomous switching. This work promises to help improve other domains involving human-machine interaction—in particular, assistive or rehabilitative devices that require switching between different modes of operation such as exoskeletons and powered orthotics.**

## I. INTRODUCTION

New devices used in human-machine interaction are designed with elegant, efficient parts that allow them to be multi-purpose—they perform a broad range of functions to accomplish their users' many goals. In particular, modern myoelectric artificial arms (arms that can be actuated via electromyography, or EMG signals [1]) are highly versatile. The most recent generation of commercial myoelectric arms are capable of upwards of 20 different grip patterns (hand positions) and joint motions. Despite their increasingly diverse set of functions, these arms are often considered by persons with amputations to be challenging to operate, largely because of the non-intuitive solutions in place to control so many different available motions [1]–[3]. A common type of control interface is called switched, or gated control. Using this control method, arm functions (which we define as either grip patterns or individual joint motions) are presented to the user as a optimized list with an order that never changes. The individual can then switch through this static list using a mechanical toggle, muscle co-contraction, a force-sensitive resistor, or other similar methods, until they select their desired function; the selected function can then be controlled using normal muscle contractions [1], [4]–[6]. A consequence of using this type of control is that amputees can select and control only one function at a time, which makes the use of the switched control interface cognitively demanding and time intensive [1], [2]. As a result, the number of controllable functions on a prosthetic arm is often deliberately reduced to only a few clinically selected options that are customized to the anticipated needs of the user.

We propose to instead preserve a user's access to the diverse functions of a myoelectric arm by reducing both the complexity of their switching choices and the frequency of required switches or other manual interactions. One way to decrease the number and complexity of switching interactions is to shift greater autonomy to the prosthetic device, thereby sharing the burden of control between the human and the machine [7]. Autonomy in both switched and non-switched myoelectric control settings is a topic of current interest, with a number of compelling approaches ranging from enhanced pattern recognition to unsupervised control adaptation, autonomous grasp pre-shaping, and intelligent object targeting (as reviewed by Castellini et al. [7]). Specifically with regard to switched myoelectric control interfaces, previous work done in our research group has demonstrated that reinforcement learning (RL) techniques can be successfully applied to increase the decision making capacity and autonomy of prosthetic control solutions [8]–[10]. Our demonstrations have leveraged real-time *nexting*, a computational approach so-named because it allows a system to build up and maintain predictions about what will happen next (c.f., Modayil et al. [11] and White [12]).

Through the use of nexting, and specifically an RL technique called general value functions (GVFs, [12], [13]), we previously developed a novel switching control method for myoelectric arms called *adaptive switching* [14]; this

This work was supported by the Alberta Innovates Centre for Machine Learning (AICML), Alberta Innovates – Technology Futures (AITF), and the National Science and Engineering Research Council (NSERC).

[1]Faculty of Rehabilitation Medicine, University of Alberta, Edmonton, AB, Canada; [2]Division of Physical Medicine & Rehabilitation, Dept. of Medicine, University of Alberta, Edmonton, AB, Canada; [3]Alberta Innovates Centre for Machine Learning (AICML) and the Reinforcement Learning and Artificial Intelligence Laboratory (RLAI), University of Alberta, Edmonton, AB, Canada; [4]Glenrose Rehabilitation Hospital, Alberta Health Services, Edmonton, AB, Canada.

* Please direct correspondence to pilarski@ualberta.ca

approach was validated in pilot studies with both amputee and non-amputee participants [9], [14]. Adaptive switching reorders a list of arm joints for the user on the basis of their predicted likelihood of being used in a given situation. For example, if joint A's predicted movement is greater than the prediction of movement for joint B, joint A will be presented first to the myoelectric arm user when they next initiate a switch to an alternate joint. This reordering is a dynamic process that occurs continuously and in real time as long as the arm is in use. Adaptive switching was shown to improve the control time and number of switches required to do a complex task with a robotic arm, as compared with switching using the conventional static list of functions (which we here denote non-adaptive switching) [14]. When using adaptive switching, subjects performing the task had to switch fewer times to select their desired function, and experienced myoelectric control users required significantly less time to complete the task.

### A. *Motivation and Contributions of the Present Work*

Although adaptive switching has the effect of reducing the number of interactions (switches) required of a user, for any given task there remains a minimum number of switches that a user must perform. This is because each time a user wishes to control a different joint—even with a joint list built out of perfect prediction accuracy—he or she is required to switch at least once. As a consequence, greater attention and time is needed for switching between control channels of the arm that otherwise could be spent on accomplishing the task. We expect further reductions in task time and effort could be gained by removing altogether the need to perform manual switching interactions in common or highly consistent situations.

In this paper, we therefore introduce a novel control method, termed *autonomous switching*, that works in parallel with adaptive switching to further reduce the number of explicit switching interactions required of a prosthetic user. Autonomous switching automatically switches between the joints of a myoelectric arm on the basis of the predicted likelihood that the user will switch to a different joint than the one that is currently selected. Like adaptive switching, autonomous switching uses GVFs to build up and maintain predictions about signals relating a user's control intentions and behaviour. Our autonomous switching algorithm is designed in such a way that a myoelectric arm is able to learn to switch autonomously, and subsequently unlearn the behaviour should the user's intent or needs change. Ensuring the user has the ability to correct the arm's behaviour through unlearning is important for daily life, in which the environment and the task are constantly shifting. Further, we expect that providing cues or feedback to the user about forthcoming automatic actions of the control system will help streamline the use of autonomous switching. Feedback, in the form of tactile cuing, is therefore an important element of our algorithm's practical implementation.

In the remainder of this article, we describe our algorithm for autonomous switching and present results from two experiments that show the practical learning and unlearning of autonomous switching behaviour during myoelectric control by an able-bodied subject. The first experiment demonstrates how autonomous switching can be both learned and then unlearned when a user's intent has changed. In the second experiment, we compare autonomous switching with previously tested control methods, and demonstrate the effect that tactile cuing has on a user's performance.

## II. METHODS

### A. *Experimental Setup*

The robotic arm used in the experiments described in this paper is called the Bento Arm: a custom-designed myoelectric arm with the proportions of a human arm (Fig. 1) [15]. The Bento Arm has five degrees of freedom, or functions, which a user can switch between and which can be moved like individual joints: shoulder, elbow, wrist rotation, wrist flexion, and gripper (i.e., hand open/close). Every time a switch is made, an audible cue alerts the subject to which joint was selected. Control by the user was effected so as to closely replicate one of the most challenging clinical control cases: a single control channel with a single momentary trigger (e.g., a button, or pulsed myoelectric co-contraction [6]) to switch the one channel of control between the multiple functions of a device. In Experiment 1, the arm was controlled with an Xbox 360 controller. A button on the right of the controller was programmed to switch between joints, and the left thumb joystick was used to move the arm bidirectionally. In Experiment 2, the arm was controlled using EMG electrodes. One pair of surface EMG electrodes was attached over a subject's wrist flexor muscles and another pair over the wrist extensor muscles (both sets on the same side). These electrodes provided the control signals for moving a selected joint back and forth. A separate pair of electrodes were attached over the opposite wrist extensors, which provided the signal for switching between the joints of the arm. EMG signals were acquired using an 8-channel

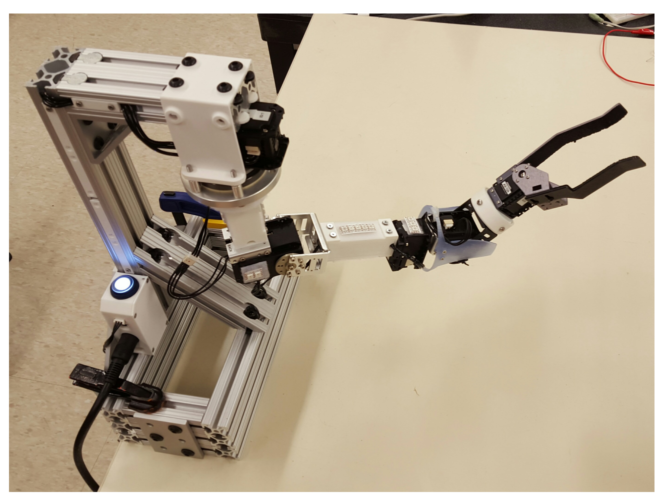


Fig. 1. The Bento Arm, a custom-designed robot arm, was controlled by able-bodied subjects in two separate experiments. The Bento Arm has five joints that can be controlled via EMG signals.

Delsys Bagnoli system. Able-bodied subjects were recruited and gave informed consent in accordance with the study's Human Research Ethics Board approval.

### *B. Learning Algorithms and General Value Functions*

Our approach to reducing switching interactions uses nexting in the form of GVFs to build up predictions about sensorimotor signals at various timescales [13]. Like conventional RL value functions, GVFs represent temporally extended predictions—they represent the weighted summation of some observable signal over a window of future experience [11]–[13]. While conventional value functions approximate the accumulated reward observed by an RL system, GVFs can be used to build up predictions about any arbitrary signal of interest [13]. In this way, GVF predictions may be considered answers to different types of questions, such as "when will a myoelectric arm user move the elbow joint?" As noted in past work, GVFs have proved to be well suited to an ongoing prosthetic control setting [8]; using standard reinforcement learning methods, GVFs may be learned from an ongoing stream of data without labeled training examples and explicit training periods (in contrast to supervised learning), and are computationally efficient to both learn and query for predictions in real time [8].

In precise terms, general value functions are functions that map a given state $S$ or *state approximation* (typically a binary column vector, here denoted $\mathbf{x}(S)$) to the expectation of a weighted summation of a signal of interest (called a *cumulant*, $Z$, [12]) when starting from state $S$ and thereafter following a given *policy* (defined as a way of behaving, or choosing actions $A$ given $S$) [11]–[13]. The temporal weighting of the cumulant is specified by a state-conditional discounting parameter $\gamma(S) \in [0,1]$ [12]. The learned expectation of the cumulant is represented for each GVF using a column vector of weights denoted $\mathbf{w}$. Predictions can then be computed for any given state via the inner product of the state representation, $\mathbf{x}(S)$, and a GVF's weights, $\mathbf{w}$.

In our experiments with adaptive and autonomous switching, we framed two distinct questions using GVFs:

1) Which joint, grasp, or function of a robotic arm will a user want to move next? (For adaptive switching.)
2) At a given moment, is the user going to initiate a switch to a different joint than the one that is currently selected? (For autonomous switching.)

These questions were answered by updating GVF weights using standard RL algorithms: in adaptive switching we used the *temporal-difference learning* algorithm, TD($\lambda$) [16], and in autonomous switching we used an off-policy learning method known as *gradient temporal-difference learning*, GTD($\lambda$) [17]. An overview of the TD($\lambda$) and GTD($\lambda$) algorithms will be presented in the respective sections below.

### *C. Adaptive Switching*

Adaptive switching was implemented as described in previous work [14] using GVFs and TD($\lambda$). We briefly summarize that implementation here, along with the relevant learning mechanisms, state representation choices, and experiment-specific parameter settings. As noted in Sec. I, adaptive switching uses the relative magnitude of predictions about joint movement to reorder the switching list for the user according to prediction magnitude, answering on a moment-by-moment basis the question "which joint will the user move next?" In order to provide these predictions, five GVFs were initialized at the start of each experiment—one for each controllable joint of the robot arm. Each GVF's cumulant $Z$ was a binary value indicating whether or not its respective joint was currently in motion, such that each GVF learned to predict the movement of a different joint.

TD($\lambda$) was used to update the weights of the system's five GVFs. In TD learning, a temporal-difference error signal, denoted $\delta$, is calculated in (1) as the difference between the prediction for the current state and the prediction for the future—i.e., the cumulant plus the prediction for the next observed state as discounted by $\gamma$. According to standard practice, replacing eligibility traces [16], $\mathbf{e}$, were computed as in (2), using a trace decay parameter $\lambda$, to specify the eligibility of features for weight updates. (For a full discussion of eligibility traces and their utility, please refer to Sutton and Barto [16].) The weight vector was then updated by adding the previous timestep's weight vector to the product of $\delta$, a scalar step size $\alpha$, and $\mathbf{e}$ (3).

$$\delta_t = Z_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}(S_{t+1}) - \mathbf{w}_t^\top \mathbf{x}(S_t) \tag{1}$$

$$\mathbf{e}_t = min(\mathbf{e}_{t-1}\gamma\lambda + \mathbf{x}(S_t), 1) \tag{2}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta_t\mathbf{e}_t \tag{3}$$

When used as updates, (1)–(3) represent the steps involved in learning GVFs that are repeated at each time step. In adaptive switching trials, the values of $\lambda$ and $\alpha$ were fixed at 0.99 and 0.0025, respectively, $\mathbf{w}$ and $\mathbf{e}$ were initialized to 0, and the discount factor $\gamma$ was set to a constant value of 0.9 (i.e., a exponentially decaying summation of the cumulant over a horizon of approximately 10 steps into the future).

Learning and control occurred at 20 Hz (hardware determined) on two laptop computers. At each timestep, the system observed signals from the myoelectric arm. These signals comprised relevant information about each joint of the Bento Arm as well as EMG signals from the user controlling the arm. These signals were used to form the state space for the leaning system. Tile-coding function approximation [16] was used as the function $\mathbf{x}(S)$. The state space $S \in \mathbb{R}^{11}$ consisted of ten signals relating to the angular movement of each joint (the position and velocity of each servo motor) and grip strength information from the gripper joint (measured in terms of current load). The range of each observed signal was normalized and divided into discrete units called tiles (n=6, empirically chosen), with four overlapping tilings that were randomly offset as per standard practice [16]. Tile-coding resulted in a sparse binary array with $\sim$1.4B features that was reduced to a more compact vector of 1M features through the use of hashing.

### D. Autonomous Switching

The algorithm for autonomous switching was also based on the temporal-difference learning of GVFs. Unlike adaptive switching, wherein the control system's objective was to learn *what* joint a user will select next, in autonomous switching the control system is trying to learn *when* a user will switch between joints. In more precise terms, the control system aims to learn a computational answer to the following question: "If the system were to switch autonomously at the current moment, would the user move the newly selected joint or manually correct to continue using the previous joint?" Should the system's prediction be strong enough that the user would accept and utilize the new joint, the system could then autonomously switch for the user. For example, when the user is reaching for an object on a table, if the system predicted that the user would readily begin closing their hand if it switched their control to a grasping actuator, an autonomous switch would occur. This behaviour—and the predictions it is based on—can be learned directly by watching the user perform the task, as in adaptive switching.

However, the *when* question posed by autonomous switching cannot be practically learned using the same TD methods as adaptive switching. In autonomous switching, one GVF per joint is initialized to predict what would be the outcome *if* the system switched for the user at a given moment. Because this GVF question relates to a switching behaviour (a policy of switching, not switching, or moving a joint in a given state) different than the one the user may be currently be pursuing, GVF learning requires an off-policy algorithm called gradient temporal-difference learning, or GTD($\lambda$) [17]. Off-policy algorithms use information about a policy that is being followed (the *behaviour policy*, or how the user is actually controlling the limb) to learn about a similar but different policy (the *target policy*, here defined as "always switch for the user if not currently switching").

Computationally implementing the autonomous switching GVF question posed above, we chose the cumulant $Z$ to be a binary signal that took a value of 1 for one timestep at the end of a switching sequence only if the user moved a different joint than the joint that was active at the start of switching. For instance, if a user switched from the elbow to a different joint and moved the selected joint, the cumulant would momentarily be 1, indicating the user's intention to switch to a new joint. Alternatively, if a user initiated a switch sequence, switching from the elbow and back again to the elbow, the cumulant would remain 0 at the moment motion resumed, indicating the intention to continue using the same joint. Further, to fully specify the predictive question being asked in autonomous switching, we use a state-conditional discounting factor $\gamma(S)$ where $\gamma = 1$ from the beginning of a switching sequence until the moment a new joint was selected and moved, and $\gamma = 0$ otherwise.

This situation-conditional way of designing the GVF question for autonomous switching provides a way for users to correct the switching behaviour of the arm and direct its learning when there are errors in training, changes to the environment, or to the user's intent. As in adaptive switching, a user's acceptance of the system's automatic switching behaviour provides *implicit reinforcement* for the system's choice to switch. If a system performs an autonomous switch in a state $S_t$ at time $t$, movement by the user of a new joint following autonomous switching will result in $Z = 1$ at time $t + n$, i.e., $n$ steps later at the end of the switching sequence. Because $\gamma = 1$ during a switching sequence, the resulting $\delta$ will be applied directly to $S_t$, modifying $\mathbf{w}$ and strengthening the system's prediction about switching in the state where it initiated the autonomous switch. Conversely, a user switching back to their original joint will result in $Z = 0$, decreasing the system's probability of switching by lowering its switching predictions for $S_t$.

In the GTD($\lambda$) algorithm, $\delta$ is computed as in (1). However, equations (4) and (5) for updating the eligibility traces and the weight vector differ from those of the standard TD($\lambda$) algorithm shown above. To compare the similarity of the behaviour and target policies used in action selection, an importance sampling ratio, $\rho$, is formed by dividing the target policy, $\pi$, by the behaviour policy, $\mu$, according to $\rho_t = \frac{\pi(S_t, A_t)}{\mu(S_t, A_t)}$. In autonomous switching experiments, the behaviour policy (how the user is switching) matches the target policy ("always switch if not switching") only from the start of a switching sequence until the moment the user selects and begins moving a new joint, leading to $\rho = 1$ during this time and $\rho = 0$ otherwise. In (4), replacing eligibility traces are now multiplied by this importance sampling ratio. To ensure stability during off-policy learning, GTD($\lambda$) also uses a second weight vector $\mathbf{h}$, initialized to zero, to correct the gradient of the learning update in (5) [17]. This second weight vector $\mathbf{h}$ is updated as in (6), with $\beta$ as a step size parameter for the update.

$$\mathbf{e}_t = min(\rho_t(\mathbf{e}_{t-1}\gamma\lambda + \mathbf{x}(S_t)), 1) \tag{4}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(\delta_t \mathbf{e}_t - \gamma(1-\lambda)\mathbf{x}(S_{t+1})\mathbf{e}_t^\top \mathbf{h}_t) \tag{5}$$

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \beta(\delta_t \mathbf{e}_t - (\mathbf{h}_t^\top \mathbf{x}(S_t))\mathbf{x}(S_t)) \tag{6}$$

In all autonomous switching experiments, $\alpha$ was set to 0.025, $\beta$ was set to 0.000025, $\lambda$ was set to 1, and all other values were initialized as in adaptive switching. To form a feature vector for the leaning agent we again used tile-coding function approximation as above, but now with a state $S \in \mathbb{R}^{13}$ comprised of the position and velocity of all five joints, torque on the gripper, and a moving average of the two EMG channels responsible for joint motion (13 inputs). The total number of features was $\sim$52B and was again reduced to an $\mathbf{x}(S)$ of $\sim$1M features (with four active features) via hashing.

Autonomous switches were initiated when switching predictions exceed a fixed threshold. The threshold for autonomous switching was empirically determined and fixed for all experiments at 0.3, though in practice this value could be determined in a state or subject conditional way using measures of predictive confidence (e.g., Sherstan et al. [19]).

A separate GVF was initialized for each of the five joints of a switching sequence, with the prediction of the last-moved joint's GVF being evaluated against the threshold to trigger switching. We chose to limit autonomous switching to times when the arm was not moving and the user had not yet switched during a switching sequence.

### E. Tasks

The task chosen for Experiment 1 to assess unlearning was a task similar to the simple task used in previous studies [10], [14]. The subject, who was able-bodied and an experienced operator of the Bento Arm, controlled the arm with the Xbox controller. The task was divided into two phases. In the first phase, the subject was instructed to open and close the gripper with the shoulder on the far left side of its angular range, rotate the shoulder to the far right side of its range, open and close the gripper again, and rotate the shoulder back to its starting position. Throughout the entire task, both adaptive switching and autonomous switching were enabled—thus, the joint list was continually being reordered, and simultaneously, the arm was learning when to switch autonomously. The first phase of the task was repeated until the arm began to switch autonomously at each instance of a switch. At this point, the task was altered slightly for the second phase: the gripper was only opened and closed on the left side of the shoulder's range. The second phase of the task was repeated in this way five times.

In Experiment 2, another able-bodied subject was asked to control the Bento Arm with EMG to perform a task similar to the first. The subject began the task with the shoulder joint positioned at the limit of its angular range; opened the gripper; moved the shoulder to the opposite side; and then closed the gripper. This motion was repeated for a total of three minutes. The total number of switches (manual and automatic) and the total time spent switching were recorded by a computer connected via USB to the experimental arm. We also recorded the number of switches and amount of time in each switching sequence, where switching sequences were defined as periods of time beginning when a switch occurred following motion and ending when a same or different joint was selected and then moved.

The subject performed the three-minute experiment using four different control methods: using a non-adaptive switching order (static switching list), using an adaptive switching order, and using an adaptive switching order combined with autonomous switching (with and without feedback about upcoming autonomous switches). Switching feedback consisted of a light vibration on the subject's forearm. The frequency of the vibration feedback was proportional to the magnitude of the prediction signal for autonomous switching. An LED also indicated to the user and the experimenter when an autonomous switch occurred. Our hypothesis was that a user will be better able to collaborate with an autonomous system when the system's intent to switch is made clear to the user via feedback. Autonomous switching creates a shared control scheme between the system and the user. For shared control to be effective, it requires bidirectional communication: direct communication through a user's actions and indirect communication of system knowledge and predictions through feedback (c.f., Parker et al. [18]).

Prior to beginning the experiments, the subject was given five minutes to practice controlling the Bento Arm and was briefed on the nature of each of the control methods. The subject completed a total of four trials, each consisting of the four control methods (i.e., 16 x 3min); in each of the four trials, the subject performed the control methods in a semi-random order (i.e. the sequence was the same, but the starting method was randomized).

## III. RESULTS

### A. Experiment 1

Fig. 2 highlights how switching behaviour was learned and unlearned during the course of a single task. The results shown for one full run through the simple task, in which the subject manipulated the Bento Arm to open the gripper on one side, close the gripper on the other side, and rotate the shoulder back and forth. The signal in red represents a prediction of whether the individual controlling the arm is going to switch to a different joint. Binary signals, indicating times during the task in which the individual or the robot arm switched, are shown in blue. The dotted line indicates the threshold that predictions in red need to exceed for the system to switch autonomously.

An alternative view of the same experiment is seen in Fig. 3. Each data point represents the magnitude of the prediction at specific states in the task. Here we defined state visits as times in which the arm was in the same unique state: not moving and within a small fixed range of angular positions corresponding to where the subject was instructed to switch between functions. Green points indicate predictions for when the shoulder actuator reached the rightmost side of the space, and red points indicate predictions at the leftmost side of the space. The dotted line represents the threshold for autonomous switching.

### B. Experiment 2

Fig. 4 compares the four control methods in terms of the time series of switching sequences generated over the 3min testing period for one of the trials, clearly presenting the onset of autonomous switching in Fig. 4d (results are representative of the other three trials). Each repetition of the task required several switching sequences. Depending on the order of the joints in the list, each switching sequence could require up to four switches (or more in the case of user error) to select the desired joint. The results are presented as the number of switches per switching sequence (left) and the respective amount of time required for each switching sequence (right). On the left, the red (dark) bars show the number of manual switches initiated by the subject, whereas the pink (shaded) bars are shown for all switches performed autonomously by the system (autonomous switching).

Fig. 5 shows the average number of switches, the average time spent switching, and the average number of times the task was repeated during the 3min of allotted time. In Fig. 5a,

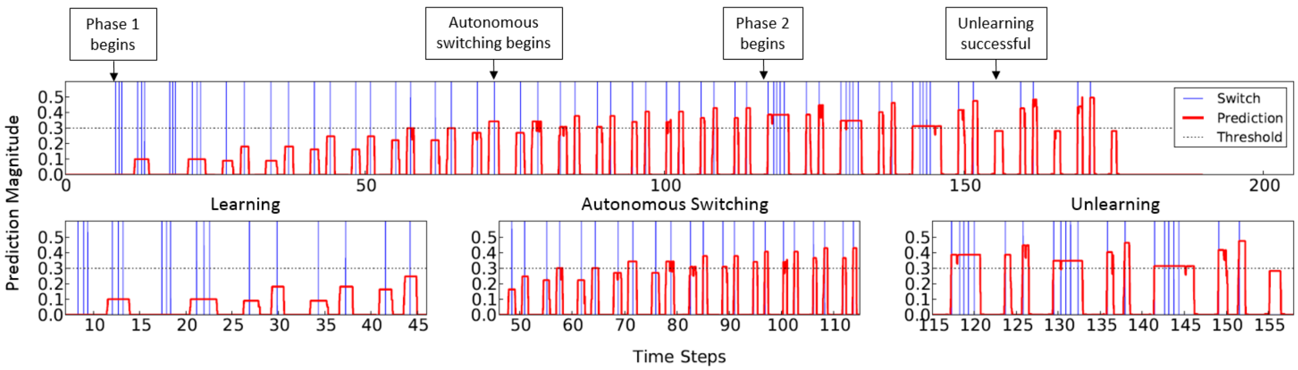



Fig. 2. The Bento Arm learns to switch autonomously during a simple task, then unlearns switching when the task is changed. The signals in red represent predictions about switching to a different joint; the signals in blue represent binary switch signals; and the dotted line is the threshold above which autonomous switching will occur.

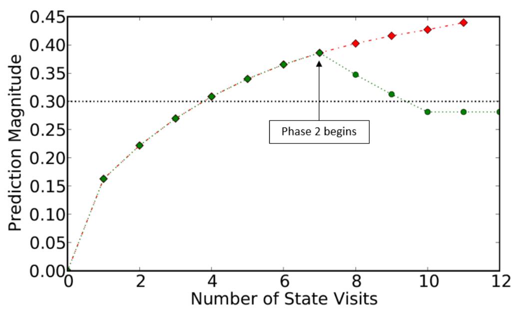
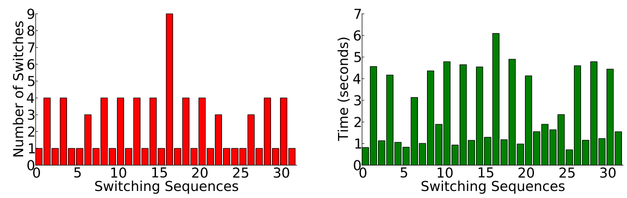
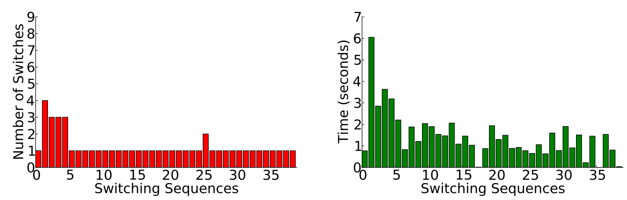


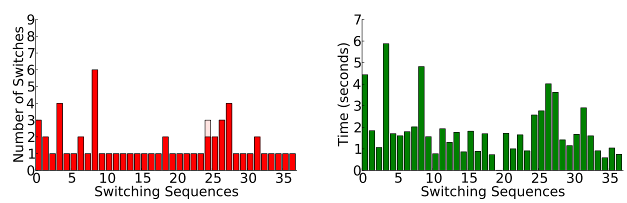

Fig. 3. The magnitude of the switching predictions at similar states throughout the task. The green line represents the prediction magnitude as the Bento Arm's shoulder reaches the right side of its range; the red line represents the prediction magnitude as the shoulder reaches the left side.



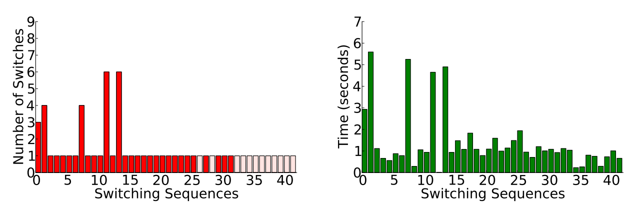

(a) Non-Adaptive




(b) Adaptive




(c) Adaptive + Autonomous (No Feedback)



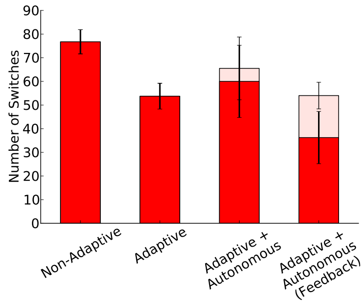

(d) Adaptive + Autonomous (With Feedback)

Fig. 4. An example of switching sequences for each of the control methods in a simple, three-minute task. The number of switches per switching sequence are shown in red; autonomous switches are shown in pink; and the seconds per switching sequence are shown in green.

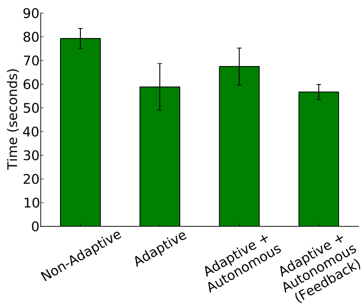
red bars represent the number of manual switches made by the individual controlling the arm for each control method. The height of the pink bars, shown in the 'Adaptive + Autonomous' data sets only, represents the total number of switches (combined total of manual switches and autonomous switches). Fig. 5b is the average amount of time during the task that was dedicated to switching. The data in Fig. 5c represents the average number of times the subject was able to repeat the task fully in the given time. Times the individual completed only part of the task were not counted.

## IV. DISCUSSION

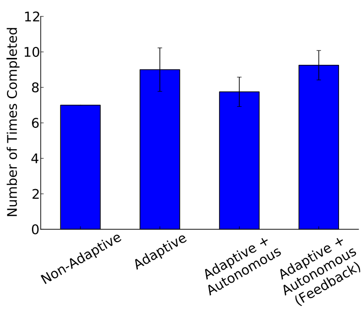
*In this work, our first aim was to demonstrate that it is possible to both learn and unlearn autonomous switching behaviours.* Our results show that learning and unlearning of switching behaviour can be effected during ongoing robot control using GVFs and TD learning. In the left bottom panel shown in Fig. 2, the arm is learning to switch autonomously and the predictions do not yet rise above the threshold of 0.3. The prediction signal exceeds the threshold for autonomous switching for the first time approximately 70 seconds from the start of the experiment, at which point the arm begins to switch autonomously (middle bottom panel). As the second phase of the task begins, at first the arm continues to switch




(a) Mean number of switches for each method. Red bars represent manual switches by the subject; pink shading represents autonomous switches.




(b) Mean time spent switching for each method.




(c) Mean task completions for each method.

Fig. 5. Results for an able-bodied subject completing the 3min task with each of the four control methods, averaged over four semi-randomized trials.

autonomously to the gripper when it reaches the right side of the space, but the user always switches back to the shoulder joint, and therefore the prediction signal for that side begins to decrease (right bottom panel). The user only reinforces the switch back to shoulder three times before the prediction signal falls below the threshold for autonomous switching to occur, and the system has effectively "unlearned" that autonomous switch. At that point, unless the user switches again to a different joint on the right side of the space, the arm will only switch autonomously on the left side.

Fig. 3 clearly demonstrates the relative magnitude of the predictions compared to each other in similar states. It takes four visits to the same state for the system to exceed the threshold and learn to switch autonomously on either side of the shoulder's range. For the first seven state visits, the green and the red lines overlap. This is because the maximal height of the predictions at the left and the right are identical in the first phase, since the subject is performing the same actions on either side. It is only after the seventh consecutive state visit on both sides that the green line diverges from the red line, as the second phase begins and unlearning starts to occur for the state on the right side.

*Our second aim in this work was to demonstrate that autonomous switching can significantly reduce the number of manual switching interactions required of a user as compared to previously presented methods.* This reduction can be seen in both Figs. 4 and 5. Fig. 4a, the non-adaptive case, shows poor performance at the beginning of the task and little improvement over the course of three minutes. Throughout the task, the subject must often switch four times to select a joint. Furthermore, sequences where the subject must switch multiple times translate into much longer switching times, typically exceeding four seconds. These results are also reflected in Fig. 5, in which more switches are made and greater time is spent on average compared to the number of times the subject was able to complete the task.

Unlike the non-adaptive case, in the adaptive experiment performance can be seen to improve with time (Fig. 4b). The number of required switches per switching sequence decreased as a direct result of the adaptive switching list. After a short period of learning by the system, the subject typically did not need to switch more than once to select a joint, and switching sequences often lasted two seconds or less. We see a marked improvement in Fig. 5 as well, where a decrease in switches and time resulted in the subject completing more repetitions of the task. These results are consistent with the adaptive switching results presented in our prior work [14].

Figs. 4c and 4d demonstrate the performance of autonomous switching when used in combination with adaptive switching. *As a key result, autonomous switching with feedback was found to significantly reduce the number of manual switches required of the user*—our primary measure of success. In the first case, autonomous switching without feedback, the subject was observed to make more errors when switching than when feedback was given to them about the system's intent to autonomously switch functions. These errors often arose out of uncertainty over whether the system was going to switch autonomously or not. The amount of time spent on each switching sequence is also greater than when feedback was given. As a result, the system only switched autonomously once in the three-minute time frame. By contrast, when the user received vibration feedback of increasing intensity as the system's prediction rose, autonomous switching dominated the latter part of the experiment. Overall performance was better than that of adaptive switching when averaged across the 3min period in terms of less variability in switching time and significantly fewer manual switches required of the user. For example, in Fig. 4d, there were a total of 12 autonomous switches made and a corresponding decrease in the average time spent on each switching sequence. We expect that further improvements in communication between the user and their device will result in stronger correlations between reduced switching interactions and saved time.

In this work, we implemented a simple task as a first demonstration of autonomous switching. The simple task described allowed us to determine the performance of autonomous switching before scaling up to more complex tasks such as the box-and-blocks task [20]. In our chosen task, autonomous switching with feedback successfully decreased the number of manual switches required while increasing the number of times the task was completed. However, on average, the amount of time spent switching was on par with adaptive switching. Of note, the participant in our second experiment reported that once the system learned to switch consistently, feedback was less essential to the task because he could accurately anticipate when the robot arm would switch for him. This suggests that with practice a subject will build up reciprocal predictions about the system's behaviour, allowing the subject to spend less time verifying the choices of the system. We anticipate that autonomous switching will result in greater time savings and less cognitive demand for longer, more complex tasks, as both user and machine learn to share control more effectively (e.g., in situations where precision is important, or where it is hard for the user to quickly decide on appropriate actions).

Shared control has potential rewards in human-machine interaction. However, in practice, autonomy and the way in which humans use machine autonomy is imperfect. Both humans and machines make mistakes. For this reason, we have demonstrated the ability of our approach to not only learn switching behaviour, but also unlearn autonomy and revert control to the user. The capacity for both learning and unlearning important in a board sense, as there are numerous examples of switching in assistive devices and devices for rehabilitation. For example, exoskeletons and powered orthotics can have multiple modes of operation or gait patterns for different terrains [21], and are strong candidates for the application of adaptive and autonomous switching. We believe our approach will transfer well to other switching-based domains, and that an expanded state representation will allow a smooth translation to multi-task settings and more continuous, daily-life environments.

## V. CONCLUSIONS

This paper presents the first demonstration of real-time learning and unlearning of autonomous switching for prosthetic control. Our results illustrate how autonomous switching in combination with adaptive switching and feedback can reduce the number of manual switches required of a user while operating a myoelectric arm. This points to potential savings in more complex tasks and a reduction in a user's cognitive load while using a prosthesis. Autonomous switching adds impact to adaptive switching, and promises to increase its practical utility in multiple domains. The next step in this work will be a larger study testing the performance of autonomous switching on a more complex task with multiple able-bodied and amputee subjects.

## ACKNOWLEDGMENT

The authors are grateful to Craig Sherstan for his technical assistance and a number of helpful conversations. They also thank Alfred Dao for his assistance with video editing.

## REFERENCES

[1] 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 Rehabil. Res. Dev.*, vol. 48, no. 6, pp. 643–660, 2011.

[2] T. W. Williams, "Guest editorial: Progress on stabilizing and controlling powered upper-limb prostheses," *J. Rehabil. Res. Dev.*, vol. 48, no. 6, pp. ix–xix, 2011.

[3] L. Resnik *et al.*, "Advanced upper limb prosthetic devices: Implications for upper limb prosthetic rehabilitation," *Arch. Phys. Med. Rehabil.*, vol. 93, no. 4, pp. 710–717, 2012.

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

[5] S. Micera *et al.*, "Control of hand prostheses using peripheral information," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 48–68, 2010.

[6] P. Parker *et al.*, "Myoelectric signal processing for control of powered limb prostheses," *J. Electromyogr. Kines.*, vol. 16, no. 6, pp. 541–548, 2006.

[7] C. Castellini *et al.*, "Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography," *Front. Neurorobot.*, vol. 8, no. 22, 2014.

[8] P. M. Pilarski *et al.*, "Adaptive artificial limbs: A real-time approach to prediction and anticipation," *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 53–64, 2013.

[9] P. M. Pilarski *et al.*, "Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots," in *Proc. 4th IEEE RAS EMBS Int Conf. Biomed. Robot. Biomechatron.*, Rome, Italy, June 24–27 2012, pp. 296–302.

[10] A. L. Edwards *et al.*, "Temporal-difference learning to assist human decision making during the control of an artificial limb," in *Proc. 1st Multidiscip. Conf. Reinf. Learn. Decis. Making*, Princeton, NJ, USA, October 25–27, 2013. Available: arXiv:1309.4714 [cs.AI].

[11] J. Modayil, A. White, R. S. Sutton, "Multi-timescale nexting in a reinforcement learning robot", *Adaptive Behavior* vol. 22, no. 2, pp. 146–160, 2014.

[12] A. White, "Developing a predictive approach to knowledge," Ph.D. thesis, Dept. Comp. Sci., Univ. of Alberta, Edmonton, Canada, 2015.

[13] R. S. Sutton *et al.*, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *Proc. 10th Int. Conf. Auton. Agent. Multi. Agent Syst.*, Taipei, Taiwan, May 2–6, 2011, pp. 761–768.

[14] A. L. Edwards *et al.*, "Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching," *Prosthet. Orthot. Int.* Published online ahead of print: September 30, 2015. DOI: 10.1177/0309364615605373.

[15] M. R. Dawson *et al.*, "Development of the Bento Arm: An improved robotic arm for myoelectric training and research," in *Proc. Myoelectric Contr. Symp.*, Fredericton, NB, Canada, 2014, pp. 60–64.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge MA: MIT Press, 1998.

[17] H. R. Maei, "Gradient temporal-difference learning algorithms," Ph.D. thesis, Dept. Comp. Sci., Univ. of Alberta, Edmonton, Canada, 2011.

[18] A. S. R. Parker *et al.*, "Using learned predictions as feedback to improve control and communication with an artificial limb: Preliminary findings," arXiv:1408.1913 [cs.AI], 2014.

[19] C. Sherstan *et al.*, "Introspective Agents: Confidence Measures for General Value Functions," *Proc. Ninth Conference on Artificial General Intelligence (AGI-16)*, New York, July 16–19, 2016, to appear.

[20] V. Mathiowetz *et al.*, "Adult norms for the box and blocks test of manual dexterity," *Am J. Occup. Ther.*, vol. 39, no. 6, pp. 386–391, 1985.

[21] S. Tanabe, S. Hirano, E. Saitoh, "Wearable Power-Assist Locomotor (WPAL) for supporting upright walking in persons with paraplegia," *Neurorehabilitation*, vol. 33, no. 1, pp. 99–106, 2013.