Adaptive and Autonomous Switching: Shared Control of Powered Prosthetic Arms Using Reinforcement Learning

by

Ann L. Edwards

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Rehabilitation Science

Faculty of Rehabilitation Medicine

University of Alberta

Abstract

Powered prosthetic arms with numerous controllable functions (i.e., grip patterns or movable joints) can be challenging to operate. Gated control—a common control method for myoelectric arms and other human-machine interfaces—allows users to select a function by switching through a static list of possible functions. However, switching between many 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.

Previous work has demonstrated that reinforcement learning (RL), and specifically general value functions (GVFs), has the potential to reduce the time and switching cost of gated control methods. In the current work, we extend previous studies by advancing an RL method termed *adaptive switching* for use during real time control of a prosthetic arm. Adaptive switching uses contextual factors to build up predictions about the use of functions during a task. Based on these predictions, adaptive switching will continually optimize and change the order in which functions are presented to the user during switching. We also combine adaptive switching with another 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 GVFs, to switch automatically between functions for the user.

Over the course of several studies, we collected results from subjects with and without amputations, performing simple and more challenging tasks with a myoelectric robot arm. As a first contribution of this thesis, we present

results from work on adaptive switching, and show that it leads to a reduction in the total switching cost (both in terms of time and the total number of switches), an effect that is especially pronounced for experienced myoelectric arm users. For our second and third contributions, 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. We found that although autonomous switching decreases the number of user-initiated switches compared to conventional control, some work remains to be done to improve accuracy of predictions leading to a switch. We also show that the addition of feedback to the user can significantly improve the performance of autonomous switching.

This work promises to help improve the gated control method for prosthetic arms, as well as 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.

Preface

This thesis contains excerpts from previously published journal and conference papers that were written in collaboration with other researchers. The first of these appears in part in Chapter 2 and Chapter 8, and was published as P.M. Pilarski, A.L. Edwards, and K.M. Chan, "Novel Control Strategies for Arm Prostheses: A Partnership Between Man and Machine," The Japanese Journal of Rehabilitation Medicine, vol. 52, no. 2, pp. 9195, 2015. P.M. Pilarski and K.M. Chan supervised this work, and were responsible for paper composition and editing. I was responsible for supporting data, paper composition, and editing.

The second paper appears primarily in Chapter 3 and was published as A.L. Edwards, M.R. Dawson, J.S. Hebert, C. Sherstan, R.S. Sutton, K.M. Chan, and P.M. Pilarski, "Application of Real-time Machine Learning to Myoelectric Prosthesis Control: A Case Series in Adaptive Switching," Prosthetics and Orthotics International. Published online before print September 30, 2015, doi: 10.1177/0309364615605373. M.R. Dawson and C. Sherstan contributed to the development of control hardware and software used in the experiments, as well as paper edits. J.S. Hebert, R.S. Sutton, K.M. Chan, and P.M. Pilarski were involved in supervision and concept design, as well as paper edits and composition. I was the principal author of the paper and led the study. I was also responsible for programming the learning software and for the collection and analysis of data. An earlier version of this paper was also published in conference proceedings as A.L. Edwards, M.R. Dawson, J.S. Hebert, R.S. Sutton, K.M. Chan, P.M. Pilarski, "Adaptive Switching in Practice: Improving Myoelectric Prosthesis Performance through Reinforcement Learning," Proc. of MEC'14: Myoelectric Controls Symposium, Fredericton, New Brunswick,

August 18-22, 2014, pp. 69-73. This article won best paper in the Signal Processing track at the 2014 Myoelectric Controls Symposium.

An adaptation of the third paper appears in Chapter 4 and was published as A.L. Edwards, A. Kearney, M.R. Dawson, R.S. Sutton, and P.M. Pilarski, "Temporal-Difference Learning to Assist Human Decision Making during the Control of an Artificial Limb," 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM), Oct. 25-27, Princeton, New Jersey, USA, 2013. A. Kearney was responsible for software programming, data collection, paper composition, and editing. M.R. Dawson contributed to concept design and paper edits. R.S. Sutton was involved in supervision, concept design, and editing; P.M. Pilarski was involved in supervision, concept design, hardware design, paper composition, and editing. I led the study and was responsible for software programming, data collection and analysis, and paper writing and editing.

The fourth paper is presented in Chapter 5. It was accepted as of May 2, 2016 as A.L. Edwards, J.S. Hebert, and P.M. Pilarski, "Machine Learning and Unlearning to Autonomously Switch Between the Functions of a Myoelectric Arm," 6th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), Singapore, June 26-29, 2016. J.S. Hebert was involved with supervision and editing. P.M. Pilarski was involved with supervision, concept design, paper composition, and editing. I was the principal author and led the study; I was also responsible for software programming, data collection, data analysis, and editing.

This research received ethics approval for two protocols from the human research ethics board at the University of Alberta. The first was for Development of a Myoelectric Training Tool for Above Elbow Amputees (Pro00017008), for renewal August 26, 2016. The second was titled The Development And Clinical Testing Of Advanced Myoelectric Technologies (Pro00030709), for renewal May 6, 2016.

Dedicated to my parents,

whose love and light give me courage to navigate storms.

Acknowledgements

First, I would like to thank my supervisor, Patrick Pilarski, and my co-supervisor, Jacqueline Hebert. Both have shaped a lab environment that enables students' creativity and enthusiasm for research, and because of this, my experience has been invaluable. To Patrick: thank you for your mentorship, your positive approach to teaching, and helping me grow by nudging me to areas outside my familiarity and giving me opportunities to be a part of such cool science. To Jackie: thank you for your kindness, your unwavering encouragement, and your guidance throughout my time as a Master's student. Thank you as well to Mike Bowling, the third member of my committee, for your thoughtful questions and suggestions.

Many thanks to other members of the RLAI and BLINC labs. In particular, I would like to thank Rich Sutton and Ming Chan for their friendly collaboration and valuable discussions. I am also hugely grateful to the students and other colleagues who worked in the lab with me, for their teamwork and their camaraderie.

I wish to express my gratitude to Hani Henein, Sandy Davidge, and Linda Pilarski, who early on gave me the chance to learn about and contribute to numerous research projects, opening my eyes to possibilities in the fields of engineering and medicine.

Support for this project was provided by grants from the Alberta Innovates Centre for Machine Learning (AICML), Alberta Innovates Technology Futures (AITF), the National Science and Engineering Research Council (NSERC), and the Glenrose Rehabilitation Hospital Foundation.

Finally, I am incredibly grateful to my friends and extended family members for supporting me and leading by example. Thank you to my loving par-

ents, Garth Edwards and Barb Every, for your unrelenting support through difficult times. You have given me strength, inspired me, and nurtured me in all aspects of my life; from walks discussing stars and fundamental forces, to shared moments working late into the night. Thank you also to my friend and partner, Alfred Dao, whose patience, caring, and humour have helped push me to achieve my goals.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem

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) are highly versatile (Scheme and Englehart, 2011). The most recent generation of commercial myoelectric arms are capable of upwards of 20 different grip patterns (unique hand positions) and joint motions.

Despite their increasingly diverse set of functions, these arms are often considered by persons with limb amputations to be challenging to operate, largely because of the non-intuitive solutions in place to control so many different available motions (Scheme and Englehart, 2011; Williams, 2011; Resnik et al., 2012). The fact that the controllable functions of a myoelectric arm largely outnumber available muscle control sites has led to a commonly used control interface called switched, or gated control. Using this control method, prosthetic arm functions (which we define as either grip patterns or individual joint motions) are presented to the user as an 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 (as in the traditional direct control approach) (Scheme and Englehart, 2011; Peerdeman et al., 2011;

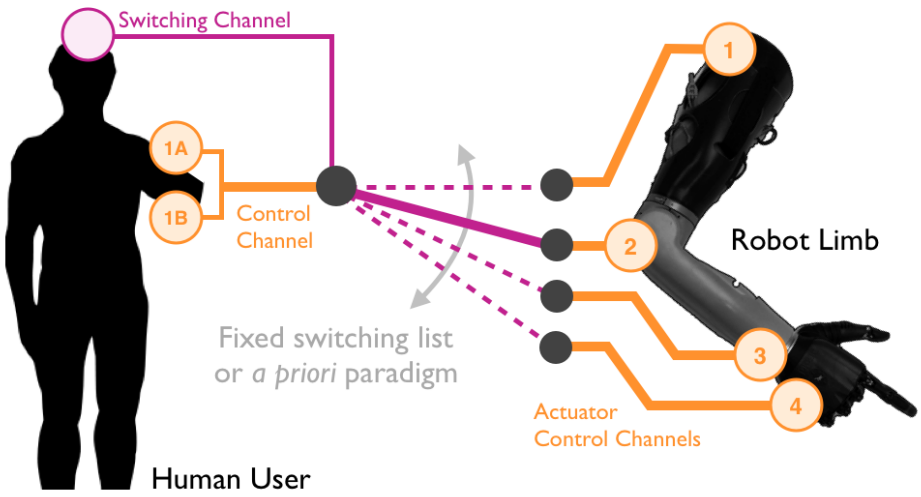


Figure 1.1: *Example of function switching as used to control an assistive device.* One problem for human-machine interaction occurs when a machine's controllable dimensions outnumber the control channels available to its human user. Here, the user has two available channels to control four possible joints of a robot limb. Gated control allows the user to switch their control between joints, one by one.

Micera et al., 2010; Parker et al., 2006). An example of this control scheme is illustrated in Fig. 1.1.

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 (Scheme and Englehart, 2011; Williams, 2011). For example, to perform a simple action like picking up a cup of coffee, a myoelectric arm user would be required to perform a series of sequential actions, such as: switch to the gripper function, open the gripper, position the gripper around the cup, close the gripper, then switch to the elbow function, bend the elbow to bring the cup to the mouth, extend the elbow to bring the cup back to the table, then switch to hand, and open the gripper to release the cup. To make this type of control more efficient, 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.

These limitations have been a driving force for the development of more

advanced control paradigms such as pattern recognition, which increase the number of degrees of freedom that can be intuitively controlled, but are typically still limited to gated control (Scheme and Englehart, 2011; Micera et al., 2010; Castellini et al., 2014). However, as device functionality increases and control becomes more challenging (Resnik et al., 2012), an alternate solution may be for prostheses to begin to assume more autonomy in interpreting and executing a user's intended movements (Castellini et al., 2014).

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 (Pilarski et al., 2013a, 2012). Our demonstrations have all 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 (Modayil et al., 2012b; White, 2015).

Prior work on increasing the number of accessible prosthetic functions demonstrated how predictions about sensorimotor signals pertaining to prosthetic joint movements could be learned and maintained using a technique from reinforcement learning known as General Value Functions (GVFs) (Sutton et al., 2011). GVFs are temporally extended predictions about signals of interest, and have been applied to build up real-time anticipatory knowledge in relation to human-machine interactions (Pilarski et al., 2012, 2013a). It was shown in experiments that GVFs may provide a way to streamline control interfaces with robotic arms (Pilarski et al., 2012, 2013a,b). In particular, it was demonstrated that the use of GVFs to predict which joint of a robotic arm a user will actuate next could lead to an adaptive or situation-specific switching list (Pilarski et al., 2012). A natural extension of that work would be to apply predictions to actual human interaction with prosthetic limbs with the intent of streamlining control. Applying GVF predictions to human-machine interaction is consistent with the idea that the human brain makes forward motor predictions of its own, using knowledge of context and immediate sensory input (Sutton and Barto, 1990; Wolpert et al., 2001; Redish, 2013; Linden, 2003).

1.2 Objectives

In this thesis, we have extended previous work by developing new prediction-based algorithms for the control of assistive devices like artificial limbs. In particular, we explore the use of RL methods, and specifically GVFs, to assist in human decision making during the control of a human-robot interface—a myoelectric arm. We suggest that by acquiring and utilizing knowledge about a user's control-related decisions, control systems and human-machine interfaces could begin to take on a more autonomous role, which could reduce the burden on the user to consciously and independently control every individual joint motion of the prosthesis. Knowledge about a user and their robotic system can take the form of learned predictions about the interactions between the human and their device.

Our main objective in this work is to move toward a more autonomous robot arm system—one that can be applied to improve control and reduce the cognitive burden for amputees—through the use of RL techniques. A secondary goal is to develop algorithms that can be used in a more general way in other switching-based control systems.

1.3 Contributions

This work has several main contributions toward more autonomous methods for control of myoelectric arms: an implementation of adaptive switching in real time, the learning and unlearning of switch predictions, and the development of a new algorithm called autonomous switching. These contributions are described, in brief, below.

1.3.1 Adaptive Switching

A first contribution of this work is adaptive switching. The adaptive switching algorithm optimizes a list of the controllable joints of a robotic arm. Joints are presented to the user in a switching order that is based on predictions about their use. A high prediction of future shoulder motion relative to other joints

results in the shoulder joint being presented first in the switching list; thus, if the user were to initiate a switch at the given moment, their control channel would shift to the shoulder. This re-ordering is a dynamic process that occurs continuously and in real time as long as the arm is in use. The algorithm is built on the hypothesis that a contextually ordered list will result in fewer switches, saving the user task time.

In prior work, our research group demonstrated that offline implementations of an adaptive switching approach could potentially reduce the total task time and the total transition time between joints (Pilarski et al., 2012). In this thesis, we build on this work by, for the first time, bringing adaptive switching into practice and demonstrating its utility during real-time use by multiple able-bodied (non-amputee) and amputee subjects.

1.3.2 Learning and Unlearning Switch Predictions

Another important contribution of this thesis is giving a system the ability to unlearn previously learned predictions. Giving a learning system the capacity to both learn and unlearn a set of predictions is essential for human-centred shared control settings, in which both the human and machine take part in decision making to accomplish a goal. Both humans and autonomous learning systems are imperfect—occasionally, they may make mistakes in the way they form predictions about future events, and they may take actions based on faulty predictions. The system's predictions about signals of interest may be incorrect because the user accidentally reinforced an incorrect action, or because there was an unforeseen change in the environment or goal. Because predictions can be wrong, it is imperative the human retains the ability to correct the autonomous agent: effectively an implicit 'undo' action that results in a decay in the prediction signal. We programmed our learning system, the robot arm, with the ability to learn not just what a user will switch to, but also when a user will switch. In addition, we contribute the ability to unlearn these predictions when tasks or user intent changes.

1.3.3 Autonomous Switching

The autonomous switching algorithm, our third contribution, was developed as an extension to adaptive switching, and designed to be used in combination with the adaptive switching algorithm. Despite the clear advantages to having a continuously optimized list of joints (i.e., from an adaptive list), adaptive switching still necessitates at least a single switch each time the user decides to redirect control to another joint. To further reduce the number of essential interactions between a user and their device, an alternate solution is needed. Autonomous switching provides the means to reduce switching interactions by initiating some switches for the user. Autonomous switching answers the question "when does a user intend to switch to a different arm function?" by building up predictions; then, after reaching a threshold of confidence in these predictions, the robot arm begins to take over the switching action for the user.

1.4 Chapter Summary

In writing this thesis, I have attempted to use an informal tone to ensure the work is understandable to students and researchers outside the computing sciences. As a result, some precision may be lost in the interest of more illustrative examples and terminology. However, all equations and parameters in this thesis are presented precisely and in full.

In the next chapter, I will discuss some essential background topics relevant to this thesis. Chapter 2 includes information on prosthetic arms, current control approaches, and their associated challenges. I will also define reinforcement learning (RL), the specific algorithms used in this work, and discuss previous studies on machine autonomy in human-machine interfaces.

Chapter 3 introduces adaptive switching during human interaction with a prosthetic arm. I will discuss the experiments used to test our adaptive switching algorithm and the results of these experiments.

In Chapter 4, I will discuss experiments with a robotic arm interface in which we make predictions about when an individual will switch functions.

Chapter 5 builds on the predictions from the previous chapter. In this chapter, I will describe the development of the autonomous switching algorithm, including the evolution of the method of unlearning previously learned predictions.

Chapter 6 extends the previous chapter with new results from autonomous switching on a more challenging task. Chapter 7 is a discussion of the work thus far. I examine its implications and limitations, as well as future directions and potential applications. The concluding chapter includes final remarks and a summary of this work.

Chapter 2

Background

There is a projected growing number of individuals with amputations in the United States. Ziegler-Graham et al. (2008) estimate that in 2005, approximately 41,000 people were living with a major (excluding finger) upper-limb amputation (Ziegler-Graham et al., 2008). Not only can the loss of a limb result in varying degrees of physical disability, but it can also cause emotional distress and, in many cases, significant loss of productivity (Ma et al., 2014; Sheehan and Gondo, 2014). The loss of a hand also signifies the loss of an essential tool, both for communication and grasping (Castellini et al., 2014; Zuo and Olson, 2014). The use of novel artificial hand technology can help to mitigate this loss, providing upper-limb amputees with functional alternatives to conventional prosthetic arms.

2.1 Prosthetic Arms

2.1.1 Types of Arms

The primary way to re-establish some functionality post-amputation is through use of an artificial limb (Carlsen et al., 2014). There exist three main classes of prosthetic arms: body-powered, passive, and myoelectric (Biddiss and Chau, 2007). Body-powered prostheses consist of cables, typically harnessed to the intact side. The motion of the shoulder blades causes tension on the cable, and the resultant excursion of the cable through a housing system enables the wearer to open and close the terminal device (usually a gripper or artificial hand). Passive prostheses are neither body-powered nor externally powered.

Although they are primarily used as cosmetic alternatives to powered prostheses, they can also be used to passively grip objects. Myoelectric prostheses have movable joints that are electrically driven by EMG signals and are the focus of this work.

Non-invasive EMG prosthesis control typically consists of electrodes affixed to the surface of a user's skin over muscle sites (Scheme and Englehart, 2011). Electrodes measure the electrical potential of active muscles, with signals proportional to the force exerted by muscle contraction. The recorded signals from each EMG channel are then amplified and processed to produce a set of signals that can be used reliably for control.

2.1.2 Control Challenges

Myoelectric arms have become increasingly powerful devices with a vast number of functions (i.e., preset motions and hand positions) to choose from. The i-limb revolution (Touch Bionics, Livingston, UK) is an example of a state-of-the-art commercial arm and comes with up to 36 pre-programmed grip patterns such as the "handshake" grip and the "index point" grip. Different selectable grip patterns allow the user to perform anthropomorphic grip actions that are functionally similar to some human hand motions. But how does an individual control so many functions at one time?

Gated Control

With only a limited number of myoelectric control sites (e.g., in the worst case scenario, there may be only two or three usable control sites), gated control is a solution which could allow the amputee to switch through an exhaustive list of functions. However, in practical terms, switching through more than three or four possible functions is both challenging and time-consuming. Thus, gated control in practice requires the clinician to reduce the number of functions available to the user to a more manageable list, such that only the functions that are used most often are accessible. While this solution enables the amputee to control their myoelectric arm, it is by no means efficient, as it trades function for control and is considered by many to be non-intuitive

(Scheme and Englehart, 2011).

Pattern Recognition

Other potential solutions exist for controlling a greater number of functions of a myoelectric arm. Pattern recognition techniques have been used to enable more intuitive myoelectric control by extracting key features from multiple EMG sites and classifying these according to desired control actions (Scheme and Englehart, 2011; Sensinger et al., 2009).

However, pattern recognition algorithms are also limited by the fact that they must be trained offline from previously recorded signals. Classification is typically accomplished in-clinic before being deployed for regular use. There are a number of factors that can contribute to a degradation in signal recognition over time: muscle fatigue, skin conductivity from sweating, a change in the position of the electrodes within the arm socket, or differences in contraction strength compared with training samples (Castellini et al., 2014). These changes ultimately cause frustration for the user, as their movement intentions no longer translate into repeatable device motions. Therefore, when using the pattern recognition approach, myoelectric arms must periodically be recalibrated (Scheme and Englehart, 2011). However, recent advances (e.g., the Complete Control System, Coapt LLC) allow the patient to repeatedly retrain their myoelectric device outside of the clinic.

Targeted Muscle Reinnervation

Targeted muscle reinnervation (TMR) surgery is a surgical intervention that increases the number of EMG sites available for control (Kuiken et al., 2004; Carlsen et al., 2014; Zuo and Olson, 2014). Patients who undergo TMR surgery have the residual nerves from the site of amputation rewired and connected to remaining non-functional muscle units in the chest or residual limb. The result is that when the amputee thinks about moving the hand or the forearm, the reinnervated muscles contract, and this signal is mapped to the corresponding motion of the prosthesis. This procedure ultimately provides a greater number of viable muscle control sites, and facilitates a more intuitive mapping to

the functions of a myoelectric arm. Control signals originally intended to move the hand can be redirected to physiologically equivalent functions on a myoelectric arm, and used, for example, to open or close a gripper. Using direct EMG control from these new signal sources also makes it possible for amputees to control multiple degrees of freedom simultaneously. Some sensory reinnervation may also occur after TMR surgery; some amputees experience sensations in their missing limb when touched on the reinnervated area of their chest or residual limb. This has led to the development of surgery, called targeted sensory reinnervation, to remap sensory nerve fibers in addition to motor nerves (Zuo and Olson, 2014). The reinnervated sites can then be used to provide feedback signals (e.g., relating to touch or grip strength) through the overlying skin (Schofield et al., 2014).

2.2 Reinforcement Learning

Despite the advancements in the number and type of interventions related to myoelectric control in recent years, the rate of prosthesis abandonment is relatively high. In the last 25 years, the mean rate of rejection of electric prostheses across 18 studies is cited to be 23% for adult wearers (Biddiss and Chau, 2007). A significant number of individuals also tend to use their active prostheses as passive devices.

The reasons for rejection of myoelectric arms have been studied via questionnaires and can be grouped in three separate categories: lack of intuitive control, lack of function, and lack of feedback (Peerdeman et al., 2011). Reinforcement learning techniques may provide us with solutions for making myoelectric control more intuitive (Pilarski et al., 2013a). In general terms, the idea behind RL is that of a learning system which aims to maximize a signal from the environment, called reward, by adapting its behaviour.

A thorough introduction of RL is provided by Sutton and Barto (1998). Reinforcement learning techniques allow an *agent* (a learner) to learn about a *policy*, π (a way of behaving, or choosing actions A given a state S) by interacting with the *environment* to accomplish a goal. Fig. 2.1 illustrates the

conventional elements of the RL framework and how they relate to one another. A *reward*, R, represents the immediate desirability of the current state, whereas a *value* represents the expected accumulated reward over the long term when starting from the current state. While a small, immediate reward might signify a positive state to the agent, in the long run, the value of that state may be less than the value of another state, making it less desirable overall. A *reward function* maps states (or state-action pairs) from the environment to scalar rewards when following a given policy. Similarly, a value function (v_π) maps states (or state-action pairs) to the expected sum of future reward when following a given policy. In this way, value functions are predictions of future reward. Computational RL is not unlike RL as it is known in biology and psychology, where positive reward is related to pleasure and negative reward to pain.

The discounted sum of future reward (the *return*, G) is expressed in (1). Here, γ represents the discount factor. A discount factor of less than 1 signifies a smaller contribution of reward at each successive time step.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

Following this, v_π can be defined in terms of G. In (2), we show the *expectation*, $\mathbb{E}$, of the return when following a given policy π starting from state s.

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right] \tag{2}$$

For all experiments described in this thesis, we use a type of RL value function, termed a General Value Function (GVF), described in section 2.2.2.

2.2.1 Function Approximation

With only a limited number of states and actions, value functions can easily be computed. However, in practical settings, the state of the world is infinitely large, and not all states will be visited by an agent. As memory requirements grow with larger state and action spaces, there is a need to generalize from

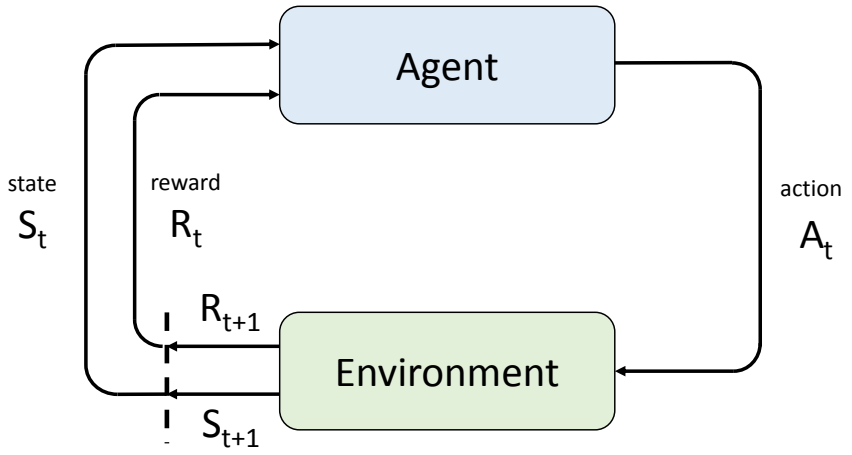


Figure 2.1: *Interactions in the RL framework.* Based on the current state, the agent takes an action, then receives a reward on the next time step. Figure reproduced from Sutton and Barto (1998).

a small subset of experience. Function approximation techniques can be used to generalize to previously unseen states by *approximating* the true function (i.e., a value function or policy) using a select number of known samples.

In this work, we use a linear function approximation technique called tile coding (Sutton and Barto, 1998). Mechanistically, tile coding is a function that takes as an input the state of the world (i.e., observations of interest) and returns a vector of features. The feature vector (denoted $\mathbf{x}(s)$), in combination with a vector of weights (denoted $\mathbf{w}$), which represents the learned expectation of reward, forms the inner product that is used to approximate the value function ($v_\pi(s)$), as in (3). For our work, the feature vector is a binary array, typically sparse, which represents the current state.

$$v_\pi(s) \approx \mathbf{w}^\top \mathbf{x}(s) \tag{3}$$

Tile Coding

Tile coding partitions a continuous state space into a number of discrete units, or tiles, which form the feature vector. For example, when representing the velocity of a car using tile coding, it might be appropriate to divide the range of possible speeds (i.e., from 0 to 200 kph) into 20 tiles, where each tile uniquely

represents a range of 10 kph. Accordingly, if the car is travelling at 45 kph, its current speed falls within the tile which represents the 40-50 kph range. If other types of inputs are also of interest (i.e., distance to destination and amount of gasoline in the tank), these can each be tiled as well, and treated as additional state dimensions. After tile coding, the three inputs suggested here (speed, distance, and amount of gasoline) would form a 3-dimensional array of tiles; however, the dimensions are not limited to three, and additional inputs could increase the number of array dimensions. In a given array, a single tile is considered active, representing the current state. This active tile has a value of 1, and uniquely determines the state at the desired level of resolution; all other (non-active) tiles have a value of 0. An illustration of tile coding with two position inputs (two dimensions) can be seen in Fig. 2.2a.

A representation like the one described above with a single active tile is called a tiling. Multiple tilings of varying granularity can be used, overlapping each other, but with slightly offset ranges to capture a finer or more general representation of the state. An example of tile coding using multiple tilings can be seen in Fig. 2.2b.

In this work, we sometimes make use of a 5-way tile coding function (also used in Pilarski et al. (2012)). This type of tile coding function provides a means of representing the state when there is a large number of features. The difference between classical tile coding function approximation and 5-way tile coding function approximation is illustrated in the following example.

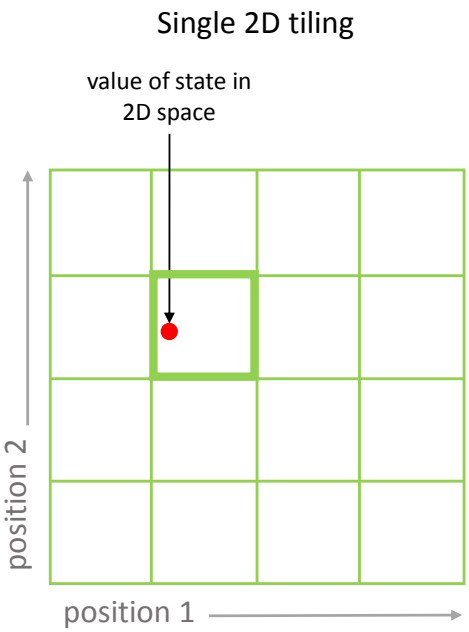
Suppose there are 8 signals that describe a robot arm (four position signals and four velocity signals). A classical tile coding function would result in an 8-dimensional tiling with one active tile. In this case, each of the 8 arm signals would form one axis of the array (Fig. 2.3a). In contrast, 5-way tile coding would result in four separate arrays, each with five dimensions, where each array contains one active tile. A tiling can be formed by stacking these four arrays. Instead of forming one 8-dimensional array, 5-way tile coding forms multiple 5-dimensional (5-way) arrays. In each 5-dimensional array, four of the dimensions are the same (i.e., the four position signals), and the fifth dimension is different between each array (i.e., one of the remaining velocity

signals). In Fig. 2.3b, the dimensions that remain constant are shown in blue, and the dimension that changes is shown in red. This type of representation places greater emphasis on the four fixed dimensions. In this example, and in many of our experiments, we chose to keep the position signals as fixed dimensions in the tile coder because changes in the robot arm's position were particularly important for our specific tasks. An important consequence of 5-way tile coding is a reduction in the size of the array—in some instances, by multiple orders of magnitude.

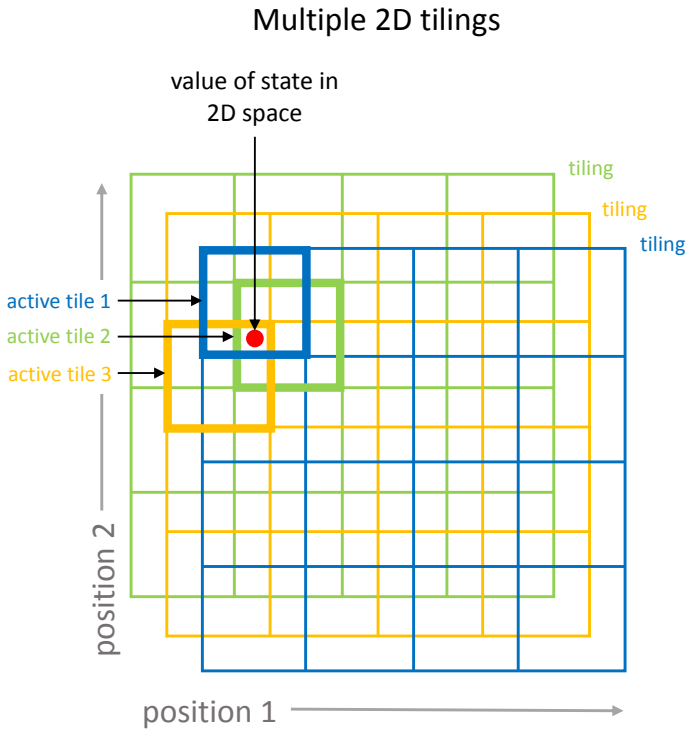
2.2.2 General Value Functions

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. While conventional value functions approximate the return observed by an RL system, GVFs can be used to build up predictions about any arbitrary signal of interest (Sutton et al., 2011). 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?"

GVFs are a form of nexting, an RL approach that enables an agent to learn about and maintain predictions about what will happen next (Modayil et al., 2012b). GVFs enable us to build up predictions about sensorimotor signals at multiple different time scales. Modayil et al. (2012a) have demonstrated that thousands of temporally extended predictions about future values of sensors can be made in parallel by a robot interacting with its environment. Furthermore, the computational requirements of these predictions scale linearly, making nexting a practical tool for learning in real time. Nexting has potentially useful applications in many robotics settings because, just like humans and other animals use sensory cues in their environment to help inform their next actions, nexting gives an autonomous agent a continuous stream of predictive knowledge to help inform their own actions.

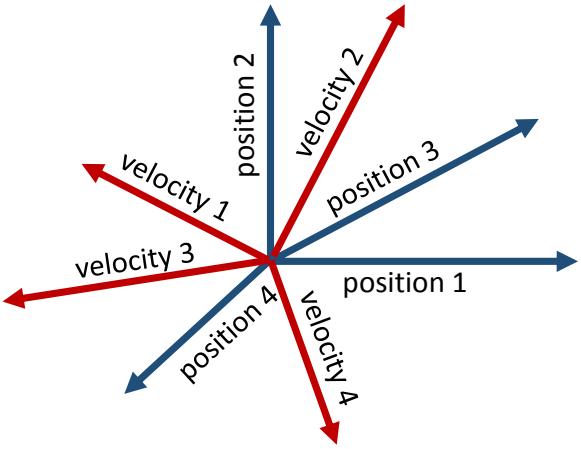


(a)

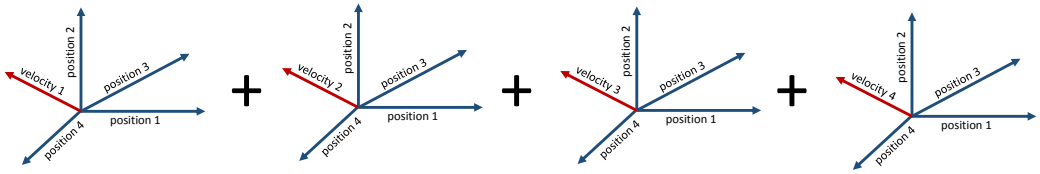


(b)

Figure 2.2: *An example of tile coding in two dimensions.* (a) A tiling with a single active tile. The axes represent two different position signals of a continuous state space; (b) three overlapping tilings, each with a slight offset from the origin. Each tiling contains a single active tile.



(a)



(b)

Figure 2.3: *Representing 8 features using two tile coding functions.* (a) Classical tile coding with 8 features; each feature forms one axis of a multi-dimensional array. The resulting 8-dimensional array has a single active tile. (b) 5-way tile coding with 8 features. The four position features make up four axes of the array, and one of four velocity signals makes up the fifth axis. Each of the resulting 5-dimensional arrays contains a single active tile.

Definition

In GVFs, the signal of interest is called the cumulant, Z, and is used in place of reward, R, in conventional value functions. The cumulant is the signal we are predicting—in other words, it defines the type of question we wish to answer using GVFs. If the question is "when will a myoelectric arm user move the elbow joint?" then the cumulant might be a binary signal that is active only while the elbow joint is in motion; other related parameters can be adjusted to establish the time scale of interest. With experience, a prediction of this type of signal should increase prior to elbow movement. In this way, GVFs can be a powerful tool for predicting a variety of signals. Here, the return, G, can be computed as shown in (4).

$$G_t = \sum_{k=0}^{\infty} \gamma^k Z_{t+k+1} \tag{4}$$

In our experiments, the feature vector, $\mathbf{x}(s)$ is an approximation of the current observed state. The weight vector, $\mathbf{w}$, represents the learned expectation of the cumulant. GVF predictions of the signal of interest are computed from a linear combination of the feature vector and the learned weights, as previously seen in (3). The weight vector can be learned using traditional RL algorithms like TD(λ), or newer algorithms like GTD(λ)—both of these are described in more detail below.

2.3 Learning Algorithms

In this section we describe the temporal-difference (TD) learning algorithms used in the following chapters. TD learning is fundamental to reinforcement learning. Updates to the weight vector are made through experience with the environment, using the difference between the future estimated value of a signal and the last sampled value, plus the reward. As such, TD learning updates the prediction error from an estimate at each state transition, as opposed to waiting for the true value of the signal at some terminal state. Two TD learning algorithms are used in this work. The TD(λ) algorithm was

the update method used in all adaptive switching work, and gradient TD(λ), or GTD(λ), was used in similar fashion in the autonomous switching work. Both algorithms are described in detail below.

2.3.1 TD(λ)

Algorithm

TD(λ) is a TD learning algorithm that is commonly used for prediction estimation. In TD(λ), a temporal difference error signal, denoted δ, is calculated in (5) from 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 γ. Here, $\mathbf{w}$ denotes the weight vector and $\mathbf{x}(s)$ denotes the feature vector, which are used to compute the predictions at S_t and S_{t+1}. The discount factor, γ, affects the weight of future signals about which we are making a prediction. Setting γ to 1 is equivalent to making a prediction about all future experience, whereas setting γ to 0 is equivalent to making a prediction about the next timestep.

$$\delta_t = Z_{t+1} + \gamma \mathbf{w}_t^\top \mathbf{x}(S_{t+1}) - \mathbf{w}_t^\top \mathbf{x}(S_t) \tag{5}$$

According to standard practice, replacing eligibility traces, $\mathbf{e}$, were computed as in (6), using a trace decay parameter λ, to specify the eligibility of features for weight updates. Eligibility traces provide a way of assigning credit to previous states for reward earned in the current state. The trace decay parameter takes on a value between 0 and 1; it defines the way credit is received by those states. Higher values of λ equate to longer lasting traces. For a full discussion of eligibility traces and their utility, please refer to Sutton and Barto (1998). The eligibility traces are computed by adding the previous traces, discounted by both γ and λ, to the feature vector $\mathbf{x}$. We used replacing traces in all our experiments, thereby capping the value of each trace at 1. The weight vector is updated in (7) by adding the previous timestep's weight vector to the product of δ, a scalar learning rate α, and $\mathbf{e}$.

$$\mathbf{e}_t = min(\mathbf{e}_{t-1}\gamma\lambda + \mathbf{x}(S_t), 1) \tag{6}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta_t\mathbf{e}_t \tag{7}$$

Together, these equations (5-7) represent the steps involved in learning GVFs with TD(λ) learning.

2.3.2 GTD(λ)

Off-Policy Learning

A policy maps observed states to actions. Off-policy algorithms use information about a policy that is being followed and acted on (the behaviour policy) to learn about a similar but different policy (the target policy). This is particularly useful in situations when a stream of data includes some, but not all the experience needed for full exploration. In off-policy learning, some samples of experience from the behaviour policy must overlap with samples from the target policy.

TD(λ) has been shown to diverge under certain conditions; specifically, when combined with off-policy estimates and most forms of function approximation (Sutton and Barto, 1998; Baird, 1995). For autonomous switching, we use tile coding function approximation and off-policy learning, and therefore a different algorithm is needed. GTD(λ) is stable for off-policy learning applications.

Algorithm

In GTD(λ), the temporal difference error signal δ is computed as in (5). However, the equations representing the eligibility traces and the weight vector differ from those used in the standard TD(λ) algorithm shown above. To compare the similarity of the behaviour and target policies used in action selection, an importance sampling ratio, ρ, is formed. The importance sampling ratio is computed in (8) from the likelihood of choosing action A_t from state S_t under the target policy, π, divided by the same likelihood under the behaviour

policy, μ. Because these two policies can differ from each other in off-policy learning, ρ acts as a correction factor.

$$\rho_t = \frac{\pi(S_t, A_t)}{\mu(S_t, A_t)} \tag{8}$$

Replacing eligibility traces $\mathbf{e}$ are multiplied by the likelihood ratio in (9). Instead of only one weight vector, $\mathbf{w}$, GTD(λ) uses a second weight vector, $\mathbf{h}$, to correct the gradient of the learning update and prevent divergence during off-policy function approximation (Maei, 2011). The original weight vector is now formed partly using the secondary weight vector, as seen in (10). In the equation for the second weight vector (11), β is a step size parameter for the update of $\mathbf{h}$.

$$\mathbf{e}_t = min(\rho_t(\mathbf{e}_{t-1}\gamma\lambda + \mathbf{x}(S_t)), 1) \tag{9}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha(\delta_t \mathbf{e}_t - \gamma(1-\lambda)\mathbf{x}(S_{t+1})\mathbf{e}_t^\top \mathbf{h}_t) \tag{10}$$

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \beta(\delta_t \mathbf{e}_t - (\mathbf{h}_t^\top \mathbf{x}(S_t))\mathbf{x}(S_t)) \tag{11}$$

2.4 Autonomy

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., 2014).

To highlight a specific example, Cipriani et al. (2008) describe shared control between subjects and an anthropomorphic myoelectric hand. In shared control situations, the human and robotic device each make decisions about or act on some aspect of a task. In the cited work, the prosthetic hand used supervised training and pattern classification to determine the appropriate grasp pattern; then, upon recognition of learned EMG patterns, initiated grasping

actions. The results of this study—that autonomous hand pre-shaping can be successfully used to grasp various objects—suggest that increased device autonomy, in the form of shared control, could present a solution to some of the challenges we face with current switched myoelectric control methods. Another important conclusion from this research was that the more interactions required of a user in terms of accessing the shared control functions of their myoelectric hand led to reduced device acceptability overall (Cipriani et al., 2008).

2.4.1 Trust

In shared control tasks, the human and autonomous device are both active participants in accomplishing an overarching goal. As a result, trust is an essential component of shared control. If trust is lacking, the user may become increasingly frustrated by his or her shared performance of the task. When a robot's autonomy consistently acts inappropriately in a given situation, the operator will begin to over-monitor the robot's actions (Goodrich et al., 2001), and in consequence, the operator's ability to perform multiple tasks is hindered and his or her cognitive load increased. If the autonomy is considered too unreliable, the user may simply stop using the device.

2.5 Experimental Setup

Each of the four main chapters in this work (Chapters 3-6) use a similar experimental setup. Here, we describe the setup used in experiments that involved EMG control, introduce the robotic arm platforms used in this work, and provide a brief overview of the types of tasks given to subjects.

2.5.1 EMG Setup

In all of our experiments involving EMG, the motion of the prosthesis was designed to be velocity-controlled in proportion to the strength of myoelectric signals. We used double-differential Delsys sensors connected to an 8-channel Bagnoli EMG system (Delsys, Inc.). Raw EMG signals were acquired at a

sampling rate of 1 kHz and digitized using a 16-bit DAQ (National Instruments).

Electrodes were affixed to subjects' skin without gel, using a specialized adhesive and touch fastener bands. The placement of electrodes differed slightly between able-bodied subjects (those without amputations) and amputee subjects. All subjects provided written informed consent to participate, and trials were approved by the human research ethics board at the University of Alberta.

Able-Bodied Electrode Placement

In a typical control scheme for able-bodied subjects, one pair of electrodes was placed over the wrist extensor muscle group on one arm. Activation of this control channel (i.e., contracting the muscle) created a signal for switching between the joints of the myoelectric arm, allowing the individual to select a new joint for control. On the other arm, two pairs of electrodes were placed on antagonistic muscle sites (e.g., biceps and triceps or wrist extensors and wrist flexors) for controlling the movement of the prosthesis; one channel was used to move the selected joint in one direction, and the opposite channel was for movement in the opposite direction.

Amputee Electrode Placement

In amputee subjects, one pair of electrodes was placed over the wrist extensor muscle group on the intact arm for switching between joints of the myoelectric arm. The two pairs of electrodes for joint movement were affixed to the residual limb over the biceps and the triceps. One of our amputee subjects had undergone TMR surgery. In this case, the electrodes used in joint movement were placed on the pectoral muscles.

2.5.2 Experimental Robotic Arms

The ExArm, the Myoelectric Training Tool, and the Bento Arm were the robotic arms used in this thesis, and represent successive generations of experimental robot arms.

The ExArm

The ExArm is a wearable, myoelectrically controlled robot arm with four controllable actuators. The shoulder joint could be controlled to move the limb left and right, and the elbow joint could be controlled in an up and down motion. The lower portion of the arm could also flex inward and outward as in wrist joint movement, and the arm terminated in a simple gripping actuator.

The Myoelectric Training Tool

We used a custom-built research platform known as the Myoelectric Training Tool (MTT) in some of our experiments (Dawson et al., 2012). The MTT includes an AX-18 smart robotic arm (Crustcrawler Inc.) that has five degrees of freedom and can be controlled via EMG signals by both amputees and able-bodied subjects. In addition, it can be used as a training tool for amputees preparing to use a myoelectrically controlled prosthetic arm, as it was designed to be functionally similar to commercial prostheses. The MTT operated within a table-top task with a workspace centred on the robot's axis of humeral rotation.

The Bento Arm

The Bento Arm, an anthropomorphic robotic arm, was designed with the MX series of actuators (Robotis Inc.), which are more powerful and robust than the AX-18 actuators (Dawson et al., 2014). Similar to the AX-18 smart robotic arm, the Bento Arm can be controlled by switching between its five degrees of freedom: humeral internal/external rotation, elbow flexion/extension, wrist rotation, wrist flexion/extension, and gripper open/close. The arm can also be mounted on a socket with a harness and controlled by persons with upper-limb amputations like a commercial myoelectric arm. Details of the design of the Bento Arm can be found in Dawson et al. (2014).

2.5.3 Tasks

In many of the experiments described in this thesis, we ask subjects to perform a task called a modified box-and-blocks task. Traditionally, the box-and-blocks task has been used as a clinical test to assess a patient's manual dexterity. In the clinical version of the task, participants are asked to move as many wooden blocks as possible from one side of a divided box to the other in a minute or less. More details of the clinical task are provided in Mathiowetz et al. (1985).

In the modified version, we use five toy balls positioned on one side of the box (Pilarski et al., 2012; Dawson et al., 2012). Balls were chosen to be used instead of blocks because they are easier to grip with the robotic arm. In the modified task, we seek to measure the time needed to move the objects as opposed to the number of objects moved in a given time. The modified box-and-blocks task is an example of a repetitive task that also varies with each pass, as not every path to the goal need be identical.

Chapter 3

Adaptive Switching

3.1 Introduction

The gated control method of switching between control channels is a common solution for controlling multiple prosthetic functions. However, frequent switching between the functions of a prosthetic arm comes at a cost of increased cognitive load, a less intuitive control interface (Carlsen et al., 2014), and longer task time. Previous work using pre-recorded data has suggested that learning predictions about which joint a user will control could decrease the associated costs of switching, reducing the total experiment time and total joint transition time (Pilarski et al., 2012).

In this chapter, we extend prior studies to present evidence that adaptive switching does in fact provide benefit during the real time operation of a robotic arm by a prosthetic user. This work includes a simple demonstration of the use of prediction learning in real time to improve the control of a prosthetic device during use by an amputee subject and an able-bodied subject. We also include preliminary results from three able-bodied subjects performing a second, more complex task. In both cases, predictions are learned and used in real time by the control system to reduce the burden of switching on the user, making it easier and faster to switch to the user's intended next joint or function. The goal of this work is to demonstrate that adaptive switching, as a core application of machine learning, could have a direct effect on reducing the effort of amputee users operating complex multifunctional prosthetic devices.

3.2 Methods

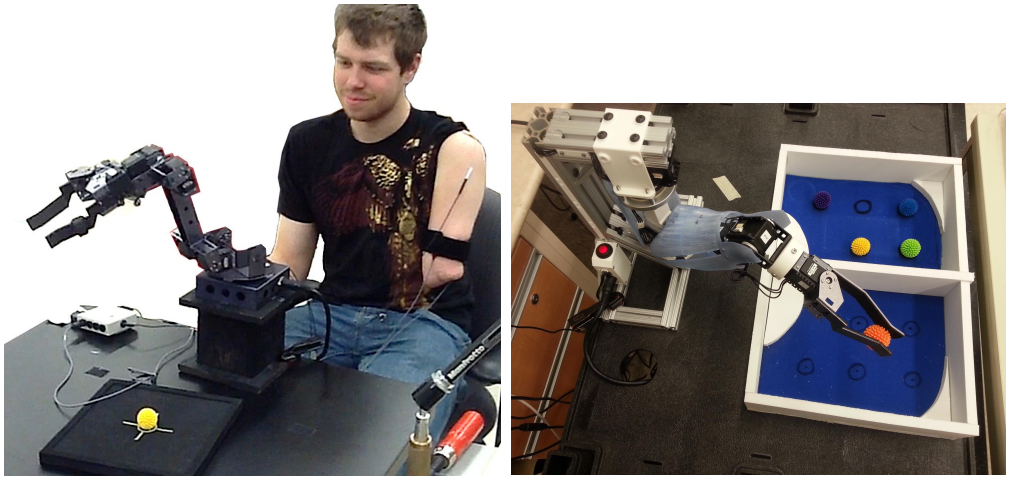
3.2.1 Simple Task

In order to implement and assess adaptive switching, a transhumeral amputee subject and an able-bodied subject were recruited to perform a simple, semi-repetitive task using an experimental robotic arm. The amputee subject was a body-powered prosthetic user and had no experience using myoelectric control or using our experimental robotic arm. The able-bodied subject had some previous experience controlling myoelectric robotic devices. EMG electrodes were attached to both subjects in the manner described in section 2.5.1.

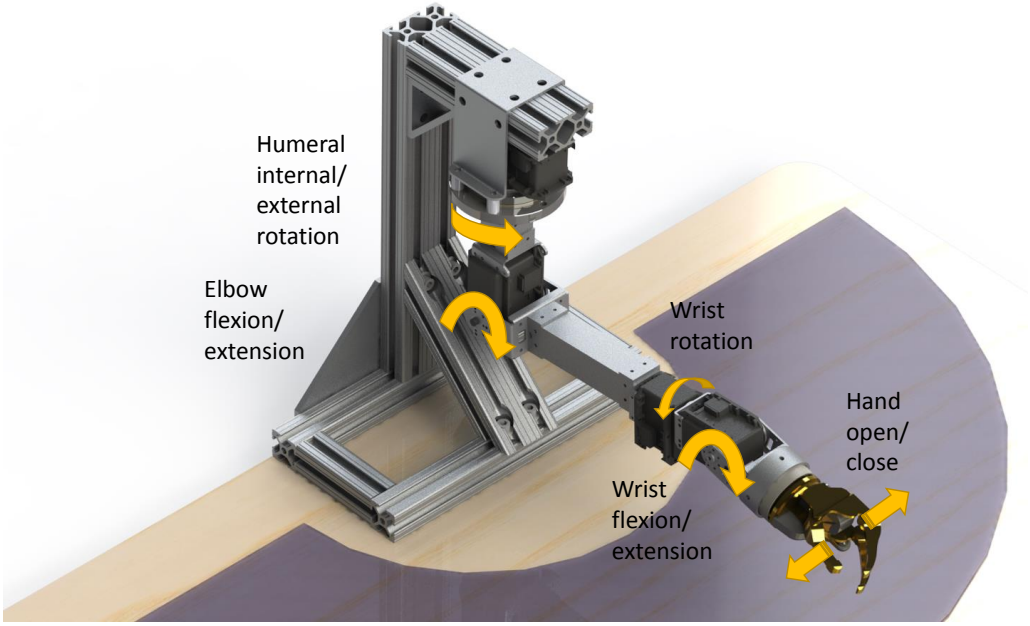
For the simple task, we used the MTT arm as the robotic interface. Fig. 3.1a shows the amputee subject using the MTT to perform a simple task. The MTT was affixed to a table, and operated within a workspace centred on the robot's axis of humeral rotation.

Each subject was given time to become familiar with the operation of the MTT. After familiarization, the subjects were presented with a specific task that involved a subset of the available joints (hand open/close, wrist flexion/extension, elbow flexion and extension, and humeral internal/external rotation). The task was chosen to be functionally comparable to tasks of daily living such as picking up a dish and placing it on a shelf. The instruction given to each subject in both the non-adaptive and adaptive trials was to manipulate the MTT to repetitively open and close the hand (i.e., as if grasping and releasing an imaginary object) on one side of the task workspace, perform humeral rotation to the opposite side of the workspace, repeatedly flex and extend the wrist joint (i.e., as if waving), and then perform humeral rotation back to the starting side of the workspace. Each trial involved repeating this sequence as many times as possible for 3 minutes.

Two types of trials were performed in order to test the predictive capabilities of our proposed control approach: trials using our adaptive switching algorithm (adaptive trials) were compared with conventional non-predictive switching methods (non-adaptive trials). Three 3-minute trials were completed for each condition of non-adaptive and adaptive switching (in alternating fash-



(a) (b)



(c)

Figure 3.1: Experimental robot arms used in adaptive switching. (a) Amputee participant performing simple tasks with the MTT arm using myoelectric control signals; (b) the Bento Arm, controlled by an able-bodied subject performing a modified box-and-blocks task; and (c) a schematic of the Bento arm (without outer casing) depicting the different motions available to the user. The MTT also operates as depicted in (c).

ion), and in both types of trials, an auditory cue was provided to the subject. This auditory cue was a synthesized voice, which named the active joint (i.e., 'shoulder', 'elbow', 'wrist', or 'hand') each time the subject switched.

In non-adaptive trials, the subjects switched their myoelectric control between four joints in a fixed switching order: hand, wrist, elbow, and humeral rotation. This non-adaptive setup was selected to reflect how traditional direct myoelectric control (gated control) is programmed in order to access all four joint motions with limited control sites. In contrast, in adaptive trials, the joints were continually re-ordered in the switching list in real time, based on their predicted likelihood of being used next. This was accomplished throughout the course of the task through the use of GVFs. GVFs allow the control system's programming to continually learn and adapt with each successive control signal received from the user and subsequent joint motion of the robot arm—in essence, the system learns to present the appropriate switch suggestion via ongoing observations of a user's actions.

Algorithms

The algorithm used in gated control, or non-adaptive switching, is described in Fig. 3.2. In the non-adaptive switching algorithm, control and switch signals from the user are observed. If the user initiates a switch to a new joint, the active joint becomes the next joint in the static list of all controllable joints. If the system receives EMG control signals from the user, the current active joint is moved.

```
1. initialize: switch_list, active_joint_index
2. repeat:
3.       observe EMG_C, EMG_S                    // Observe EMG control and switch signals
4.       if switch(EMG_S):                       // If a switch is initiated
5.            increment active_joint_index       // Switch to the next joint in the switching list
6.       move_joint(switch_list[active_joint_index], EMG_C)   // Move active joint according to EMG control signals

switch_list = [S_1, ..., S_N]                    Cyclic list of all controllable joints
active_joint_index = index of switch_list        Points to index of active joint in switching list
switch()                                         Function that returns a boolean based on switch input signal
move_joint()                                     Function that returns a signal that directs arm motion
```

Figure 3.2: The non-adaptive switching algorithm.

The procedure for using GVF predictions during adaptive trials was as

described by Pilarski et al. (2012). Learned GVFs are able to represent predictions about a subject's situation-specific use of each joint in a myoelectric switching list. In the current work, and in contrast to prior demonstrations, GVF predictions were learned during real-time robotic arm use by the subjects and were continually ranked based on their relative magnitudes. The system learned to predict the intended joint for the given task in advance of the switch signal from the user. When a switch signal was received by the system, the highest-ranked joint in the adaptive switching list became the active joint, with the remaining joints filling in the new switching list in decreasing order of prediction strength. Motion control signals were used to drive the current active joint. Fig. 3.3 describes the algorithm used in adaptive switching.

```
1.  initialize: 1 GVF per joint, switch_list, active_joint_index, switching
2.  repeat:
3.      observe EMG_C, EMG_S, s          // Observe EMG control and switch signals, state
4.      get predictions p from GVFs given s   // Compute predictions
5.      if not switching:                // If switching flag set to false
6.          update switch_list based on magnitude p   // Update switching list
7.      if switch(EMG_S):                // If a switch is initiated
8.          switching ← true             // Switching flag set to true
9.          increment active_joint_index // Switch to the next joint in the switching list
10.     move_joint(switch_list[active_joint_index], EMG_C)   // Move active joint according to EMG control signals
11.     if moved:                        // If arm has moved
12.         switching ← false            // Switching flag set to false
13.     for all gvf in GVFs:             // For each joint
14.         gvf.update(s)                // Update GVF and learner using TD learning

switch_list = [S_1, ..., S_N]          Cyclic list of all controllable joints
active_joint_index = index of switch list   Points to index of active joint in switching list
p = vector of predictions of motion of each joint   Adaptive switching predictions
switch()                               Function that returns a boolean based on switch input signal
move_joint()                           Function that returns a signal that directs arm motion
```

Figure 3.3: The adaptive switching algorithm.

To learn real-time predictions of the next active joint, we combined sensorimotor data from the robot with EMG data from the human user. Each of the AX-18 motors that make up the joints of the MTT produced a number of useful sensory signals, including measures of angular position, angular velocity, load (current), temperature, and voltage. We provided joint angular position and angular velocity observations to the learning system as information about the current state. Features based on the current state of the arm enabled the system to build up expectations about future switching decisions made by the user. The machine learning system was re-initialized at the beginning of each trial such that GVFs started each trial with no stored knowledge (predictions) about the user or the task in question.

TD(λ) learning with tile coding function approximation was used to make prediction estimates at a rate of 20 Hz for each of the joints. The values of the learning parameters were as follows: α was set to 0.0125, λ was set to 0.999, and γ was 0.97. Each position and velocity input was tiled into 6 units, and there were a total of 6 tilings, each offset by a small fraction of the tile space. We used a 5-way tile coder, with the position of each joint (shoulder, elbow, wrist, and hand) making up dimensions 1 to 4 of each array, and one of the velocity signals making up the 5th dimension (see Tile Coding, section 2.2.1).

3.2.2 Modified Box-and-Blocks Task

In order to demonstrate transferability of this approach to a functional task commonly used in the clinical setting, we conducted a second experiment with three able-bodied subjects using the Bento Arm. A more complex task was designed to evaluate predictive learning, based on a modified box-and-blocks task. The traditional box-and-blocks task assesses a subject's manual dexterity by counting the number of blocks a subject can move from one side of a divided box to the other in a predetermined amount of time (Mathiowetz et al., 1985). Subsequent studies have shown that simplifying the task to fewer block movements does not result in a loss of valid information on performance (Dawson et al., 2014). For our experiments, we therefore used a modified task that measured the amount of time required for a subject to move five rubber balls from one side of a box to the other (Dawson et al., 2012).

Two of the three able-bodied subjects had no previous experience controlling a myoelectrically driven robotic arm. Myoelectric signals were acquired from each subject through an EMG setup identical to the Bagnoli setup used on the simple task. The Bento Arm (Fig. 3.1b) was controlled using the Robot Operating System (ROS) in a multi-computer configuration, with a central computer handling the direct communication with the arm and recording the EMG. A second computer recorded data over TCP/IP communication using ROS's data logging functionality. Visualization and management of EMG and robotic arm parameters was managed on a third computer.

After a period of familiarization, subjects were asked to control the Bento

Arm, alternating between adaptive and non-adaptive switching trials (for a total of three adaptive and three non-adaptive trials). Each trial consisted of five iterations of moving all five balls from one side of the divider to the other.

3.3 Results

Fig. 3.4 compares the number of switches required per event for non-adaptive switching (top) with the number of switches required during adaptive switching (bottom) for the amputee subject. Each switching event began when the user triggered a joint switch, and ended when the user initiated movement of any of the MTT joints. Therefore, all switches made while shifting control to a new joint were counted as a single switching event. In the adaptive switching mode, the system very quickly adjusted to choose the correct joint for each part of the task with only one exception, as compared to the non-adaptive mode which consistently required additional switches through the trials.

Fig. 3.5 shows that with the simple task performed by both subjects, the average amount of time (measured in seconds) dedicated to switching and the total number of switches required to complete the task were significantly less for the adaptive trials.

Fig. 3.6 compares each iteration of the non-adaptive and adaptive switching trials in the modified box-and-blocks task, averaged over three data sets, where each iteration involved moving all five of the balls from one side of the box to the other. Figs. 3.6a, 3.6c, and 3.6e illustrate the mean total time spent by each subject completing each iteration of the task, and Figs. 3.6b, 3.6d, and 3.6f illustrate the mean total number of switches required for each subject per iteration of the task. The dotted line in each of Figs. 3.6b, 3.6d, and 3.6f represents the minimum number of switches required to complete one iteration of the non-adaptive trial (i.e., the best-case performance that a subject can obtain using the static switching list for this task); the dashed line depicts the optimal number of switches (i.e., the number of switches required if the system predicted each joint with 100% accuracy and switched perfectly with no errors). To perform a single iteration of the box-and-blocks task us-

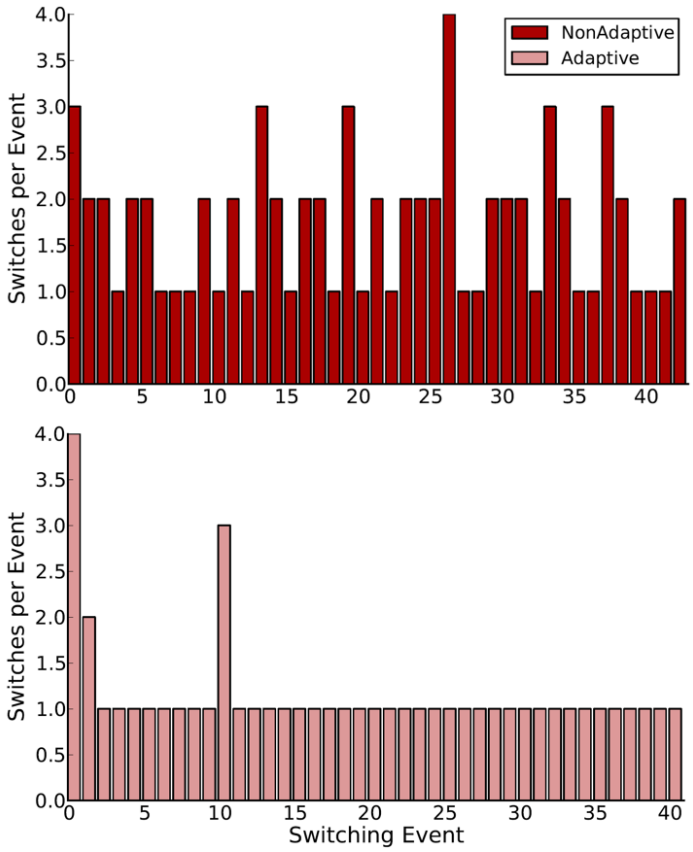


Figure 3.4: Number of voluntary switches initiated by the amputee subject per switching event over the course of a single 3 minute trial. Shown for both non-adaptive (top) and adaptive control (bottom) approaches. The figure is also representative of switching by the non-amputee subject.

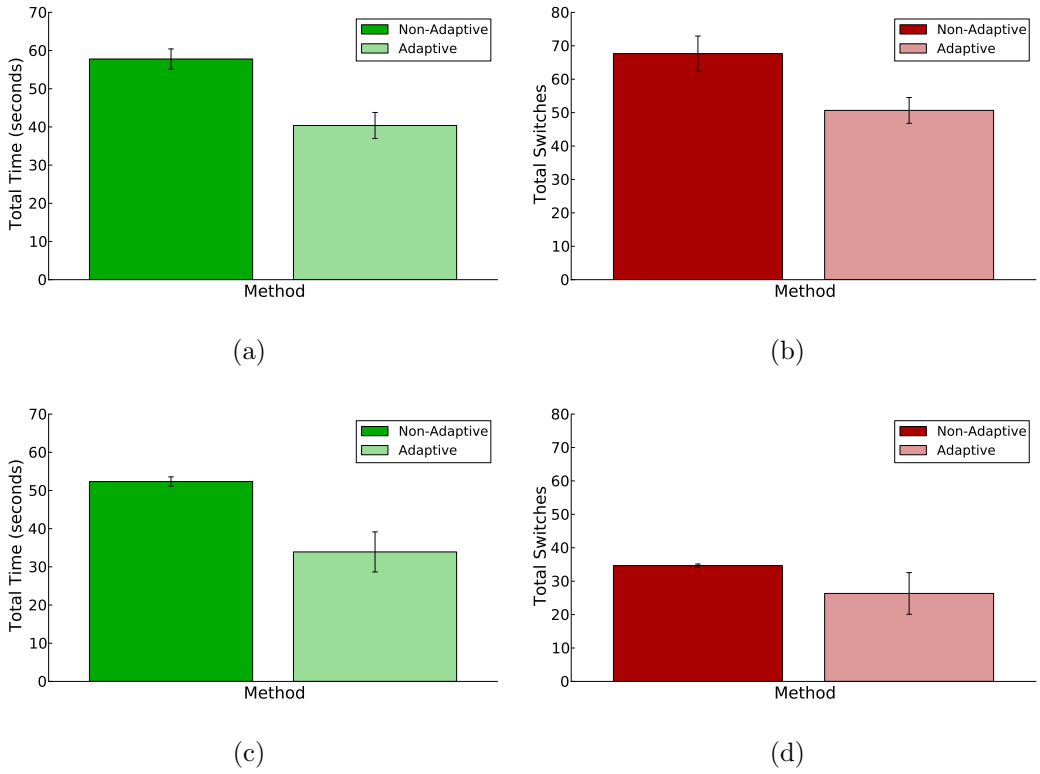
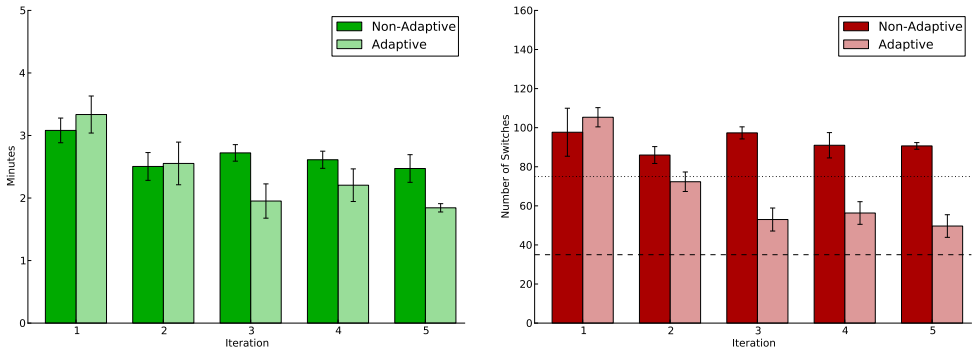
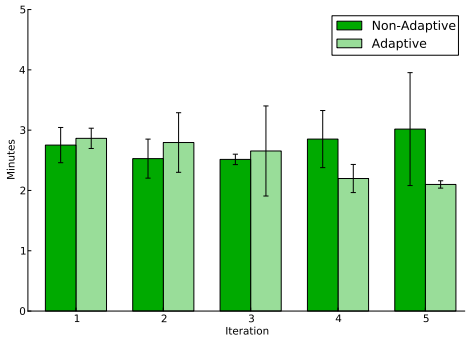


Figure 3.5: Average time and switches per trial for both the amputee and able-bodied subject. (a) Average time the amputee subject spent switching per trial when using non-adaptive and adaptive switching (left and right, respectively, average over 3 trials); (b) average number of switches made by the amputee subject per trial when using non-adaptive and adaptive switching (left and right, respectively, average over 3 trials); (c) average time the able-bodied subject spent switching per trial when using non-adaptive and adaptive switching (left and right, respectively, average over 3 trials); (d) average number of switches made by the able-bodied subject per trial when using non-adaptive and adaptive switching (left and right, respectively, average over 3 trials).

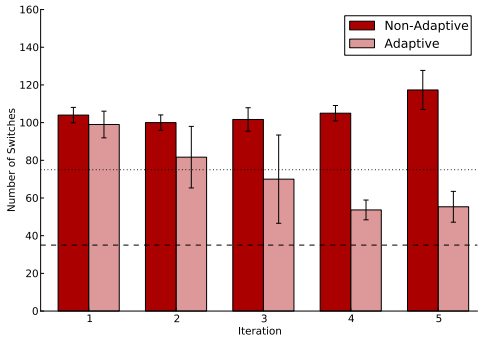


(a)

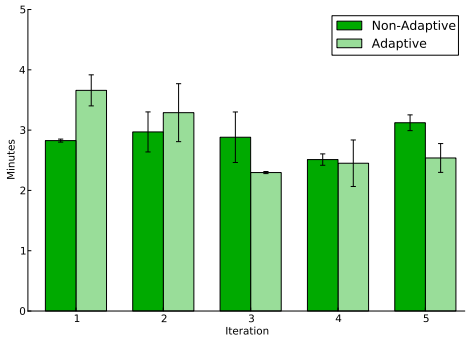
(b)



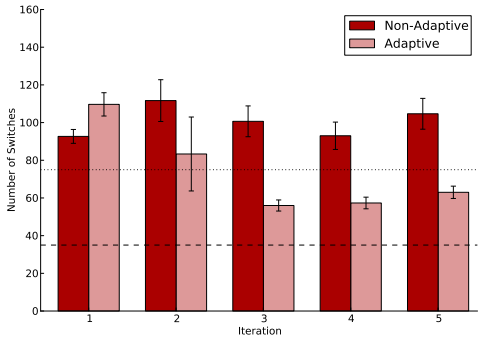
(c)



(d)



(e)

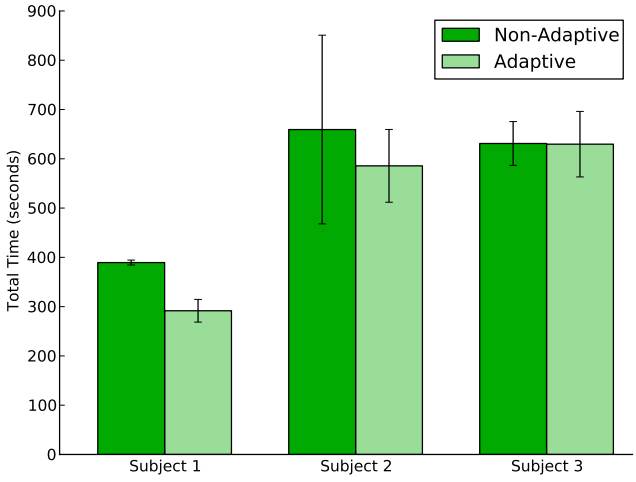


(f)

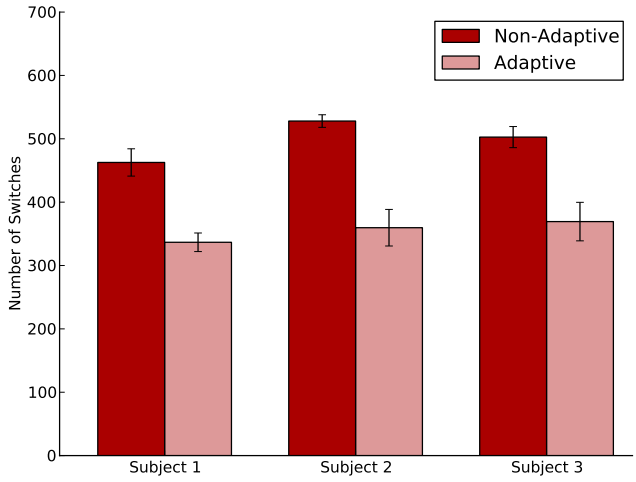
Figure 3.6: Comparison between adaptive and non-adaptive switching averaged over 3 trials. Total time (a) and number of switches (b) of the modified box-and-blocks task are shown by iteration of the task. Dotted line shows the minimum number of switches possible using a static switching list. Dashed line shows the best possible switching performance possible on this task with no user errors and perfect switching (a single switch) at each switching event.

ing the non-adaptive switching list for this robot, a minimum of 75 switches would be required. As illustrated by Figs. 3.6b, 3.6d, and 3.6f, after the first or second iteration (in which the system is still learning), adaptive switching required fewer than 75 switches to complete each iteration. Furthermore, adaptive switching came close to meeting the optimal target of 35 switches. For the modified box-and-blocks task, 35 switches represents the perfect situation where the user made no errors and was given the correct joint every time they switched, with only one manual prompt per switching event. For all subjects, both experienced and inexperienced, in the fifth and final iteration of the modified box-and-blocks task adaptive switching demonstrated significant improvements in terms of both the number of switches required to complete the task iteration and the task completion time.

Fig. 3.7 shows the mean time spent switching and the total number of switches per trial for adaptive and non-adaptive switching during the box-and-blocks task. For the experienced myoelectric user (Subject 1), the mean completion times for the adaptive and non-adaptive trials of the box-and-blocks task were 11.9 $\pm$ 1.2 min and 13.4 $\pm$ 0.8 min, respectively, whereas the mean times spent switching were 4.9 $\pm$ 0.4 min and 6.5 $\pm$ 0.1 min, respectively (Fig. 3.7a). The mean total number of switches for this subject was 337 $\pm$ 15 for the adaptive trials and 463 $\pm$ 22 for the non-adaptive trials (Fig. 3.7b). The non-experienced myoelectric users spent 12.6 $\pm$ 1.1 min and 14.2 $\pm$ 1.2 min completing the adaptive trials; they spent 13.7 $\pm$ 2.1 min and 14.3 $\pm$ 0.6 min completing the non-adaptive trials. They also spent 9.8 $\pm$ 1.2 min and 10.5 $\pm$ 1.1 min switching during adaptive trials and 11.0 $\pm$ 3.2 min and 10.5 $\pm$ 0.7 min switching during non-adaptive trials (Fig. 3.7a). The mean total number of switches for adaptive trials were 359 $\pm$ 29 and 369 $\pm$ 30; for non-adaptive trials, the mean number of switches required were 528 $\pm$ 10 and 503 $\pm$ 17 (Fig. 3.7b).



(a)



(b)

Figure 3.7: Average time spent switching and average number of switches per trial for able-bodied subjects. (a) Average time the able-bodied subjects spent switching per trial when using non-adaptive and adaptive switching (average over 3 trials); (b) average number of switches made by the able-bodied subjects per trial when using non-adaptive and adaptive switching (average over 3 trials).

3.4 Discussion

The goal of this work was to directly compare adaptive to non-adaptive switching to determine the potential utility of adaptive switching in reducing the effort and burden of controlling a robotic arm with more degrees of freedom available than direct control sites. There was a significant difference between non-adaptive switching and adaptive switching during the simple task. With adaptive switching enabled, after a brief initial period of learning by the system (i.e., the first several switching events), typically only one switch was required by the user to move control to the most appropriate joint. Adaptive switching also produced a large decrease in time spent switching compared with non-adaptive switching. For each 3-minute trial with the MTT, each subject saved an average of about 20 seconds when adaptive switching was enabled (11% of the total task time). This could have potential implications on prosthetic users performing more complex tasks requiring multiple joint modal switching by reducing the total amount of time and burden to affect the desired joint control. If joint switching could be made more intuitive (i.e., if the prosthesis reliably selected the correct joint at the right time), it might encourage prosthetic users to utilize additional joint control motions more often, rather than to deliberately restrict control options.

Adaptive switching was also able to reduce time and switching effort in the more complex task. In comparing the decrease in time and number of switches (Fig. 3.6) between the first and fifth task iterations, for all subjects there was only a minor decrease during non-adaptive switching, which is most likely due to improved performance with experience. However, the decrease in time and switches was more rapid and greater for the adaptive switching. By the final iteration of the task, the average number of switches made during adaptive switching decreased to approximately half that of non-adaptive switching. Consequently, by the final iteration, adaptive switching saved the experienced subject more than 30 seconds, or 20% of the total time.

The potential impact that user training may have on the functionality of adaptive switching is suggested by the relationship between the per-iteration

data in Fig. 3.6 and the aggregate data in Fig. 3.7. In those datasets, it is clearly evident that the use of adaptive switching reduced the number of switching interactions needed by the user to well below the level possible with non-adaptive switching. Indeed, by the third iteration of the task, experienced and non-experienced users averaged less switches per iteration than the 75-switch best-case performance of the non-adaptive approach. If the time to perform a switch and resume motion was constant for all subjects, the switching data would suggest significant time savings for both experienced and non-experienced users (as in the simple task, Fig. 3.5). However, for non-experienced myoelectric users, the difference in the total time spent switching between the adaptive and non-adaptive methods was less (Fig. 3.7, Subjects 2 and 3). Non-experienced users were observed to hesitate more following each adaptive switching event and before moving the selected prosthetic joint, primarily in earlier task iterations (as seen in the completion times for earlier iterations as compared to later iterations; Fig. 3.6). One subject indicated that during adaptive switching he paused slightly after each switch to determine the correctness of the choice made by the adaptive switching system. Conversely, the experienced myoelectric user reported greater trust in the choice made by the adaptive switching system, and demonstrated less total time and per-iteration time to complete the task (Fig. 3.6 and 3.7, Subject 1). Based on both these quantitative and qualitative observations, we believe that with more repetitions and greater experience with the adaptive switching, our non-experienced users would demonstrate time savings similar to those of the experienced user. More detailed studies are needed to determine the roles that trust, experience, and training play in the successful use of adaptive switching and related control adaption paradigms.

The results from both the simple and more complex tasks suggest there are efficiencies with adaptive switching, and they agree with our expectations inferred from the simple task presented to the subject: there were clear regions of the task space that corresponded to the use of specific joints. For the simple task, it would have been possible to hand-code several different switching lists in response to the different positions of the shoulder actuator. The simplicity

of the task design allowed us to easily verify the correctness of the adaptive switching options proposed by the learning system. However, a key observation from the present work is that situation-specific switching orders do not need to be hand-coded; our system learns situational delineations as the robotic arm is being used, and without prior information about the user or the task (thus implementing a form of adaptation, or ongoing self-calibration, which has been pointed out to be of great clinical interest as it removes the need for regular recalibration by clinical staff (Scheme and Englehart, 2011; Sensinger et al., 2009). Furthermore, we have observed that as the task changes or becomes more complex, as in the case of the modified box-and-blocks task, the learning system can scale up naturally and easily without the need for manual tuning.

It is important to note that while our results closely align with expectations from prior work (Pilarski et al., 2012), the present experiments are case studies, which limits the generalizability of the conclusions. In addition, transferability to other tasks will need to be explored in order to assess relevance of this approach to the multitude of tasks required of prosthetic users in daily life. One strength of our proposed approach is that it is able to optimize prosthetic control for a repeated pattern of prosthetic movements, without that pattern being explicitly specified by a user or clinician. The adaptive switching system begins with simple deterministic switching, as would be familiar to myoelectric prosthesis users, and gradually optimizes control as regularities are observed in the movements made by the user. Because of the way that information is presented to the adaptive switching system (function approximation, c.f., Pilarski et al., 2012), the system is able to leverage similarities and generalities in the movements being performed—i.e., it is able to form generalizations that are applicable to multiple tasks, such as "after reaching forward the user usually grabs an object". While these generalities allow some translation of learned patterns to novel tasks, it is important to note that the system will still need experience with a new class of tasks before being able to form a reasonable set of adaptive switching suggestions. For daily-life usability, we suggest that adaptive switching may also need to be engaged and disengaged based on the control systems confidence in its own predictions; in other words, it may be

more intuitive for users if prosthesis control defaults to simple deterministic switching until the system has enough experience to appropriately adapt the switching list in a given setting. How best to engage and disengage adaptive switching in this way, and what function approximation methods will allow the best generalization between real-world tasks remain open questions. Future work will also need to identify the most effective way to communicate the system's switching suggestions back to the user—auditory (as used here), visual, tactile, and vibratory signals are all valid possibilities for feedback.

Ideally these experiments will be repeated in multiple subjects with a range of experience in myoelectric control, and on subjects using donned prostheses during daily-life tasks. However, the current case study is a necessary and encouraging first demonstration of the practical applicability of our approach with both able-bodied and amputee subjects.

3.5 Conclusion

The primary contribution of this chapter is a concrete demonstration of adaptive switching in an applied setting of robotic arm control, in both able-bodied subjects and an amputee subject. This study is the first time that real-time prediction learning has been used to improve the control interface of a prosthetic device during uninterrupted use. Our experiments with an amputee subject showed that for simple tasks, enabling adaptive switching on a robotic arm significantly decreased the time spent switching between functions. This extends previous work using pre-recorded (non-real-time) data that indicated the potential merit of adaptive switching. These results have been extended to a box-and-blocks task that is more representative of complex everyday tasks.

We believe that adaptive switching can help to decrease the time and cognitive load required by amputees during complex tasks and real-world functional situations involving wearable prostheses. An extension of this work can be seen in Chapter 5, wherein we study the use of adaptive switching in shared-control functional tasks with prosthetic users. Switching control is therefore further delegated to a control system to reduce the cognitive burden on the user.

Chapter 4

Predicting Switch Signals

4.1 Introduction

In the previous chapter, we demonstrated that it is possible to predict *which* joint a user intends to use. A natural extension is to ask "*when* will a user switch to a new joint?" In this chapter, we further explore the unification of RL with conventional switching-based control interfaces. In particular, we demonstrate the use of online TD learning with GVFs to predict when a user will switch their control influence between the different control functions of an articulated robot arm. We expect that when a system is certain enough about its predictions regarding a user's switching target and switching timing, it can begin to take over some function switching decisions from the user to streamline control and potentially decrease the time and effort needed to complete tasks. Our over-arching goal is to continue to develop predictive approaches that ultimately enable the more natural control of complex assistive devices by reducing the number of manual interactions required by a user.

4.2 Methods

The ExArm was used as the experimental platform for this work (Figure 4.1). Electrodes were affixed to the skin of non-amputee subjects and used to measure EMG signals. These EMG signals were mapped to two control channels: one to actuate a robotic joint, and one to switch between the different joints in a fixed, sequential fashion.

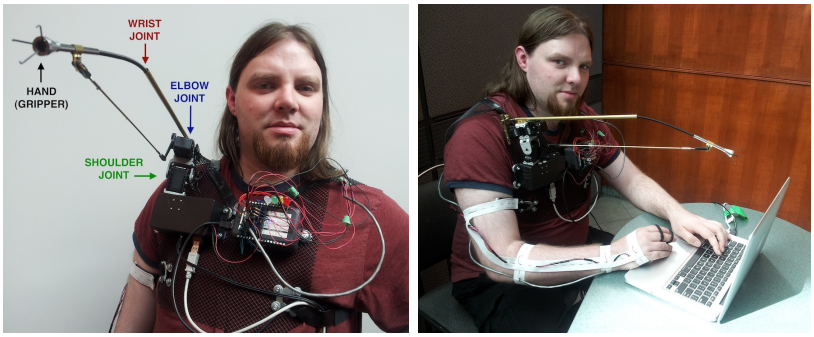


Figure 4.1: *The experimental platform used in this work*: a wearable robot limb that is controlled using muscle signals from the human body, where the user sequentially controls and switches between the available joints using voluntary muscle contractions (similar to the control interface for a commercial forearm prosthesis).

In the previous chapter, we developed GVF predictions about *joint motion* from human interaction. In this chapter, we examine the ability of GVF-based TD learning to predict *joint switching* from human interaction with the robot system during simple movement tasks. An able-bodied (non-amputee) subject actuated the myoelectric arm using the EMG setup described in section 2.5.1. Using this wearable system, each subject performed a semi-repetitive motion, moving the robot's shoulder to the right, moving the elbow up and down an arbitrary number of times, moving the shoulder joint back to the left, and then moving the wrist up and down an arbitrary number of times. This H-shaped movement pattern was repeated for 10–30 minutes. As shown in Fig. 4.2, this resulted in a rich stream of data for use by the RL system, and provided a challenging setting for learning due to the temporal variability and non-stationary nature of the user's myoelectric control signals and switching behaviour.

The knowledge learned by our system regarding a user's switching actions took the form of temporally extended predictions about a user's switching prompts; these predictions were similar to the predictions made in previous work on anticipating the activity of user-controlled actuators (Pilarski et al., 2012; Pilarski and Sutton, 2012). In this work, predictions were acquired and updated through multiple iterations using an implementation of GVFs (Sutton et al., 2011) and nexting (Modayil et al., 2012b). Following the approach

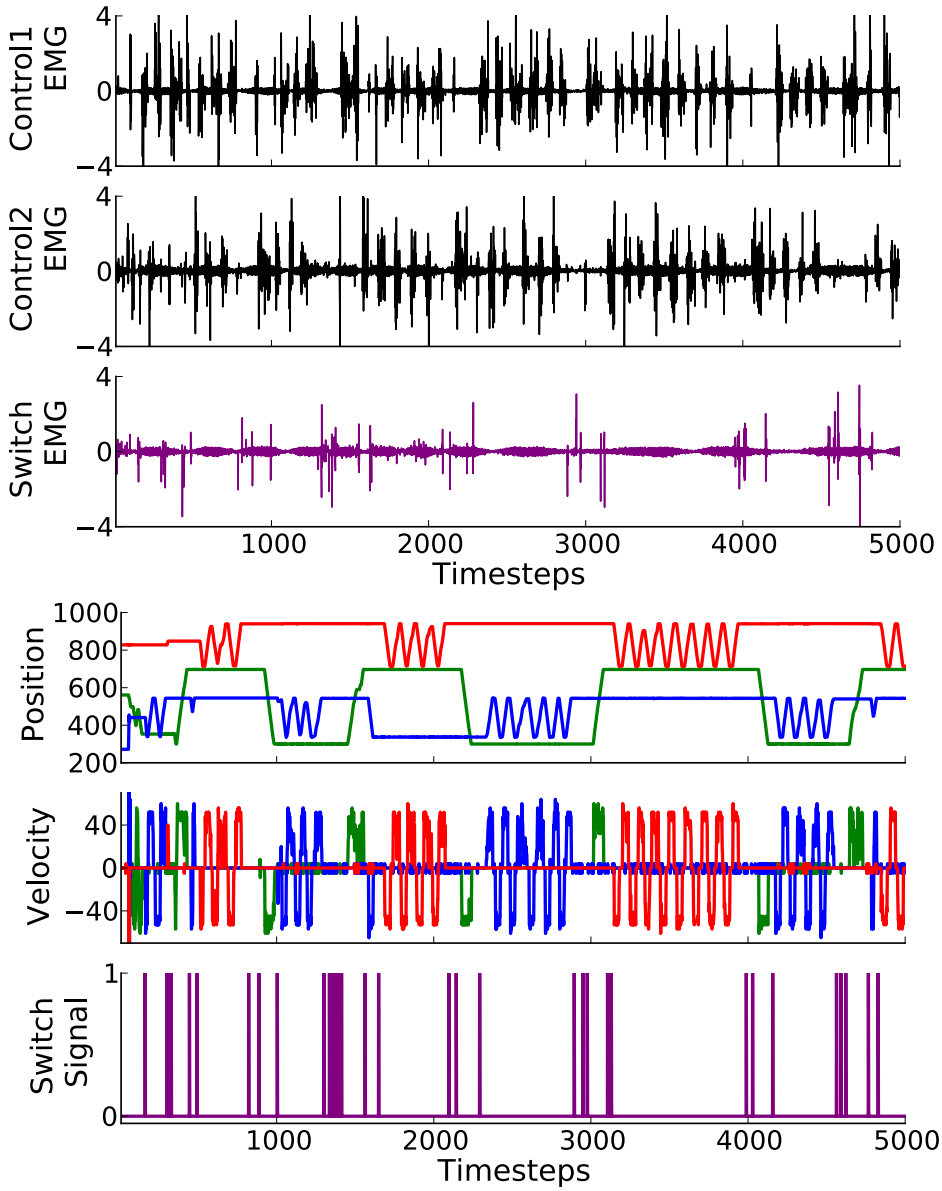


Figure 4.2: *Example of the sensorimotor data stream from the human-machine interface*, including recorded muscle activity (EMG) for control and switching channels (black and purple traces, top), human switching actions (purple trace, bottom), and the angle and speed for three of the robot's joints (red, blue, and green traces, bottom).

of Pilarski et al. (2012), GVFs were updated on each time step using TD learning with eligibility traces and 5-way tile-coding function approximation. Each GVF learner was initialized with parameters specifying the prediction of interest, including the timescale of the temporally extended prediction and the target signal of interest—here a binary signal that was active when the user prompted the system to switch motor functions. In this experiment, α was set to 0.1 (divided by the number of active features—in this case, 56), γ was set to 0.9, and λ was set to 0.99. Signal sampling and learning updates occurred at 15 Hz. The state representation used by the machine learner was comprised of motor feedback (e.g., position, speed) from the robot arm and signals relating to the human's recorded muscle activity (e.g., processed EMG signals and switching cues), as well historical information in the form of decayed traces of these signals. Each feature was divided into 8 tiles (with 8 tilings). As in previous work, our system also learned a series of temporally extended predictions regarding the motion of each user-driven joint and the user's myoelectric signals. These temporally extended predictions were initialized with the same parameters and state representation as the predictions about switch signals.

4.3 Results and Discussion

As shown in Figure 4.3, top, following a period of learning our approach demonstrated the ability to forecast an upcoming switching cue from the human user. Advance knowledge of switching (a rise in the dark purple trace) was observed to arrive a fraction of a second before the actual human-initiated event (grey pulse). Predictions on both training and testing data (dark purple traces) were also observed to begin to approach the true, computed return (light purple trace) as learning progressed. Additional learning is expected to improve the agreement between the true and learned predictions, and testing is ongoing to determine the best state representation for this learning scenario. As expected from previous work, our approach was also able to consistently anticipate which joint a human user would actuate next while performing their task (Figure 4.3, bottom). Testing and training data were sampled from the

same human user, with the training and testing sessions being conducted on different days. The testing data were not seen by the learning system prior to evaluation.

By ranking the magnitude of joint activity predictions prior to manual switching by the user, a learning system is able to determine the most appropriate joint to select at the time of switching (Pilarski et al., 2012). In other words, simple relationships between the predictions can be used to formulate the system's switching suggestions, i.e., which joint to actuate next. These suggestions depend on both context and learned knowledge about a user's preferences. The present work contributes a way to determine the desired timing of switching actions. Taken together, these straightforward applications of learned predictive knowledge provide a way to allow a learning-based control system to gradually assume more autonomy and decision-making responsibility during ongoing human-robot interactions. One useful feature of this approach is that no explicit or time-consuming reinforcement is needed from the human user to correct or affirm the learning system's suggested decisions; the use of a mode or function by the user verifies the system's choices, while use of an alternate function decreases the learning system's predictions about the suitability of a given control option (Pilarski and Sutton, 2012). Our approach therefore differs from predominant approaches to human-directed RL like human reward (Thomaz and Breazeal, 2008) and demonstration learning (Lin, 1992).

The ability shown in the present work to anticipate a switching event promises to greatly reduce the need to manually initiate switching. The time needed for switching could potentially even be eliminated in certain situations. As suggested in Pilarski and Sutton (2012), removing the need to explicitly switch functions during commonly performed tasks could result in almost as great a time savings as selecting the optimal switching target or function.

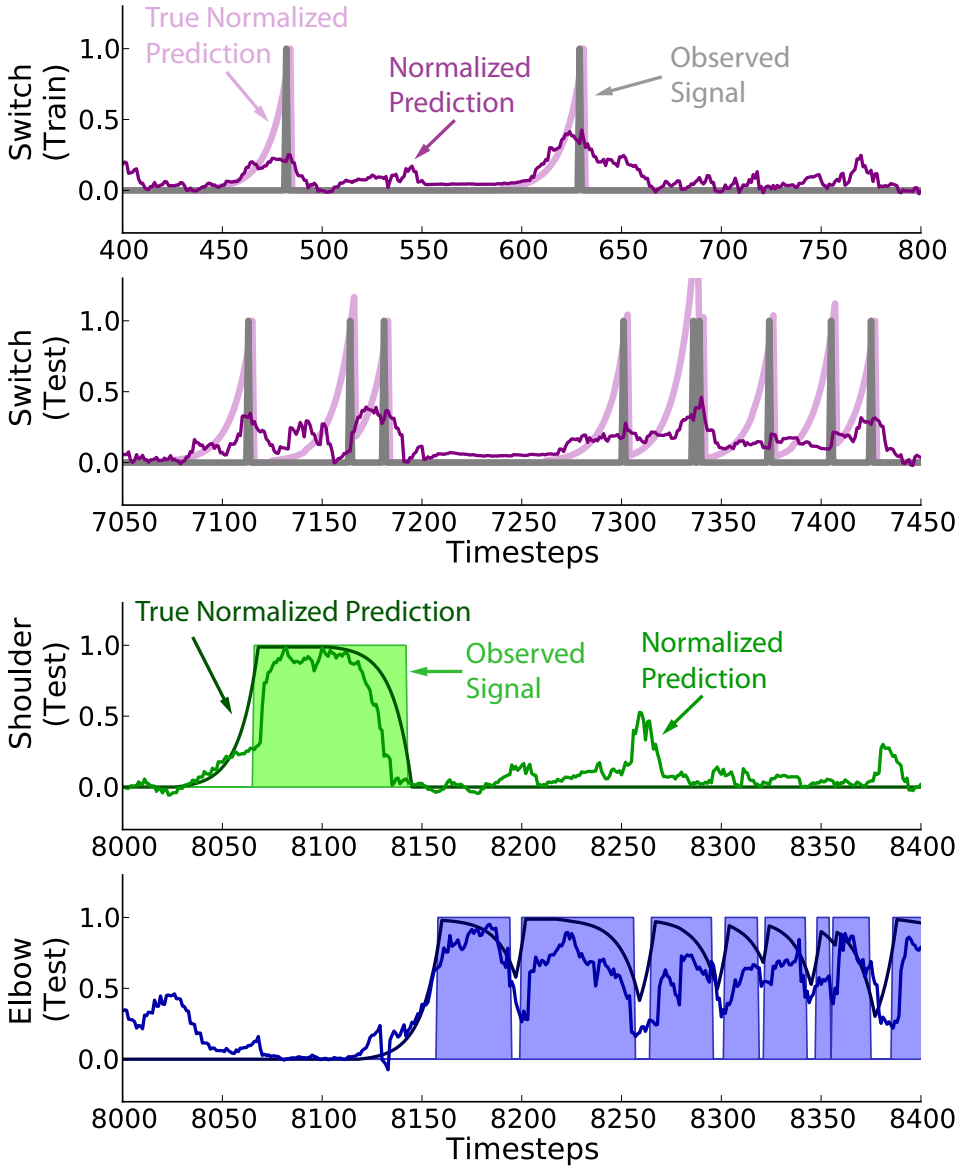


Figure 4.3: *Example of switching event and joint activity predictions on previously unseen testing data* after five iterations of TD learning through 28k steps of training data. *Top*: switching event predictions begin to rise in advance of actual switching events initiated by the user, as shown on both training and testing data. *Bottom*: predictions about joint activity rise in advance of expected joint actuation. Modulating a control interface based on these predictions promises to reduce the time needed for a user to complete a task with a multi-joint robot arm (Pilarski et al., 2012; Pilarski and Sutton, 2012).

4.4 Conclusions

In this work we demonstrated the use of RL to help with human decision making, and specifically provided another step towards intuitive human interaction with a switching-based robot arm. Function switching is a common way to deal with increasing device complexity, but it poses additional challenges to the natural and efficient control of devices by a user. To help address the barriers to streamlined human-robot interactions, we deployed state-of-the-art RL techniques to acquire and maintain knowledge about a user and their robotic system. Our approach was able to build up and maintain forecasts about a user's switching behaviour in real time. We also confirmed previous observations that our approach can detect a user's control intent prior to their explicit control actions. Bringing together these two ideas, a system could potentially determine what function a user intends to deploy, and when they wish to begin using the new function.

Our approach allows a user to remain in direct control of a system while still allowing the device to suggest or initiate increasingly appropriate control options. Furthermore, the opportunity for ongoing yet optional human interaction in the decision-making process provides an intuitive and reward-free way for users to correct or reinforce the decisions made by a semi-autonomous machine learning system—there is no need for the user to explicitly reward the learning system. This preliminary study therefore opens the way for naturally blending the control decisions made a human and their assistive robot or other human-machine interface. Future work will continue to pursue the integration of biological and synthetic reinforcement learning and decision-making systems.

Chapter 5

Autonomous Switching

5.1 Introduction

Gated control allows users to access multiple prosthetic functions by switching through a list of functions and selecting the one they wish to control—not unlike switching through and selecting channels on television. However, to reduce the complexity of control, the user is typically given access to a few of the most common functions. We propose instead to 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 (Castellini et al., 2014).

Through the use of nexting, and specifically GVFs (White, 2015; Sutton et al., 2011), we previously developed a novel switching control method for myoelectric arms called *adaptive switching* (see Chapter 3); this approach was validated in pilot studies with both amputee and non-amputee participants (Edwards et al., 2015; Pilarski et al., 2012). Adaptive switching re-orders a list of arm joints for the user on the basis of their predicted likelihood of being used in a given situation. 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). When using adap-

tive 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.

5.1.1 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 with 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 chapter, we introduce a novel control method, termed *autonomous switching*, that works in parallel with adaptive switching. 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 to a user's control intentions and behaviour. Autonomous switching builds on previous work described in Chapter 4 (Edwards et al., 2013), in which we demonstrated that it is possible to build up predictions about *when* an individual is likely to switch their control between functions. 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 there be a change in the parameters of a task or the user's intent. 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 chapter, 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 once a task is changed. In the second experiment, we compare autonomous switching with previously tested control methods, and demonstrate the effect of providing feedback on the user's performance.

5.2 Methods

5.2.1 Experimental Setup

The robotic arm used in the experiments described in this chapter was the Bento Arm (Fig. 5.1). Control by the user was set up 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) to switch the one channel of control between the multiple functions of a device (Parker et al., 2006). 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. Able-bodied subjects were recruited and gave informed consent in accordance with the study's Human Research Ethics Board approval.

5.2.2 Learning Algorithms and General Value Functions

As noted above, our control approach uses nexting in the form of GVFs to build up predictions about sensorimotor signals at various timescales (Sutton et al., 2011). In our experiments with adaptive and autonomous switching, we framed two distinct questions through the use of GVFs:

1. Which joint, grasp, or function of a robotic arm will a user want to move next? (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? (Autonomous switching.)

5.2.3 Adaptive Switching

Adaptive switching was implemented as described in previous work (Chapter 3; Edwards et al. 2015). We used the standard temporal difference learning algorithm, TD(λ) (Sutton and Barto, 1998), to allow a system to computationally answer the first of these questions, and in doing so, use its learned predictions to re-order the options presented to the user during switching.

Learning and control occurred at 20 Hz. 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. For the adaptive switching state representation, we provided input signals relating to the angular movement of each joint (the position and velocity of each servo motor) and torque information from the gripper joint only, which is measured in terms of current load. This representation provided clear predictions of joint motion after only a brief period of learning, and thus a basis for a reasonable, adaptive joint order during ongoing use.

We used a tile coding function approximation method (Sutton and Barto, 1998) (Chapter 2) to create a mapping of the signal space. Since the representation consisted of 11 observations (five position signals, five velocity signals, and the load on the gripper), we began with an 11-dimensional signal space. The range of each observed signal was normalized and divided into six discrete units, or tiles, to form a large binary array, with a single unit representing the current active state. We generalized learned weights over a broad area of the space by overlapping four identical tilings, each with an offset from the origin, and each containing a single active tile. Although the total number of features was over 1.4 billion, a hashing technique was used to reduce the size of the

array to slightly over 1 million features.

For adaptive switching trials, γ was set to 0.9. Replacing eligibility traces (discussed in Sutton and Barto, 1998) and the weight vector were initialized to 0. The values of λ and α were fixed at 0.99 and 0.0025, respectively.

We followed the equations outlined in Chapter 2 for learning GVFs. This sequence of equations was repeated at each timestep in adaptive switching experiments. A separate prediction agent was initialized for each joint of the robot arm. As per the procedure from Chapter 3, at each timestep, the resulting joint predictions were ordered in terms of relative magnitude, and the joint list was updated and presented to the user in a descending order of magnitude. For example, if the prediction of shoulder joint motion for the current state was highest of all the joint predictions and if the user were to switch, the shoulder would become the next active joint. However, in the adaptive switching control scheme, from the moment a switch sequence began until a joint was chosen and moved, the joint order was frozen. The joint that was last moved always occupied the last position in the list (Edwards et al., 2015).

5.2.4 Autonomous Switching

The algorithm for autonomous switching was also based on TD 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 preliminary experiments, we found that autonomous switching with on-policy TD learning about the switching signal itself created the unstable condition in which switches made by the system are self-reinforcing, causing predictions about switching to rise indefinitely. In such a scenario, errors made by the user in switching or any changes in the task or task environment could not be corrected. For autonomy to be accepted and effective, a system should be able to modify previously learned switching behaviour. For these reasons, autonomous switching required GVFs and a related learning system with the capacity to both learn and unlearn autonomous switching behaviour throughout the course of a task.

In autonomous switching, GVFs predict what the outcome would be *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 currently be pursuing, autonomous switching requires the use of an off-policy algorithm, GTD(λ) (Maei, 2011). In these experiments, the behaviour policy was defined by how the user was controlling the robot arm, and the target policy was 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 remained 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 used 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 or changes to the environment or to the user's intent. As in adaptive switching, a user's use of the system's automatic switching behaviour provides *implicit reinforcement* for the system's choice to switch, by way of strengthened predictions, while using the same joint will decrease the system's probability of switching by decreasing the predictions for a given situation.

To form a feature vector for the learning agent we used tile coding function approximation as above, but now using the position and velocity of all five joints, the torque on the gripper, and also a moving average of the EMG channels responsible for joint motion as signals (13 inputs). The total number of features in the array was over 52 billion and was hashed down to just over 1 million (with four active features).

In autonomous switching experiments, the behaviour policy matches the target policy from the start of a switching sequence until the moment the user selects and begins moving a new joint. In all autonomous switching experiments, α was set to 0.025 and β was set to 0.000025.

Throughout the task, we used $\gamma(S)$ as described above with λ set to 1. This had an important effect on the shape of the predictions leading up to a change in the active joint. Because γ had a value of 1 throughout switching sequences, this resulted in a "square" shape to the predictions, which maintained the same height regardless of the length of the current switching sequence. In this way, the beginning of a switching sequence determines whether the predictions exceed the predetermined threshold for autonomous switching to occur.

The threshold for autonomous switching was empirically determined and fixed for all experiments at 0.3. There was a separate prediction learner for each of the five starting joints of a switching sequence, with the prediction of the last-moved joint's learner being the one evaluated against the thresh-

old. Although other techniques could have been used to determine when an autonomous switch should occur, a threshold has proven to be a simple but effective way to initiate a switch, and approximately corresponds to a level of certainty in the prediction. 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.

The algorithm for autonomous switching is summarized in Fig. 5.2. Because autonomous switching is acting in combination with adaptive switching, the algorithm is similar to the adaptive switching algorithm presented in Chapter 3.

5.2.5 Tasks

Experiment 1

The task chosen for Experiment 1 to assess unlearning was a task similar to the simple task used in previous studies (Edwards et al., 2015, 2013). 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 re-ordered, 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.

Experiment 2

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

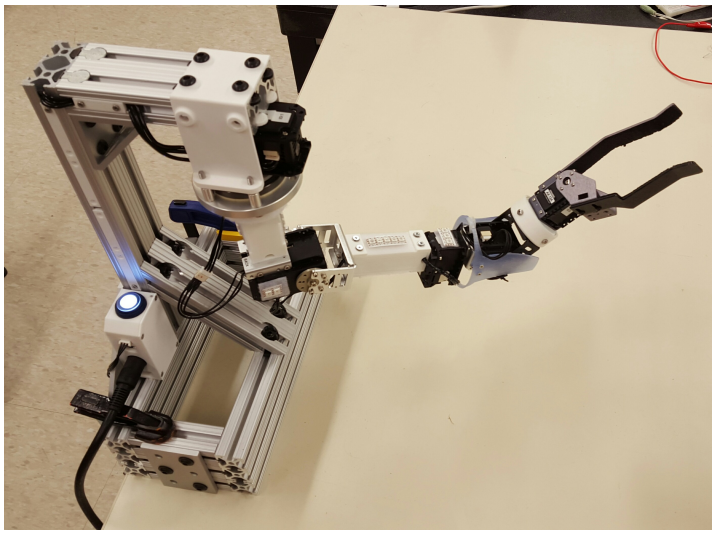


Figure 5.1: The Bento Arm, a custom-designed robot arm, was controlled by able-bodied subjects in two separate experiments. The Bento Arm has five joints that can be controlled via EMG signals.

```
1.  initialize: 2 GVFs per joint, switch_list, active_joint_index, switching
2.  repeat:
3.      observe EMG_C, EMG_S, s_1, s_2                  // Observe EMG control and switch signals, state
4.      get predictions p from GVFs given s_1           // Get predictions about joint motion from GVFs
5.      get predictions p_switch from GVFs given s_2    // Get predictions about switching from GVFs
6.      if not switching:                               // If switching flag set to false
7.          update switch_list based on magnitude p     // Update switching list
8.          if auto_switch(p_switch):                   // If joint prediction exceeds the preset threshold
9.              increment active_joint_index            // An autonomous switch is initiated
10.             switching ← true                        // Switching flag set to true
11.     if switch(EMG_S):                               // If a switch is initiated
12.         switching ← true                            // Switching flag set to true
13.         increment active_joint_index                // Switch to the next joint in the switching list
14.     move_joint(switch_list[active_joint_index], EMG_C)  // Move active joint according to EMG control signals
15.     if moved:                                       // If arm has moved
16.         switching ← false                           // Switching flag set to false
17.     for all gvf in adaptive switching GVFs:         // For each joint
18.         gvf.update(s_1)                             // Update GVF and learner using TD learning
19.     for all gvf in autonomous switching GVFs:       // For each joint
20.         gvf.update(s_2)                             // Update GVF and learner using GTD learning

switch_list = [S_1, ..., S_N]                   Cyclic list of all controllable joints
active_joint_index = index of switch list       Points to index of active joint in switching list
p = vector of predictions of motion of each joint   Adaptive switching predictions
p_switch = prediction of switching to a different joint starting from current joint   Autonomous switching predictions
switch()                                        Function that returns a boolean based on a switch input signal
move_joint()                                    Function that returns a signal that directs arm motion
auto_switch()                                   Function that returns a boolean based on a prediction input
```

Figure 5.2: The autonomous switching algorithm.

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 time spent switching and the total number of switches (manual and automatic) 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 instances in which the individual switched from one joint until a new 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 is 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 (Parker et al., 2014).

Prior to beginning the experiments, the subject was given 5 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 3 min); 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).

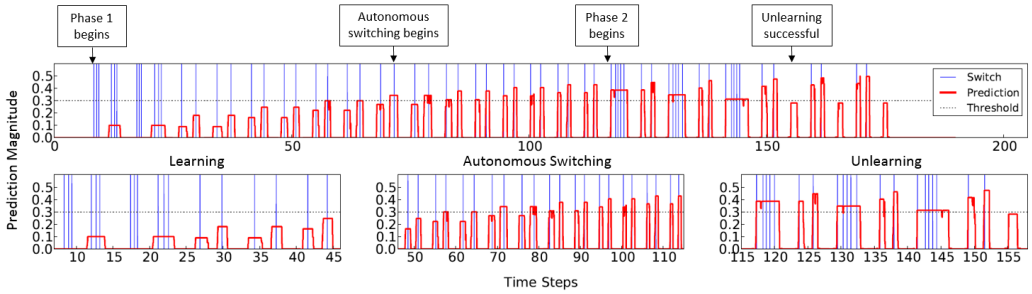


Figure 5.3: The Bento Arm learns to switch autonomously during a simple task, then unlearns switching when the task is changed. The signals in red represent predictions about switching to a different joint; the signals in blue represent binary switch signals; and the dotted line is the threshold above which autonomous switching will occur.

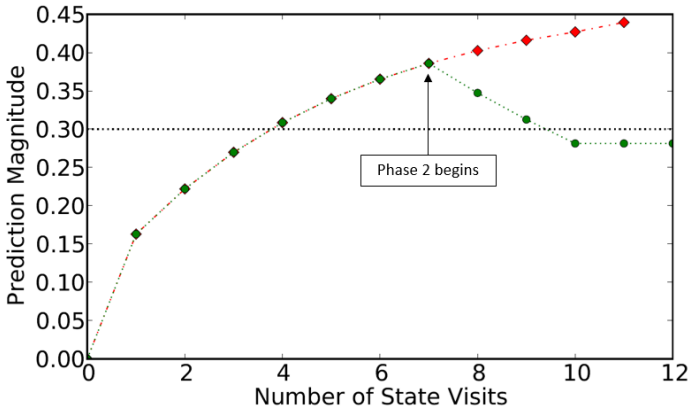


Figure 5.4: The magnitude of the switching predictions at similar states throughout the task. The green line represents the prediction magnitude as the Bento Arm's shoulder reaches the right side of its range; the red line represents the prediction magnitude as the shoulder reaches the left side.

5.3 Results

5.3.1 Experiment 1

Fig. 5.3 highlights how learned switching behaviour can also be unlearned through reinforcement. The results shown are from 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. 5.4. Each data point represents the magnitude of the prediction at identical states in the task. Here we define state visits as times in which the arm is in the same unique state (i.e., paused at the same position). Green points occur when the shoulder reaches the rightmost side of the space, and red points occur when the shoulder reaches the leftmost side. The dotted line represents the threshold for autonomous switching.

5.3.2 Experiment 2

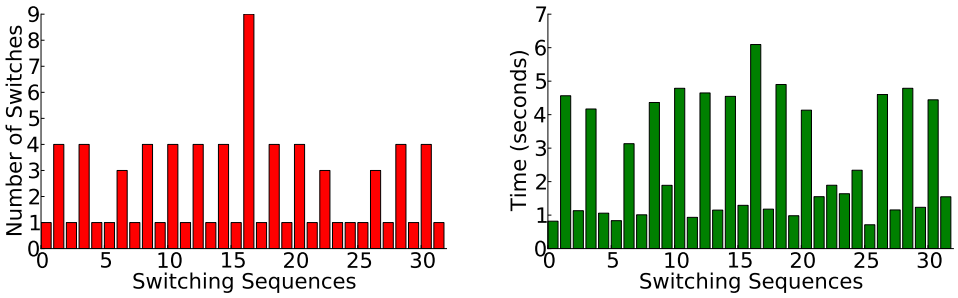
Fig. 5.5 compares the four control methods in terms of the switching sequences generated over the 3 minute testing period for one of the trials, clearly presenting the onset of autonomous switching in Fig. 5.5d (results are representative of the other three trials). Each repetition of the task required 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 represent the total number of manual switches plus autonomous switches.

Fig. 5.6 shows the average number of switches, the average time spent switching, and the average number of times the task was repeated during the three minutes of allotted time. In Fig. 5.6a, 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. 5.6b shows the average amount of time during the task that was dedicated to switching. The data in Fig. 5.6c 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.

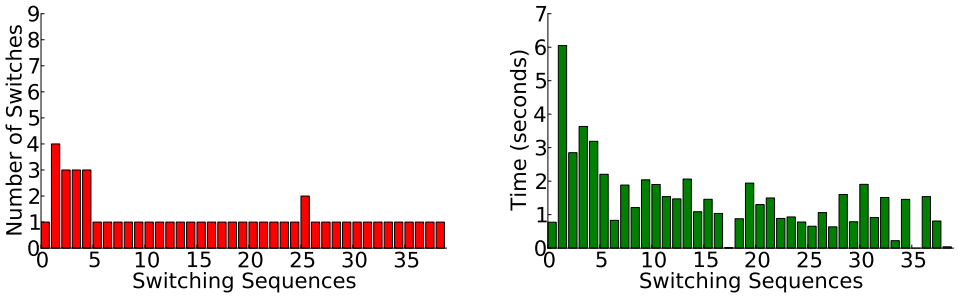
5.4 Discussion

In this work, our first aim was to demonstrate that our control approach was able to both learn and unlearn autonomous switching behaviours. In the left bottom panel shown in Fig. 5.3, the arm is learning to switch autonomously and the predictions do not 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 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.

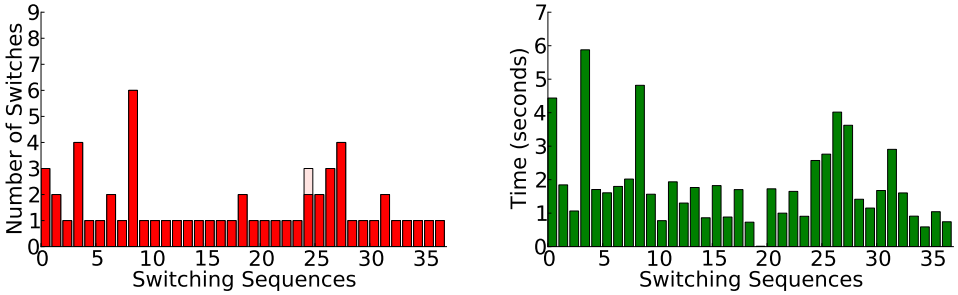
Fig. 5.4 clearly demonstrates the relative magnitude of the predictions compared to each other in similar states. It takes four visits to the same



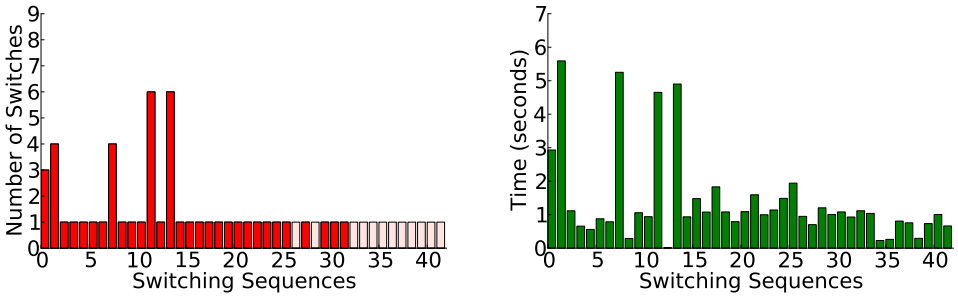
(a) Non-Adaptive



(b) Adaptive

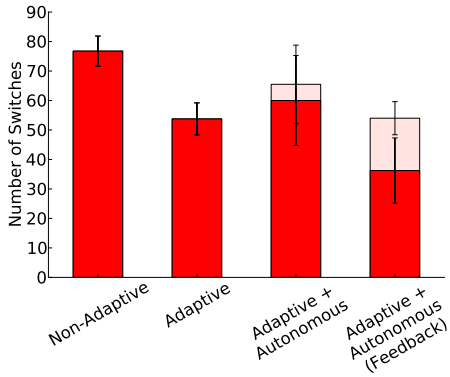


(c) Adaptive + Autonomous (No Feedback)

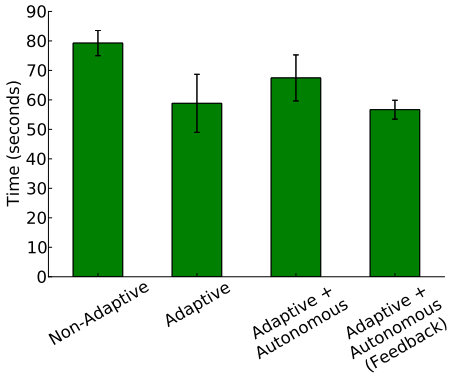


(d) Adaptive + Autonomous (With Feedback)

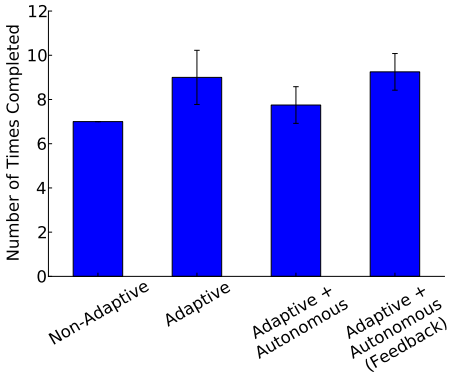
Figure 5.5: An example of switching sequences for each of the control methods in a simple, three-minute task. The number of switches per switching sequence are shown in red; autonomous switches are shown in pink; and the seconds per switching sequence are shown in green.



(a) The mean number of switches for each of the control methods. Red bars represent manual switches (made by the subject); pink shading represents autonomous switches.



(b) The mean time spent switching for each of the control methods.



(c) The mean times the task was completed for each of the control methods.

Figure 5.6: Results for an able-bodied subject completing the 3 min task with each of the four control methods, averaged over four semi-randomized trials.

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 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. 5.5 and 5.6.

Fig. 5.5a, the non-adaptive experiment, 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. 5.6, 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. 5.5b. 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 observed a marked improvement in Fig. 5.6 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 (Edwards et al., 2015).

Figs. 5.5c and 5.5d demonstrate what happens when autonomous switching is used in combination with adaptive switching. As a key result, autonomous switching was found to significantly reduce the number of manual switches re-

quired of the user—our primary measure of success. In the first case, without feedback, the subject was observed to make more errors when switching compared with the second case; 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 higher. 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 vibration 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 three-minute period in terms of significantly fewer manual switches required of the user. For example, in Fig. 5.5d, 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 effectiveness of autonomous switching before scaling up to more complex tasks such as the box-and-blocks task (Mathiowetz et al., 1985). In this 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 also 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).

The system was able to build up predictions relatively quickly in the task with the Xbox controller compared with the similar task involving EMG control. We have observed that myoelectric control adds a level of noise that reduces the overall effectiveness of autonomous switching. With different modes of control (e.g., a button or mechanical toggle) that have reduced intrinsic noise, we have observed increased performance on autonomous switching tasks. A possible explanation is that joystick control results in more precise movements and requires fewer fine motor adjustments from the user to achieve the intended arm position. Although shared control has potential rewards in human-machine interaction, in practice, autonomy and the way in which humans use machine autonomy is imperfect. For this reason, we have demonstrated the ability of the system to unlearn and reverse learned autonomy, which reverts control to the user.

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 (Tanabe et al., 2013), and are strong candidates for the application of adaptive and autonomous switching. We believe our approach will transfer well to other switching-based domains.

5.5 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 time 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.

Chapter 6

New Results from Autonomous Switching

6.1 Introduction

This chapter represents an addition to the work presented in the previous chapter. In preliminary experiments using the simple task (Chapter 5), autonomous switching decreased the number of switches made by the user (manual switches). In the experiment described below, we chose a more challenging task, and once again assessed the performance of autonomous switching with three subjects.

6.2 Methods

Three able-bodied subjects controlled the Bento Arm using EMG in a modified box-and-blocks task. Five balls were placed on one side of the divided box, and subjects were asked to pick up and drop each ball on the other side of the divider, as in Chapter 3. Successfully picking up and dropping all five balls comprised one iteration, and each iteration was repeated a total of 10 times. Subject 1 and Subject 2 had some prior experience controlling a myoelectric arm. Subject 3 was the experimenter, and an expert myoelectric robotic arm user. The third subject performed the task twice: once using EMG control and once using the joystick setup described in the previous chapter.

Three control methods were used: non-adaptive switching, adaptive switching, and adaptive switching combined with autonomous switching. For this

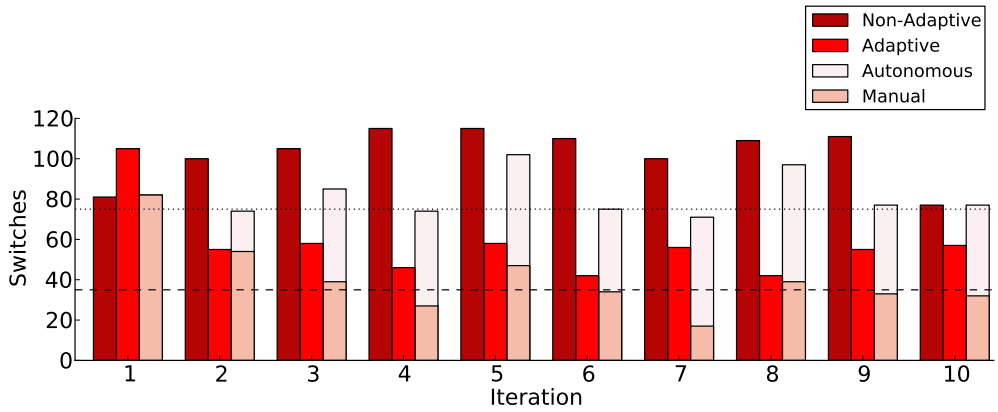
last control method, subjects were always given vibration feedback (as described in Chapter 5) to indicate an autonomous switch would occur. Preliminary data suggested that starting with the autonomous switching method is more difficult than starting with other methods; as a result, for subjects who had no prior experience with the system, we chose to stage the experiment in a way that was closer to the proper training environment, where subjects built up to more autonomous paradigms (adaptive switching and autonomous switching) after experiencing the current clinical default paradigm (non-adaptive switching). Thus, Subjects 1 and 2 completed the control methods in the order listed above. Subject 3, being familiar with the system, performed the task in the order of autonomous switching, adaptive switching, and non-adaptive switching. All subjects were given time to become familiar with each method prior to starting experiments.

This experiment was similar to the experiment described in Chapter 3, in which we compared subjects' performance on adaptive switching and non-adaptive switching using the modified box-and-blocks task. In this experiment, we tested an additional control method (autonomous switching with an adaptive switching list) and doubled the length of the task to 10 iterations. Subjects tested each control method once, without repeating trials.

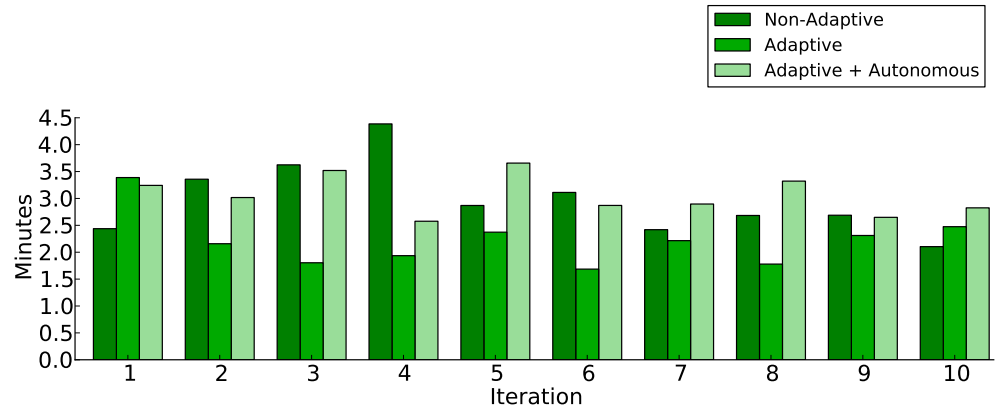
Adaptive and autonomous switching used the same TD(λ) and GTD(λ) algorithms as described in the previous chapter. However, the function approximation parameters of autonomous switching were altered for this more challenging task. All state inputs (except for the moving EMG averages) were instead each represented in a single tile and overlapped 40 times. The moving averages of both EMG signals were tiled into 6 bins.

6.3 Results and Discussion

Figs. 6.1 and 6.2 show the number of switches and time spent per iteration of the box-and-blocks task (for Subjects 1 and 2, respectively). Figs. 6.3 and 6.4 show the time and iterations for Subject 3 performing the task using EMG control and joystick control, respectively. The dark red bars in each of Figs.

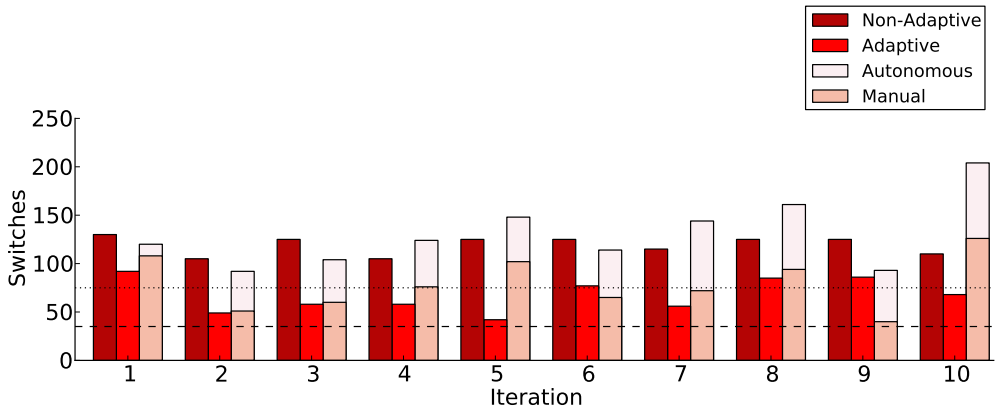


(a) The number of switches per iteration of the box-and-blocks task. Three types of control methods were tested: non-adaptive (dark red), adaptive (light red), and adaptive with autonomous switching. Autonomous switches are shown in dark pink and manual switches are shown as light pink.

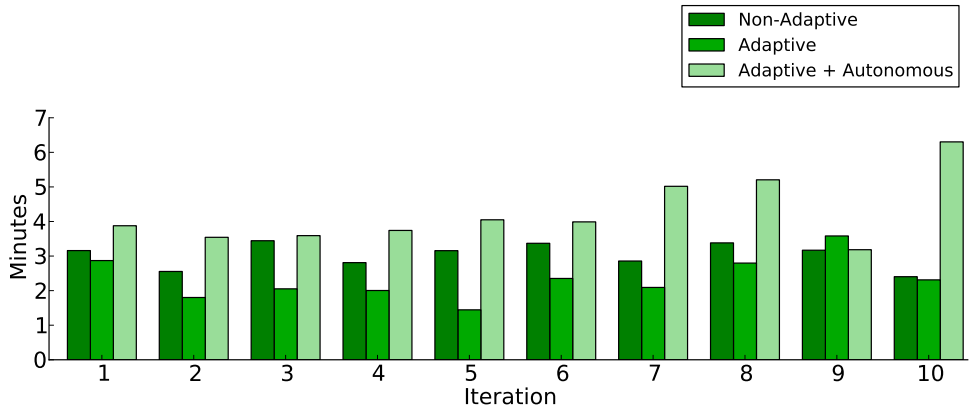


(b) The time (in minutes) spent on each iteration of the box-and-blocks task. The control methods are represented as follows: non-adaptive (dark green), adaptive (medium green), and adaptive with autonomous switching (light green).

Figure 6.1: Number of switches and time per iteration for Subject 1.

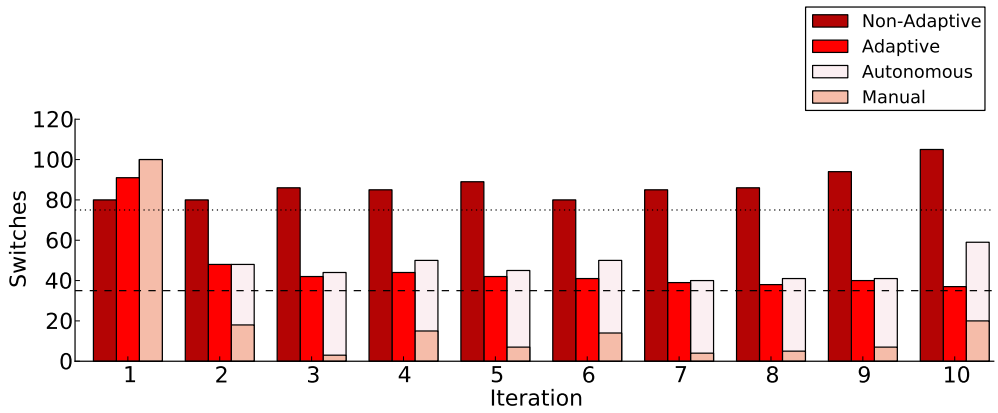


(a) The number of switches per iteration of the box-and-blocks task. Three types of control methods were tested: non-adaptive (dark red), adaptive (light red), and adaptive with autonomous switching. Autonomous switches are shown in dark pink and manual switches are shown as light pink.

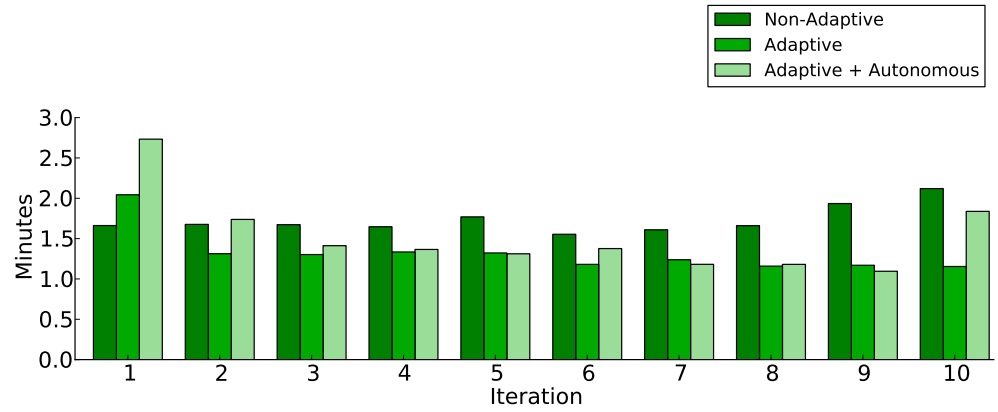


(b) The time (in minutes) spent on each iteration of the box-and-blocks task. The control methods are represented as follows: non-adaptive (dark green), adaptive (medium green), and adaptive with autonomous switching (light green).

Figure 6.2: Number of switches and time per iteration for Subject 2.

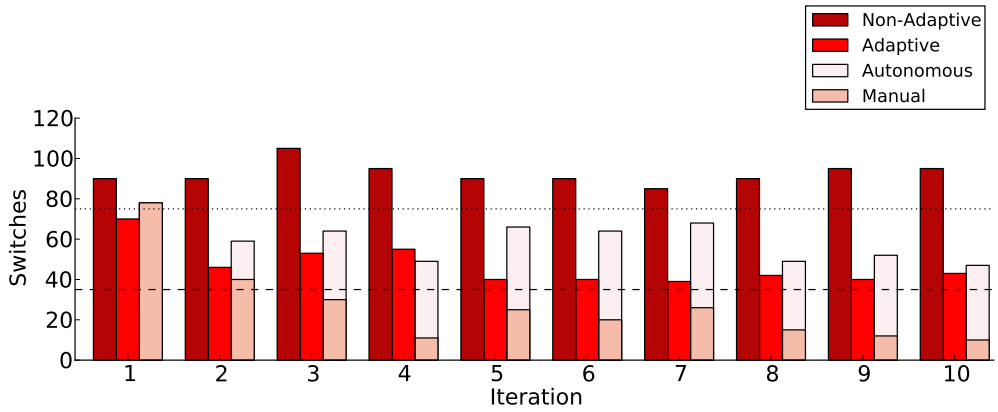


(a) The number of switches per iteration of the box-and-blocks task. Three types of control methods were tested: non-adaptive (dark red), adaptive (light red), and adaptive with autonomous switching. Autonomous switches are shown in dark pink and manual switches are shown as light pink.

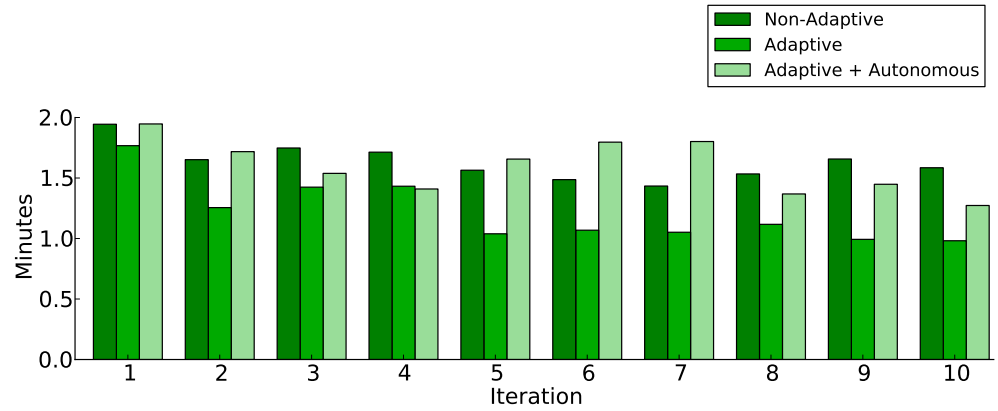


(b) The time (in minutes) spent on each iteration of the box-and-blocks task. The control methods are represented as follows: non-adaptive (dark green), adaptive (medium green), and adaptive with autonomous switching (light green).

Figure 6.3: Number of switches and time per iteration for Subject 3 performing the task with EMG control.



(a) The number of switches per iteration of the box-and-blocks task. Three types of control methods were tested: non-adaptive (dark red), adaptive (light red), and adaptive with autonomous switching. Autonomous switches are shown in dark pink and manual switches are shown as light pink.



(b) The time (in minutes) spent on each iteration of the box-and-blocks task. The control methods are represented as follows: non-adaptive (dark green), adaptive (medium green), and adaptive with autonomous switching (light green).

Figure 6.4: Number of switches and time per iteration for Subject 3 performing the task with joystick control.

6.1a, 6.2a, 6.3a, and 6.4a represent the number of switches made by the subject in each iteration of the non-adaptive control method and the light red bars represent switches made in the adaptive method. Dark pink bars represent the number of autonomous switches made by the system, and the light pink represents the number of manual switches made by the user. The dotted line denotes the minimum number of switches (75 switches) required to complete the task given a non-adaptive switching list. The dashed line shown at 35 switches represents the best possible outcome for an adaptive switching list—it is the minimum number of switches required once the system has learned the optimal switching list for the given task, assuming no errors on the part of the user.

In each of Figs. 6.1b, 6.2b, 6.3b, and 6.4b, dark green is the total amount of time (in minutes) in each non-adaptive iteration and medium green is the time in each adaptive iteration. Light green bars represent the time spent on iterations that used the adaptive switching list with autonomous switching.

Figs. 6.1, 6.2, 6.3, and 6.4 demonstrate the effect of autonomous switching with feedback on a more complex task (i.e., box-and-blocks). In Chapter 3, we presented results from adaptive switching on the same modified box-and-blocks task, and specified the optimal number of switches for adaptive switching. An adaptive switching list requires the user to manually switch their control between joints at least 35 times during a single iteration. In Fig. 6.1a, by the second iteration, fewer manual switches are required when autonomous switching is used compared with the non-adaptive and the adaptive switching list. Furthermore, during several iterations, the number of manual switches falls below the optimal for adaptive switching (35 switches). This implies that, for some users, autonomous switching can decrease the number of interactions between a user and their device.

Results from Subject 2 show that autonomous switching does not save manual switches for all users. Furthermore, compared to adaptive switching by itself, autonomous switching increased the overall number of switches (the combination of manual and autonomous switches). A possible explanation for this increase is that occasionally, autonomous switches were made at the

wrong time, and had to be corrected by the user. Both Subject 1 and Subject 2 reported that the arm switched autonomously at times they did not expect. For example, while controlling the shoulder joint, the arm will learn to switch automatically when motion has stopped and the hand is in line with a ball. This caused some frustration for users when they had simply paused motion to make fine adjustments in the arm's position. Although the arm is learning a way of behaving that is correct in one context, there is likely a need to provide the system with a greater ability to distinguish between these different, but similar, situations: a user stopping the arm precisely in line with a ball; a user stopping the arm directly in line with the position of a ball that has been moved (either to the other side of the box or nudged out of place); and a user briefly stopping the arm's motion near a ball to make small motor adjustments to the joint they are currently controlling.

Although autonomous switching resulted in fewer manual switches in some instances, the corresponding total time was not always reduced. During experiments, we observed that users would often wait for the arm to switch if they anticipated an autonomous switch (based on vibration feedback or previous switches in the same context). Furthermore, in some instances the user had to correct poorly timed autonomous switches by switching an additional four times, back to the joint they had been using. This correction causes both an increase in the number of manual switches and the total time.

In Fig. 6.3, Subject 3 showed a significant decrease in the number of manual switches during the autonomous switching part of the task, often having to manually switch fewer than 5 times. Results from the same task using joystick control reflect a similar decrease in switches as well. By the second or third iteration in both of the Subject 3 data sets, the number of manual switches had fallen below the dashed line. In the EMG-controlled experiment, while the iteration time of autonomous switching was not as high as that of non-adaptive switching, it was more or less equivalent to the time spent doing adaptive switching. However, the results from Fig. 6.4b with joystick control showed an increased amount of time needed in autonomous switching. It is interesting to note that the joystick control task reduced the overall time in

each of non-adaptive, adaptive, and autonomous switching. Although joystick control offers smoother movements in general, it is possible that EMG control outperformed the joystick because joystick control facilitates time savings in all three control methods. Fatigue also appears to have played a role in the EMG-control task, and is likely a factor in the significant increase in the number of minutes per iteration at the end of the non-adaptive task.

Although autonomous switching has shown promise in reducing the number of switches and time spent on the box-and-blocks task, it has not shown consistent benefits for all Bento Arm users. It is clear that some challenges remain to be addressed, particularly the challenge of better differentiating between states in which autonomous switches should occur.

6.4 Conclusions

In this work, we compared autonomous switching with the non-adaptive and adaptive methods of switching, continuing work from the previous chapter by adding to the complexity of the task. The next step in this work will likely be to address the present limitations of autonomous switching, modifying our current approach to make control using autonomous switching easier and more functional.

Chapter 7

Discussion and Future Directions

The work in this thesis involved three main contributions: the application of an algorithm, adaptive switching, to optimize the gated control method of driving a myoelectric arm; the development of unlearning as a means for the user to maintain control of their device; and the augmentation of adaptive switching with autonomous switching. We discussed two main aims in the first chapter. The first was to improve gated control of a myoelectric arm using RL. The second was to find novel algorithms that could be applied in other settings. Here, we discuss the significance of our contributions with respect to these aims. We also discuss the limitations of the work and possible future extensions.

7.1 Significance and Discussion of Contributions

7.1.1 Addressing User Needs

As discussed in Chapter 2, there are three main causes for the rejection of powered prosthetic limbs: lack of intuitive control, lack of function, and lack of feedback. In our approach for improving gated control, we have tried to address each of these concerns such that the requirements of persons with amputations are met, and to ensure future adaptations of this work could be deployed in clinical settings.

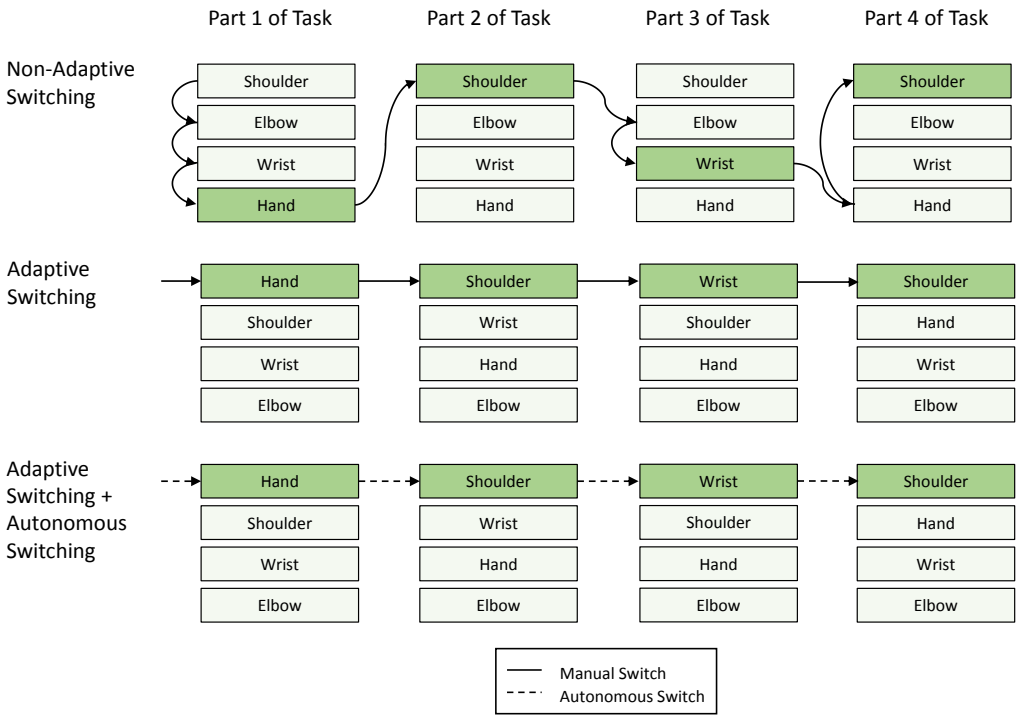


Figure 7.1: A comparison of non-adaptive switching, adaptive switching, and adaptive switching with autonomous switching using a robot arm with four joints. In this example, a subject is given a simple manipulation task that requires opening/closing the hand, rotating the shoulder, flexing/extending the wrist, and rotating the shoulder again. With the fixed joint list used in non-adaptive switching, the user would need to manually switch 8 times to complete the task. In comparison, after learning by the system, adaptive switching would require only 4 manual switches. Furthermore, adaptive switching with autonomous switching should require no manual switches (the system would perform all 4 switching actions automatically). In both adaptive switching cases, the list of joints is contextually re-ordered, resulting in a more efficient switching list being updated and presented to the user. Solid lines represent manual (user) switches, and dashed lines represent autonomous (system) switches.

Results from adaptive switching have shown a decrease in experimental time and number of switches in multiple experiments, particularly for experienced subjects. While improvements in these areas are not necessarily direct evidence of a more intuitive user experience, overall, subjects reported greater ease of use for adaptive switching than for non-adaptive switching. With adaptive switching, we also give users access to a larger number of functions than would be possible with typical gated control schemes. Although we limited the number of controllable joints in each of our experiments to no more than five, with additional learning time, adaptive switching could conceivably be applied to myoelectric arms with many more functions, comparable to the number of functions (i.e., grip patterns) offered by many commercial devices. All of our experiments incorporated feedback (visual, audio, or vibrotactile) to clearly communicate to the user the results of learning and the timing of automated actions.

7.1.2 User Trust

We believe that the predictability of the autonomy played a pivotal role in users' trust of the myoelectric arm. As subjects became more familiar, both with their execution of a task and with the role of machine learning as it adapted to a task, they often came to trust the autonomy more. There were certain regions of the task space in particular, in which the learning system performed with close to 100% prediction accuracy; in these regions, it was clear from subjects' behaviour that they felt less need to monitor the myoelectric arm.

An example of this behaviour occurred consistently during the modified box-and-blocks task after learning had occurred. Upon grasping a ball with the gripper and moving it across the divider, users would quickly switch to the next joint in the list; then, without waiting for auditory confirmation that the system had switched to the desired joint (the gripper), they would immediately move the newly active joint, thereby releasing the ball as intended. This suggested a certain level of trust in the adaptive switching list—that it had successfully learned and maintained accurate predictions about joint motion.

A similar behaviour was observed in the experienced user when performing the box-and-blocks task with autonomous switching. After the arm had consistently switched by itself in similar situations, the user began to anticipate autonomous switches, often moving the next joint prior to hearing the cue alerting them to the switch.

7.1.3 Unlearning and Feedback

The effect of user trust on the performance of a task hints at the importance of the user-centred approach to the development of new control methods. This includes giving the user sufficient ability to correct their device and providing them with ample feedback about the device's autonomous actions.

The unlearning aspect of our experiments was an important step toward a functional and adaptable autonomous system. In adaptive switching, the system never initiates actions for the user; instead, predictions serve as suggestions for a more context-appropriate switching order. In autonomous switching, however, sufficiently high predictions lead to the system initiating switching actions of its own accord. Unlearning ensures that only switching actions that are acceptable to the user are given credit. Extraneous autonomous switches that are corrected by the user will gradually decrease the prediction magnitude, keeping the user in full control while simultaneously decreasing the switching cost in times of high system confidence.

From experiments in Chapter 5, we found that shared control without feedback yielded much worse performance than the same experiments with feedback. This is not the first study to directly use predictions as feedback. Our work follows a previous study in which learned predictions acted as feedback about anticipated robot arm collisions within a workspace (Parker et al., 2014). The learning system contains key forecasts about future events that should be communicated to the user, but which are often used only for control. We believe that communication in this way closes the loop between the user interacting with the machine learning system, and this is reflected in the increased performance from both studies.

7.1.4 Adaptations

The results of autonomous switching in both simple and more complex tasks point to differences in the way subjects approach tasks. Users who have an understanding of GVFs and our specific learning system tend to perform actions that will benefit learning, allowing the machine to build up expectations about their behaviour, faster. For example, expert users will often move the myoelectric arm in repeatable ways, taking advantage of the way the task space is partitioned into different states. During adaptive switching experiments, the same overarching goal (move the ball to the other side of the box) could be accomplished by selecting joints each time in a different sequence. The expert users were more likely to accept the first joint suggested to them by the learning system, even if it was not what they would have chosen originally. In autonomous switching tasks, expert users compensate for unintended autonomous switches with smoother, precise arm motions. These are examples of a mutual adaptation that occurs: while the system learns about user behaviour, users will also adapt to the way the machine learns. While this mutual adaptation is beneficial to performance, it is uncertain whether less experienced users will adapt in similar ways.

7.2 Limitations of the Current Work

7.2.1 Study Population

The local population of amputees available to participate in experimental trials is relatively small. Consequently, for most of our experiments, we recruited able-bodied subjects. We expect the performance of non-amputee subjects to be similar to that of amputee subjects that have equivalent experience with myoelectric control. In addition, experiments were often treated as case series or individual case studies to highlight specific differences in task performance. However, this work warrants a more comprehensive study involving a larger sample of amputee and able-bodied subjects.

7.2.2 Challenges in Autonomous Switching

Although autonomous switching has been shown to reduce a user's manual switches, one limitation that remains to be addressed is the unintentional reinforcement of switches in specific situations. When the learner performs actions the user does not expect, trust between the user and the system begins to degrade. Currently, the system is unable to distinguish between certain key contexts; for example, between times when a ball is in its starting position and times when it has already been moved. Although similar, these contexts demand altogether different switching behaviours from the autonomous system. In the first case, autonomous switching is a desired behaviour; in the second case, the user may not want to immediately switch to a different joint, and an autonomous switch in such a situation would serve only to frustrate the user. One potential solution is to give the learning system additional state information. Visual information from a camera, such as colour averaging or object recognition, may be valuable for tasks like the box-and-blocks task, in which contextual cues can be learned from colour and shape.

In the second highlighted case above, the user may be unable to correct the system when it switches unexpectedly. Often, an autonomous switch occurs at the same instant the user resumes motion of their prosthesis; to the learning system, this signifies the user has "accepted" the switch to a new joint, and therefore the switch is unintentionally reinforced, despite being incorrect. While additional state information may indeed help solve this problem, an alternative approach may be to introduce a suitable delay after an autonomous switch that would give users time to realize and react to a change in the active joint. During this brief delay, which could be determined empirically, the user would be able to move the arm, but this movement would not serve to reinforce the switch. Another approach to the same problem would be a delay which prevents the arm from moving altogether.

We used a simple threshold as the trigger for autonomous switching. However, different measures of confidence may be more effective, and provide another solution to autonomous switching and the switches that originate from

an inability to differentiate between specific contexts. A system that measures variance, in addition to a threshold, may be more appropriate. A high variance in the cumulant (switch signal) of a given state may be an indication that the system should not switch for the user (Sherstan et al., 2016).

One observation that has frequently emerged over the course of experiments involving autonomous switching is that there is a marked difference in performance between EMG and joystick control. EMG signals have more intrinsic noise than mechanical toggles, and as such, myoelectric control requires a learning system that is robust to occasional imprecision in arm motion that results from noise. One solution to counter the effects of noise may be to introduce a separate learning agent that adjusts EMG parameters with increased noise induced by muscle fatigue and other factors. Another possibility may be to use two separate thresholds for EMG control: one at the level of initiating arm motions, and one at the level of terminating arm motions.

Despite some remaining challenges, this work offers promise for reducing the time and cognitive cost to switching. Autonomous switching, with some modifications, therefore merits further investigation as an extension to the adaptive switching control method.

7.2.3 Tasks

One criticism of the learning system often stems from the fact that in our experiments, the tasks given to subjects are typically repetitive in nature. The simple task described in a few of our studies involved repeating simple joint actions as many times as possible. Even the box-and-blocks task, although not as strictly repetitive, involves many iterations of similar series of actions. The assessment that the system learns from prior experience and therefore requires some repetition is correct. However, on a larger scale, everyday tasks and actions are typically repetitive to some degree. Putting dishes away, for example, requires similar gross arm movements—moving from the dish rack, to the cupboard, and back again. Furthermore, there is benefit to a system that learns from repetition, in that it may reduce the burden and fatigue during some of the most repetitive (and thus switching intensive) tasks in daily life.

7.3 Future Work

Future investigations will continue to test adaptive and autonomous switching in more challenging task domains, incorporating some of the changes to autonomous switching outlined above. We will also test these algorithms on a larger population of amputee and able-bodied subjects to gain more insight into performance averages. In this work, we limited the workspace to the table top; however, the Bento Arm was also designed to be a wearable system. Therefore, in future experiments, amputee subjects may be asked to perform tasks while wearing the arm.

We also plan to integrate more sensors into the robotic arm, and use these added signals to bolster the stream of information provided to the learner. In addition, we will increase the number of feedback channels available to the user. For example, it may be useful to communicate not only *when* a switch will happen, but also *which* joint will become active upon switching. This information could be conveyed to the user visually, through LEDs embedded in the joints of the Bento Arm.

Ultimately, we would like to test these algorithms on myoelectric arms that are capable of switching between various grip modes, as in many commercial prostheses. This work may also be applicable to other switching-based control systems. Recent work has been done by Tanabe et al. on powered orthotic devices that are capable of operating in different modes, which users can actively switch between with a mechanical trigger (Tanabe et al., 2013). Autonomous switching has potential utility in a similar system, learning context-appropriate times to switch between modes of operation on behalf of the user.

Chapter 8

Conclusions

State-of-the-art myoelectric arms offer numerous grip actions that can be individually controlled by amputees. However, a major constraint that leads to marked degradation of performance is the limited number of intuitive myoelectric signals that can be used to control those myoelectric prostheses. There is a significant mismatch between the number of controllable prosthesis functions and the possible EMG control sites on an amputee's residual limb. This is a barrier that seriously hinders an amputee's ability to use their prosthesis to their fullest potential.

A common approach that gives amputees access to multiple functions is called gated, or switched control. Using gated control, users select the function they wish to control by switching through a list of options. However, gated control cannot provide amputees with access to more than three or four possible functions without significant costs to switching time and mental effort. In this work, we have tried to address the limitations of the gated control method through real-time machine learning. Specifically, we applied RL techniques to myoelectric arms, with the goal of creating an arm that is capable of sharing the control burden with users by learning about and adapting to users' needs during continued use. To this end, we developed two algorithms that assist with the control of a myoelectric robot arm: *adaptive switching* and *autonomous switching*. During adaptive switching, the machine uses contextual factors to predict which prosthetic function a user intends to use next. During autonomous switching, the machine learns predictions about when a user

intends to switch to a different function, then begins to initiate switches for the user once it reaches a pre-established level of confidence in its predictions.

We carried out several studies, testing these algorithms against conventional gated control, which we call non-adaptive switching. We compared results from subjects performing tasks multiple times using different control methods. We found that in experienced users, adaptive switching can reduce the task time by approximately 20% compared with non-adaptive switching. Furthermore, adaptive switching reduces the number of switches by half. When combining adaptive switching with autonomous switching, we see similar time savings for experienced subjects, and an additional reduction in the number of switches made by the user (the number of manual switches made during autonomous switches amounted to less than half the number of switches required during adaptive switching). In the process of developing these alternative switching methods, we also provided the learning system with a means of unlearning autonomous behaviour.

We believe this work was successful in addressing a major challenge of myoelectric arm use and the gated control method specifically. The adaptive and autonomous switching methods increase the number of accessible prosthesis functions while reducing the overall switching cost to the user. In addition, we implemented a more adaptive autonomy by giving the system the ability to unlearn with changes to the environment or user goal. There are still some limitations to be overcome; in particular, autonomous switching does not currently differentiate between certain essential contexts, occasionally causing the arm to switch when a switch is unsolicited. In the future, we will continue to enhance these RL algorithms, assessing their performance in more in-depth studies and more demanding user tasks.

References

L C Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*, pages 30–37, 1995.

E Biddiss and T Chau. Upper-limb prosthetics: Critical factors in device abandonment. *American Journal of Physical Medicine and Rehabilitation*, 86:977–987, 2007.

B T Carlsen, P Prigge, and J Peterson. Upper extremity limb loss: Functional restoration and targeted reinnervation to transplantation. *Journal of Hand Therapy*, pages 106–114, 2014.

C Castellini, P Artemiadis, M Wininger, A Ajoudani, M Alimusaj, A Bicchi, B Caputo, W Craelius, S Dosen, K Englehart, D Farina, A Gijsberts, S B Godfrey, L Hargrove, M Ison, T Kuiken, M Marković, P M Pilarski, R Rupp, and E Scheme. Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography. *Frontiers of Neurorobotics*, 8:22, 2014.

C Cipriani, F Zaccone, S Micera, and M C Carrozza. On the shared control of an emg-controlled prosthetic hand: Analysis of user-prosthesis interaction. *IEEE Transactions on Robotics*, 24:170–184, 2008.

M R Dawson, F Fahimi, and J P Carey. The development of a myoelectric training tool for above-elbow amputees. *Open Biomedical Engineering Journal*, 6:5–15, 2012.

M R Dawson, C Sherstan, J P Carey, J S Hebert, and P M Pilarski. Development of the bento arm: An improved robotic arm for myoelectric training and research. In *Proceedings of the 2014 Myoelectric Controls Symposium*, pages 60–64, Fredericton, NB, Canada, 2014.

A L Edwards, M R Dawson, A Kearney, R S Sutton, and P M Pilarski. Temporal-difference learning to assist human decision making during the control of an artificial limb. In *Proceedings of the 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making*, Princeton, NJ, USA, October 25-27, 2013.

A L Edwards, M R Dawson, J S Hebert, C Sherstan, R S Sutton, K M Chan, and P M Pilarski. Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching. *Prosthetics and Orthotics International*, 2015. [Online early access.] DOI: 10.1177/0309364615605373. Published Online: September 30, 2015.

M A Goodrich, D R J Oslen, J W Crandall, and T J Palmer. Experiments in adjustable autonomy. In *IEEE International Conference on Systems, Man, and Cybernetics*, Tucson, USA, 2001.

T A Kuiken, G A Dumanian, R D Lipschutz, L A Miller, and K A Stubblefield. The use of targeted muscle reinnervation for improved myoelectric prosthesis control in a bilateral shoulder disarticuation amputee. *Prosthetics and Orthotics International*, 28:245–253, 2004.

L.-J Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8:293–321, 1992.

D J Linden. From molecules to memory in the cerebellum. *Science*, 301: 1682–1685, 2003.

V Y Ma, L Chan, and K J Carruthers. Incidence, prevalence, costs, and impact on disability of common conditions requiring rehabilitation in the united states: Stroke, spinal cord injury, traumatic brain injury, multiple sclerosis, osteoarthritis, rheumatoid arthritis, limb loss, and back pain. *Archives of Physical Medicine and Rehabilitation*, 95:986–995, 2014.

H R Maei. *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta, Edmonton, AB, Canada, 2011.

V Mathiowetz, G Volland, N Kashman, and K Weber. Adult norms for the box and blocks test of manual dexterity. *The American Journal of Occupational Therapy*, 39:386–391, 1985.

S Micera, J Carpaneto, and S Raspopovic. Control of hand prostheses using peripheral information. *IEEE Reviews in Biomedical Engineering*, 3:48–68, 2010.

J Modayil, A White, P M Pilarski, and R S Sutton. Acquiring a broad range of empirical knowledge in real time by temporal-difference learning. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1903–1910, Seoul, South Korea, October 14-17, 2012a.

J Modayil, A White, and R S Sutton. Multi-timescale nexting in a reinforcement learning robot. In *Proceedings of the 2012 Conference on Simulation of Adaptive Behavior*, pages 299–309, Odense, Denmark, 2012b.

A S R Parker, A L Edwards, and P M Pilarski. Using learned predictions as feedback to improve control and communication with an artificial limb: Preliminary findings. [Online]. Available: arXiv:1408.1913 [cs.AI], 2014. [Accessed Dec 21, 2015].

P Parker, K Englehart, and B Hudgins. Myoelectric signal processing for control of powered limb prostheses. *Journal of Electromyography and Kinesiology*, 16:541–548, 2006.

B Peerdeman, D Boere, H Witteveen, R Huis in't Veld, H Hermens, S Stramigioli, H Rietman, P Veltink, and S Misra. Myoelectric forearm prostheses: state of the art from a user-centered perspective. *Journal of Rehabilitation Research and Development*, 48:719–738, 2011.

P M Pilarski and R S Sutton. Between instruction and reward: Human-prompted switching. In *AAAI 2012 Fall Symposium on Robots Learning Interactively from Human Teachers (RLIHT)*, pages 46–52, Arlington, VA, USA, November 2-4, 2012. AAAI Technical Report FS-12-07.

P M Pilarski, M R Dawson, T Degris, J P Carey, and R S Sutton. Dynamic switching and real-time machine learning for improved human control of assistive biomedical robots. In *Proceedings of the 4th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 296–302, Roma, Italy, June 24-27, 2012.

P M Pilarski, M R Dawson, T Degris, J P Carey, K M Chan, J S Hebert, and R S Sutton. Adaptive artificial limbs: A real-time approach to prediction and anticipation. *IEEE Robotics & Automation Magazine*, 20:53–64, 2013a.

P M Pilarski, T B Dick, and R S Sutton. Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints. In *Proceedings of the 2013 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–8, Seattle, USA, June 24-26, 2013b.

A D Redish. *The mind within the brain: How we make decision and how those decisions go wrong*. Oxford University Press, New York, USA, 2013.

L Resnik, M R Meucci, S Lieberman-Klinger, C Fantini, D L Kelty, R Disla, and N Sasson. Advanced upper limb prosthetic devices: Implications for upper limb prosthetic rehabilitation. *Archives of Physical Medicine and Rehabilitation*, 93:710–717, 2012.

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. *Journal of Rehabilitation Research and Development*, 48:643–660, 2011.

J S Schofield, K R Evans, J P Carey, and J S Hebert. Applications of sensory feedback in motorized upper extremity prosthesis: A review. *Expert Review Medical Devices*, pages 1–13, 2014.

J W Sensinger, B A Lock, and T A Kuiken. Adaptive pattern recognition of myoelectric signals: Exploration of conceptual framework and practical algorithms. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17:270–278, 2009.

T P Sheehan and G C Gondo. Impact of limb loss in the united states. *Physical Medicine and Rehabilitation Clinics of North America*, 25:9–28, 2014.

C Sherstan, M Machado, A White, and P Pilarski. Introspective agents: Confidence measures for general value functions. *Ninth Conference on Artificial General Intelligence (AGI-16)*, July 16-19, 2016. To appear.

R S Sutton and A Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, Massachusetts, 1998.

R S Sutton and A G Barto. *Learning and computational neuroscience: Foundations of adaptive networks*, chapter Time-derivative models of pavlovian reinforcement, pages 497–537. MIT Press, Cambridge, Massachusetts, 1990.

R S Sutton, J Modayil, M Delp, T Degris, P M Pilarski, A White, and D Precup. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 761–768, Taipei, Taiwan, May 2-6, 2011.

S Tanabe, S Hirano, and E Saitoh. Wearable power-assist locomotor (wpal) for supporting upright walking in persons with paraplegia. *Neurorehabilitation*, 33:99–106, 2013.

A L Thomaz and C Breazeal. Teachable robots: understanding human teaching behaviour to build more effective robot learners. *Artificial Intelligence*, 172:716–737, 2008.

A White. *Developing a predictive approach to knowledge*. PhD thesis, University of Alberta, Edmonton, AB, Canada, 2015.

T W Williams. Guest editorial: progress on stabilizing and controlling powered upper-limb prostheses. *Journal of Rehabilitation Research and Development*, 48:ix–xix, 2011.

D M Wolpert, Z Ghahramani, and J R Flanagan. Perspectives and problems in motor learning. *Trends Cogn. Sci.*, 5:487–494, 2001.

K Ziegler-Graham, E J MacKenzie, P L Ephraim, T G Travison, and R Brookmeyer. Estimating the prevalence of limb loss in the united states: 2005 to 2050. *Archives of Physical Medicine and Rehabilitation*, 89:422–429, 2008.

K J Zuo and J L Olson. The evolution of functional hand replacement: From iron prostheses to hand transplantation. *Canadian Journal of Plastic Surgery*, 22:44–51, 2014.