

Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control

by

Pouria Faridi

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

Master of Science

Neuroscience
University of Alberta

© Pouria Faridi, 2023

Abstract

The overall goal of this work was to design an intelligent method to reduce the cognitive and physical burdens associated with walking using lower-limb exoskeletons after paralysis. Lower-limb exoskeletons with many operating modes (i.e., walking patterns) can be challenging to work with. Manufacturers have allocated a switch button, allowing their users to select an operating/walking mode by switching through a list of available modes. This approach however, consumes a lot of time and energy from users, as they have to switch many times to get their desired mode at each switching instance. The work in this thesis used temporal-difference (TD) learning from the field of computational reinforcement learning (RL), that requires no previous modeling and/or a training dataset, to reduce the switching-related issues. Through the use of biologically-inspired general value functions (GVFs), an adaptive controller (referred to as adaptive switching method) was designed to reduce the number of required switching actions on the part of the user and limit it to a single switching action (one time hitting a switch button) at each switching instance. The adaptive switching method used the environmental and contextual representations to create predictions on the future usage of each operating/walking mode, specific to each individual. Using TD learning, the predictions about the GVFs related to each operating/walking mode were updated and adapted to the exoskeleton users' (the experimenters) preferences.

Three users each performed three unique experimental scenarios, wearing the exoskeleton and using the adaptive switching method. The scenarios were designed to be most representative of the real-world situations. Adaptive switching method created a ranking mechanism in the switching list, ranking the operating modes based on their likelihood of being used next, from top of the list to the bottom. The order of the operating modes in the switching list was updated

at each time step. Learning parameters (e.g., learning weights) were initialized to zero and built upon users' switching behavior. Predictions were quickly learned and formed the ideal order of the modes in the switching list based on the users' walking patterns. In the case of uncertainties (i.e., when more than one operating mode could be utilized), the machine-learned method (adaptive switching) was able to predict all of the likely mode utilizations and ranked the desired modes at the top of the switching list. When a change in the users' behavior was seen, the adaptive controller was able to quickly adapt to that changing behavior, unlearn the previous behavior and learn the new walking pattern. The adaptive controller did not force the users to select a mode, but optimized their switching actions. This work demonstrated that the developed machine-learned controller can adapt to different walking behaviors and changing environments, without the need for offline training. It created an avenue for personalized walking and smart, optimized human-robot interactions.

This proof-of-concept work is the first demonstration of GVF prediction and learning in lower-limb exoskeleton control. The outcomes of this work contribute to the fields of neuroscience, robotics, computing science and engineering, and sets the path for further investigation of biologically-inspired learning methods in wearable robots and human-robot interactions.

Preface

This thesis is an original work by Pouria Faridi. All of the experiments were conducted in the Dr. Vivian K. Mushahwar's laboratory at the University of Alberta. This study was conceived by Dr. Vivian K. Mushahwar, Dr. Patrick M. Pilarski and Pouria Faridi. I designed and programmed the machine-learned algorithm. Experimental scenarios were designed by Pouria Faridi and Vivian K. Mushahwar. The required sensors for the study were determined by Pouria Faridi, Mahdi Tavakoli, Patrick M. Pilarski and Vivian K. Mushahwar. The exoskeleton joint controller was designed by Pouria Faridi, Mojtaba Sharifi and Javad Mehr. Experiments were conducted by Pouria Faridi, Javad Mehr and Don Wilson. The sensors' 3D enclosures mounted on the walker were designed, printed and mounted by Don Wilson. Arduino programming for data transmission from wireless sensors was performed by Don Wilson and Frank Gauthier. Pouria Faridi analyzed all data.

The literature review in chapter 1 was written by Pouria Faridi. Chapters 2 and 3 were written by Pouria Faridi, with feedback from co-authors. Chapter 2 was published as a proceeding of the 2022 International Conference on Rehabilitation Robotics (ICORR), Rotterdam, Netherlands, doi: 10.1109/ICORR55369.2022.9896611. Chapter 3 has been submitted to a peer-reviewed journal. Chapter 4 was written by Pouria Faridi.

Dedication

This work is dedicated to my mother and my sister, who supported me in all of the stages of my life.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Vivian K.

Mushahwar, for the continuous support of my MSc study and my professional development; for the scientific and professional opportunities provided and supported by her, and for her motivation, patience, and immense knowledge.

I would like to thank my MSc supervisory committee, Dr. Patrick M. Pilarski and Dr. Dave Collins for their encouragement and insightful comments. I would also like to thank Dr. Mahdi Tavakoli for his guidance and valuable inputs during my project.

I thank my fellows in the Mushahwar lab and the SMART Network Center: Javad Mehr, Don Wilson, Frank Gauthier, Michel Gauthier, Rod Gramlich, Soroush Mirkiani, Dr. Mojtaba Sharifi, Dr. Ashley Dalrymple, David A. Roszko, and Adam Parker. I really enjoyed working with them.

Last but not the least, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support. I am forever indebted to my mom and my sister for guiding me towards the opportunities and experiences that have made me who I am.

This thesis research was conducted with the financial support of the Canadian Institutes of Health Research, the Canadian Institute for Advanced Research, Natural Sciences and Engineering Research Council of Canada, Canada Foundation for Innovation, Alberta Ministry of Advanced Education, the grant of the Alberta Ministry of Jobs, Economy and Innovation to the Centre for Autonomous Systems in Strengthening Future Communities. Pouria Faridi was supported by the University of Alberta Master's Entrance Scholarship and the Alberta Graduate Excellence Scholarship.

Table of Contents

Abstract	ii
Preface	iv
Dedication	v
Acknowledgements	vi
List of Tables	x
List of Figures	xi
List of Algorithms	xiii
List of Abbreviations and Notations	xiv
Chapter 1: Introduction	1
1.1 Powered Lower-limb Exoskeletons – current research and challenges.....	2
1.1.1 Commercially Available Lower-limb Exoskeletons.....	3
1.1.1.1 The Hybrid Assistive Limb (Hal) Exoskeleton	3
1.1.1.2 The ReWalk Exoskeleton	4
1.1.1.3 The Robotic Exo H3 Exoskeleton.....	4
1.1.1.4 The EksoNR Exoskeleton	5
1.1.1.5 The REX Exoskeleton.....	5
1.1.1.6 The Indego Exoskeleton	5
1.1.2 Sensors and Actuators in Exoskeletons	7
1.1.3 Mechanical Limitations of Exoskeletons.....	8
1.1.4 Control Challenges of Exoskeletons.....	9
1.1.4.1 High-level Control	10
1.1.4.1.1 Brain-Computer Interfaces - still not reliable as a high-level controller	10
1.1.4.1.2 Movement Recognition – promising for people with detectable motor thresholds	10
1.1.4.1.3 Terrain Detections – have potential, but must be coupled with other strategies	12
1.1.4.1.4 Manual User Input: most common, have shown potential for improvements	12
1.2 Intelligent Learning Methods.....	15
1.2.1 Supervised Learning Methods	16
1.2.1.1 Regression Analysis.....	17

1.2.1.2 Support Vector Machines	18
1.2.1.3 Artificial Neural Networks	20
1.2.2 Reinforcement Learning Methods	23
1.2.2.1 RL Framework	23
1.2.2.2 Linear Function Approximation	25
1.2.2.2.1 Tile Coding	25
1.2.2.2.2 Selective Kanerva Coding.....	26
1.2.2.3 Temporal-Difference Learning	27
1.2.2.4 General Value Functions.....	28
1.3 Thesis Outline	32
Chapter 2: Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton	
Control: Preliminary Results.....	34
2.1 Introduction.....	34
2.2 Methods.....	36
2.2.1 Robotic Platform.....	36
2.2.2 Experimental Procedures	38
2.2.3 Machine Learning Strategy.....	39
2.3 Results.....	42
2.4 Conclusions and Discussion	44
Chapter 3: Continual Machine Learning and Unlearning in Lower-limb Exoskeletons to	
Modify Switching-based Interfaces	46
3.1 Introduction.....	46
3.2 Methods.....	50
3.2.1 System Configuration	50
3.2.2 Communication Methods.....	51
3.2.3 Experimental Procedures	51
3.2.4 Experimental Setup.....	52
3.2.5 Learning Strategy.....	55
3.2.6 Function Approximation Method	57
3.2.7 Statistical Analysis.....	59
3.3 Results.....	60

3.4 Discussion and Conclusions	68
Chapter 4: General Discussion	71
4.1 Summary and Significance	71
4.1.1 Thesis Summary and Significance.....	71
4.1.2 A Machine-learned System for Lower-limb Exoskeletons' Control – Offline Verification	72
4.1.3 A Machine-learned System for Lower-limb Exoskeletons' Control – Real-time Verification	73
4.2 Limitations	75
4.3 Future Directions	76
4.4 Links to Neuroscience.....	77
4.4.1 Dopamine and TD Error	77
4.4.2 Cerebellum and General Value Functions	80
Bibliography	82

List of Tables

Chapter 2

Table 2.1: Characteristics of walking modes	39
--	----

List of Figures

Chapter 1

Figure 1.1: Some types of the commercially available lower-limb exoskeletons	3
Figure 1.2: Baseline state control strategy of the Indego exoskeleton	7
Figure 1.3: An example of an exoskeleton (TWIICE) utilizing a switch-button panel	13
Figure 1.4: An example of the SVM decision process	18
Figure 1.5: A comparison between connections in the CNS and artificial neural networks	21
Figure 1.6: A general RL framework	23
Figure 1.7: A general overview of two feature vector representation methods.....	26

Chapter 2

Figure 2.1: A study investigator wearing the exoskeleton with a walker equipped with sensors.....	37
Figure 2.2: Experimental design scenarios.....	40
Figure 2.3: Signals used in the state-space of the system.....	41
Figure 2.4: Number of switches required under the adaptive and the best possible non-adaptive strategy.....	43
Figure 2.5: Normalized GVF prediction values for a sample of recorded data.....	44

Chapter 3

Figure 3.1: An experimenter (user) wearing the exoskeleton with a walker equipped with a switch button panel, sensors and a reflective marker.....	51
Figure 3.2: Three experimental design scenarios.....	54
Figure 3.3: Vicon and LiDAR prototype distributions for a time step.....	59
Figure 3.4: An example of GVF predictions during the Static-Learning and unlearning scenario.....	61
Figure 3.5: Weight values for the second user experiments.....	62
Figure 3.6: A comparison between the prediction values, their true return and the raw activity levels.....	64
Figure 3.7: Number of required switching actions and errors at each switching instance.....	66
Figure 3.8: Statistical comparison between the average number of required switches under the best possible fixed list and the adaptive list.....	67

Figure 3.9: Statistical comparison between the total number of required switches under the best possible fixed list and the adaptive list..... 68

Chapter 4

Figure 4.1: A comparison between the traditional non-adaptive switching method and the adaptive switching method 72

Figure 4.2: GVF values for 2 modes in a post-hoc test..... 79

List of Algorithms

Chapter 2

Algorithm 2.1: GVF prediction and learning with TD(λ)..... 42

Chapter 3

Algorithm 3.1: Adaptive switching algorithm.....57

Algorithm 3.2: 2D+3D Selective Kanerva Coding..... 59

List of Abbreviations and Notations

General value function – GVF

Central nervous system – CNS

Spinal cord injury – SCI

Degrees of freedom – DOF

Food and Drug Administration – FDA

Electromyography – EMG

Electroencephalography – EEG

Inertial measurement unit – IMU

Artificial intelligence – AI

σ – Softmax function

Kernel ridge regression – KRR

Support vector machine – SVM

Convolutional neural network – CNN

ImageNet Large Scale Visual Recognition Challenge – ILSVRC

Gait neural network – GNN

Reinforcement learning - RL

Selective Kanerva coding - SKC

Temporal-difference - TD

Ventral Tegmental Area – VTA

S, s - Set of states, state

A, a - Set of actions, action

R - Set of rewards

G – Return

v- Value function

e - Eligibility trace

w - Weight vector

x - Feature vector

α - Step size

δ - Temporal difference learning error

λ – Trace decay parameter

γ - Discount factor

T – Time step

Light Detection and Ranging – LiDAR

Central pattern generator - CPG

Universal Datagram Protocol – UDP

Bluetooth Low Energy –BLE

u_i – user i

B – Blue

G – Green

Y – Yellow

R – Red

C – Cumulant

p_j – GVF predictions

k – Number of prototypes

m - Parameter for closest prototypes

Chapter 1: Introduction

This thesis work is the first demonstration of the utility of a predictive approach, called general value functions (GVFs), for generating a smart system for lower-limb exoskeletons. As part of this thesis, the system is shown to be capable of adapting to the intention of lower-limb exoskeleton users in real-time and is inspired by the cerebellar neural circuitry. This adaptation to the users' intention mimics the adaptation of a Purkinje cell to signals sent by mossy and climbing fibers in the cerebellum and its error-correction behavior, which is the basis of motor coordination. Using this biologically-inspired method for controlling an exoskeleton is a step towards incorporating biologically-inspired mechanisms for generating advanced human-like intelligence in robots.

The brain and spinal cord, forming the central nervous system (CNS), are responsible for receiving sensory information, processing those, and issuing motor signals. In case of any damage to either of these, however, an important function may be lost. Being able to walk is among the functions that can be significantly affected during impairments to the CNS. Spinal cord injury (SCI), stroke, multiple sclerosis, Parkinson's disease, etc. are among those neurological conditions that can weaken or completely disrupt the ability to walk.

Regeneration of the damaged tissues in the CNS may be the eventual solution for recovery from those neurological impairments. However, ongoing regeneration studies [1], [2] have failed, to date, to improve function after CNS injuries in humans [3]. This has led research studies to search for alternate approaches. Some approaches are designed and tested specifically for restoring walking function after damage to either the brain [4], [5] or the spinal cord [6]–[8], while other approaches are targeting overall walking recovery regardless of the impaired area,

mostly through wearable technologies [9]. Substantial functional improvements after injuries have been attained through innovative rehabilitation and assistive devices [10]. One domain for restoration of walking after injury which still has a lot of room for improvement, is using assistive robotic systems; namely, powered lower-limb exoskeletons [11].

1.1 Powered Lower-limb Exoskeletons – current research and challenges

Powered lower-limb exoskeletons are wearable robotic devices that provide mobility assistance and have different active joints that can be controlled either automatically or by their users, depending on the implemented control strategy for the device. What makes these exoskeletons beneficial for rehabilitation and gait restoration research is their capability to track the desired motions presented to them with high accuracy, their application in both the community and the home environment, collecting data from different joint motions with their built-in sensors that can be used as feedback, and providing different levels of assistance to users with a variety of conditions such as people with complete SCI (no motor function) and incomplete SCI (limited motor function) [12]. Many powered lower-limb exoskeletons such as Indego [13], Lokomat [14], ReWalk [15], and Exo-H3 [16], are developed and currently being used for rehabilitation research with one or more active joints. Figure 1.1 presents some examples of these devices, currently used in clinical domains. The available exoskeletons, despite their noticeable capabilities in assisting their users still have some limitations that are preventing them from being easily and widely used in society. There are both hardware (mechanical design) and software (controller) limitations that will be discussed later in this introduction.

1.1.1 Commercially Available Lower-limb Exoskeletons

Different types of lower-limb exoskeletons are currently being used in clinical settings and research laboratories to investigate their capabilities for rehabilitation and mobility assistance. Researchers are aiming to enhance both the structural arrangements and the underlying control strategies of these available exoskeletons to make them suitable for daily usage. Many companies and their associated research laboratories are manufacturing lower-limb exoskeletons with various capabilities. Some of the widely used devices in the clinical and research settings are summarized below:

1.1.1.1 The Hybrid Assistive Limb (Hal) Exoskeleton

The hybrid assistive limb (Hal) Exoskeleton [17] is developed by researchers from Tsukuba University and Cyberdyne robotics company (Japan) to support and enhance motor function (Fig 1.1 a) and is the first lower-limb exoskeleton used in clinical studies in 2009 [18], [19]. This exoskeleton is designed in two versions; two legs and one leg only with active hip and knee joints in the sagittal plane. It is purposed to be a hybrid exoskeleton in its underlying control strategy since it is capable of detecting the users' bioelectrical signals of the lower limb (if

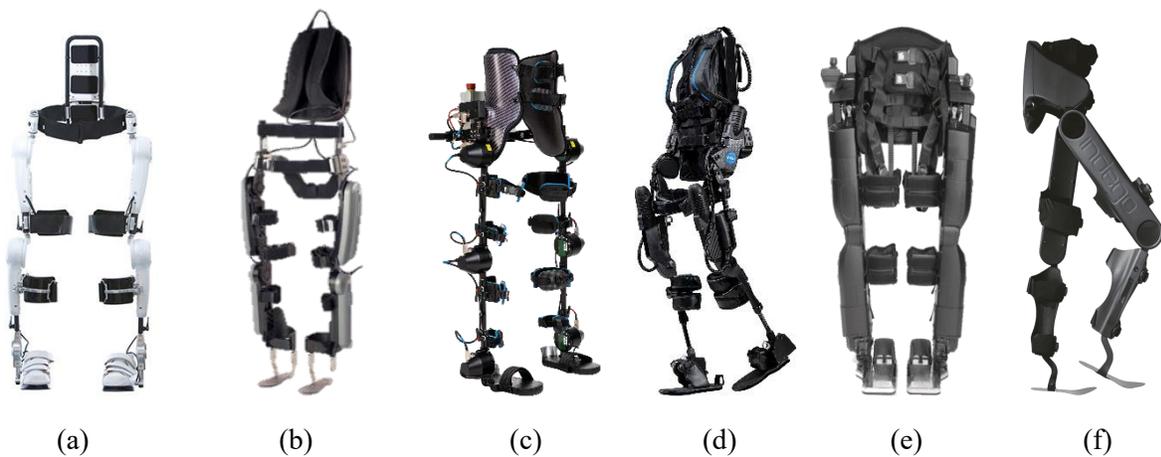


Figure 1.1 Some types of the commercially available lower-limb exoskeletons. a) Hybrid assistive limb (Hal) exoskeleton b) ReWalk exoskeleton c) Robotic Exo H3 exoskeleton d) EksoNR exoskeleton e) REX exoskeleton f) Indego exoskeleton.

detectable) and incorporating that into its motion planning system. It also has free joints (not controlled) for the hip and ankle with 2 degrees of freedom (DOF) in the frontal and horizontal planes. The main purpose of this exoskeleton is to support or provide assistance for sitting, standing up, and walking. Moreover, it is equipped with a remote monitoring system that sends the joints' signals as visual feedback to the operator for updating and adjusting the design properties, if needed. Therefore, this device can be a great candidate for use in remote areas as well [17].

1.1.1.2 The ReWalk Exoskeleton

The ReWalk exoskeleton (Argo Medical Technologies Ltd., Israel) is the first lower-limb exoskeleton (Fig 1.1 b) that received Food and Drug Administration (FDA) clearance for use in rehabilitation and for personal use at home in the United States. Similar to the Hal, the motors are designed to control the hip and knee joints in the sagittal plane, while the ankle joints have a mechanical structure, with spring-assisted dorsiflexion [15]. The recommended walking speed for this device is 0.6 m/s. Its underlying control strategy uses the joint angles as feedback to trigger movements. It also arguably, has stair climbing capabilities [20].

1.1.1.3 The Robotic Exo H3 Exoskeleton

The Exo-H3n (Technaid S.L., Madrid, Spain), utilizes brushless DC motors to actuate 3 DOFs of each leg (hip, knee, ankle) in the sagittal plane (Fig 1.1 c). It is equipped with security considerations such as a power shut-down button and mechanical stops if needed. The built-in sensors can provide the joint angles in real-time and pressure sensors can provide the center of pressure information for each foot [16]. This device has been primarily designed for research purposes and allows for the integration of custom control strategies into its onboard computers.

The basic device movements (such as gait speeds, sitting, and standing up) can be also controlled using an Android app.

1.1.1.4 The EksoNR Exoskeleton

The EksoNR (Ekso Bionics, Richmond, CA, USA) (Fig 1.1 d) is considered the first lower-limb exoskeleton with FDA approval to be used for people with stroke. It also has FDA approval for people with SCI at levels C7 to L5. It is equipped with hip and knee active joints and a passive dorsiflexion ankle joint. The captured joints' data are stored in a cloud-based dashboard for faster and better access. The software includes a variety of control programs to help with the users' balance, squat and in-place stepping (eksobionics.com).

1.1.1.5 The REX Exoskeleton

The REX exoskeleton (Rex Bionics, Auckland, New Zealand) is recognized as the first (and only by the time of writing this thesis) commercially available exoskeleton with self-supporting characteristics, which makes it independent of walkers or crutches. However, this characteristic makes it heavier than the other exoskeletons and reduces the overall walking speed capabilities. It is capable of executing forward, backward, sideways and turning movements and also can be considered for rehabilitation exercises such as squats, leg swings, sit-to-stand, and stretches (www.rexbionics.com).

1.1.1.6 The Indego Exoskeleton

The Indego exoskeleton (Parker Hannifin Corporation, Cleveland, Oh, USA) is the device used in this thesis work where the hip and knee joints are powered by brushless DC motors (Fig 1.1 f) and is considered the second wearable lower-limb exoskeleton (after Hal) used for clinical evaluation in 2011 [18] under the name, Vanderbilt exoskeleton, at that time [21]. The motors of this device can provide up to 12 Nm of continuous and 40 Nm short-duration (less than 2 sec)

torque [13]. In case of a power failure, there are brakes located on the knee joints to prevent knee buckling. The device is powered by a lithium battery, located in the hip piece. The distributed embedded system of the device is responsible for data processing, feeding the electronics, power management, and controlling the communication between different boards and interfaces. The device weight is $\sim 13\text{kg}$, considered the lightest among previously introduced exoskeletons, and should be accompanied by a walker or crutches when walking. It is considered for use by people with spinal injury at levels of T3 to L5 in both community and home environments. The desired control strategy for controlling its hip and knee joints can be designed in the Real-time Desktop Simulink environment of MATLAB (The MathWorks, Inc., Natick, MA, USA) which then communicates with the exoskeleton system through the Vector VN1610 CAN interface. The device can also be controlled with its baseline controller via Bluetooth connection through an iOS pad that is preloaded on an iPod. The iOS app can be utilized for tracking the joint angles and changing some basic settings and is useful for clinical settings. The baseline controller uses a state control strategy, starting from sit-to-stand movement and cycling through alternating the left and right legs. Figure 1.2 shows the baseline control strategy of the Indego exoskeleton and the possible transitions between states in the Real-time Desktop Simulink environment. Each of these states are initiated based on the changes in the users' hip tilt angle and the distance between the center of mass projection and the forward ankle joint onto the ground plane [13]. This makes the executed movements slow and robotic (not continuous) in appearance.

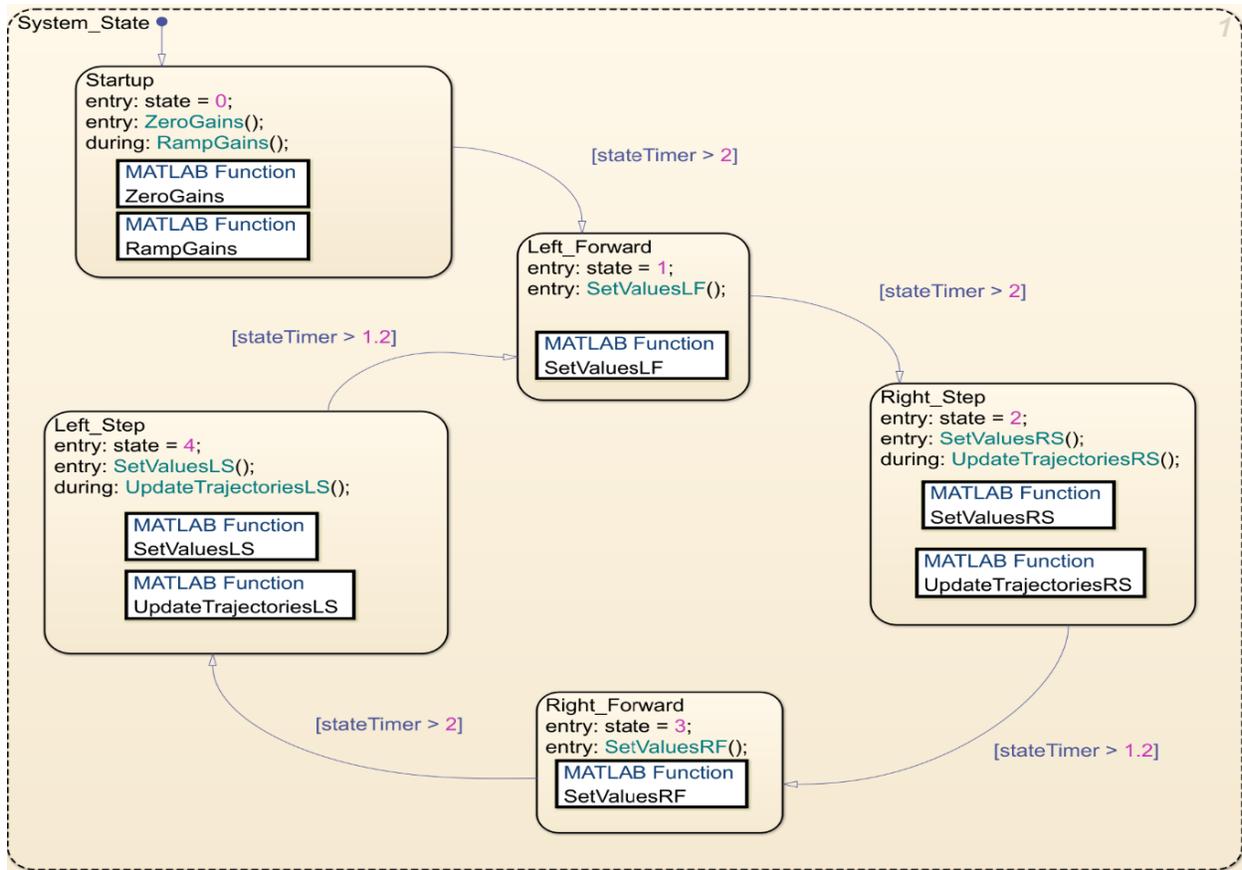


Figure 1.2 Baseline state control strategy of the Indego exoskeleton in the Simulink environment. The startup state performs the sit-to-stand movement and then cycles through the left-right alternations. A stateTimer also has been placed to automatically perform the next action in an open-loop setup in case the desired pressure or tilt signal is not detected before the allocated period.

1.1.2 Sensors and Actuators in Exoskeletons

Different sensors and actuators (motors) can be implemented in the exoskeletons to increase their capabilities. Electromechanical motors are responsible for motor actuation and providing the desired movements of the limbs. In some cases, and depending on the level of the injury, electrical muscle stimulation can also be used for augmenting the generation of the desired walking patterns. Exoskeletons are also equipped with sensors. Kinematic and kinetic sensors such as pressure, torque, force, and tilt sensors, gyroscopes, accelerometers, infrared distance sensors and ultrasonic devices are all examples of sensors that can be incorporated into these devices. These sensors have the capability of recording the relevant signals (such as joint angles,

and ground reaction forces) to be analyzed later or for tracking changes in walking over time. They also can provide feedback to the control system in real-time; therefore, increasing the awareness of the devices of their own movements and also the environment. In addition to the sensors equipped on the device, exoskeletons can also benefit from body-worn sensors such as electromyography (EMG) and electroencephalography (EEG) signals. These signals also can be used either in real-time or offline to assist with the control. They can also be analyzed later to modify the rehabilitation pattern or the training routine of the users. Another application of this recorded signals is as a training dataset for designing intelligent controllers.

1.1.3 Mechanical Limitations of Exoskeletons

Currently available lower-limb exoskeletons are facing major concerns in their mechanical design. In Rodríguez-Fernández et al. [18], over 25 lower-limb exoskeletons (6 had FDA approval) were analyzed to detect the structural and mechanical limitations. The first structural limitation seen in the lower-limb exoskeletons is the number of DOF of joints. Except for the REX exoskeleton [22] which allows movements in all planes, other reviewed devices only allow movements in the sagittal plane. This can be problematic especially when performing turning movements that require flexibility in other planes as well. However, increasing the number of active joints will require an additional actuation system which will lead to an increase in both weight and cost of the device. According to this systematic review [18], 76% of the reviewed devices incorporate 2 active joints (hip and knee) with the ankle joint either fixed (such as Indego) or passive. Reducing the number of active joints and as a result, the DOF of movements, will make the devices less desirable and reduces the ease with which daily activities can be performed. Another structural challenge can be seen in the actuator types that are currently used. Of the investigated devices in Rodríguez-Fernández et al. [18], 88% had electric motors and the

remaining 12% either used hydraulic or pneumatic actuation. Although electric motors are capable of providing immediate feedback and can be programmed easily with accurate control positioning, they are expensive. They are not suitable for hazardous areas (unlike pneumatic ones) and can overheat if used continuously. Each electric motor has a specific speed, thrust and force limits and if these limits need to be exceeded, the motor needs to be changed [23]. Power supply can be seen as another major limitation. These devices are only capable of providing 2-4 hour of continuous supply and as a result, this can be problematic when a charging station is not available [18].

1.1.4 Control Challenges of Exoskeletons

In addition to the mechanical design limitations mentioned in the previous section, controlling the movements of the available joints and being responsive to users' commands is another equally important challenge in exoskeletons. The ultimate goal of many research avenues in the control domain is to take users' intention into account while reducing the effort needed to perform tasks, and eventually making the orthosis adaptive to the users' need in a safe manner. To this end, many control strategies have been designed and tested experimentally. Generally, there are three main areas of focus for designing control strategies for exoskeletons: high-level, mid-level and low-level control [10], [24]. In simple terms, high-level controllers determine the walking mode, while mid-level controllers shape the desired joint trajectories or torques [25]–[27]. Low-level controllers aim to track the desired joint trajectories or joint torques, and are known as either position-controllers [28], [29] or torque/force-controllers [30], [31]. The main focus of the work in this MSc thesis is on high-level control and as a result, the remainder of this section is allocated to the high-level control concept.

1.1.4.1 *High-level Control*

A high level controller can be seen as a perception and motion planning layer [32] that characterizes the overall status or behavior of the robotic device. Both knowledge of the environment and user-dependent measurements (such as ground reaction forces, joint torques, joint angles, brain-activity signals, etc.) can be used as the inputs to the high level controller, while the output is expected to be a specific mode of walking [10]. The modes available on a control system are predefined modes, ideally designed based on the needs of the users. They can consist of straight walking with different speeds and step lengths [20], turning left/right, and various tasks such as stair ascending, stair descending [20], [33], [34], sit-to-stand and stand-to-sit transitions [20], [35], [36]. In this regard, selecting the user's intended next mode has been identified as the major concern of high-level controllers, especially when a variety of modes are present [10]. In a recent review [10], high-level controllers were divided into 4 main categories as: brain-computer interfaces, movement recognitions, terrain detections and manual user inputs.

1.1.4.1.1 *Brain-Computer Interfaces - still not reliable as a high-level controller*

This category of high-level controllers, which mostly uses electroencephalography (EEG) recordings from the brain [37]–[40], faces many practical issues such as artifact removal, requirement of high concentration of the user which prevents them from performing other cognitive activities at the same time, the lengthy procedure of preparing the EEG electrodes, and losing recording accuracy over time [41], [42]. Therefore, using brain activity signals cannot be considered as a practical method in the near future.

1.1.4.1.2 *Movement Recognition – promising for people with detectable motor thresholds*

Another group of strategies (movement recognitions) aim to determine the desired mode of walking by relying on the user's body movements and forces, mostly originating from the lower

body. In one study [43], a state controller was implemented for walking in 4 different ground inclination scenarios. By crossing predetermined thresholds on the ankle angles of an ankle-foot orthosis, the different scenarios were activated. In another study [44], a nonlinear network was trained to estimate the passive torques. Based on the estimated passive torques and total amount of measured joint torques, the human-robot interaction torque was calculated and passed through a threshold-crossing algorithm to detect user intention of increasing or decreasing speed and step-length. Despite the simplicity of these threshold-crossing techniques, some practical issues still exist. All of the tests using these strategies have to date been performed in neurologically-intact users in laboratory settings, and it is unclear if they are applicable to people with SCI. Specifically, it is uncertain that persons with SCI can apply the desired amount of detectable torques to cross the pre-defined thresholds. Therefore, each user with SCI would need to go through preliminary assessments in which a certain level of threshold is assigned for their case; however, this level may not be appropriate when the user's ability to generate torque is affected by fatigue. Furthermore, these methods are limited to people with motor incomplete SCI (ASIA C and D [45]) and cannot be used for people with motor complete SCI who would be unable to generate any torque. Instead of threshold-crossing methods, lower-limb signals were also used for training machine learning algorithms, mostly supervised learning, to classify different sensor values for a specific mode. (The common machine learning models utilized nowadays are described in Sec 1.2). In Villa-Parra et al. [46], surface EMG signals acquired from the trunk and lower-limbs were used to train a classifier that distinguished between different states of over-ground walking. In Chinimilli et al. [47], thigh angles acquired from inertial measurement units (IMUs) were used in a machine-learned supervised algorithm to provide hip assistive torque during different gait activities. These learning systems require a high amount of recorded data for

their training sets due to the high variability among different users, and face major challenges in becoming fully generalizable [48]. In addition to the lower body movements, trunk and upper body movements were also investigated for mode detection. In Farkhatdinov et al. [49] head and trunk angular positions acquired from neurologically-intact participants were used to detect users' intention for turning left and right. As another example, the basic control strategy in some exoskeletons such as the Indego exoskeleton [50], requires leaning forward (changing the center of pressure) to perform sit-to-stand or stand-to-walk movements. Therefore, although using upper body movements can make the control decisions more reliable in people with injuries that have affected the function of their lower limbs, it has limited capabilities in real-world as it forces the users to perform specific and often unnatural upper body movements.

1.1.4.1.3 Terrain Detections – have potential, but must be coupled with other strategies

The terrain detection category uses different sensors or cameras such as IMUs [36], [51], infrared distance sensors [51], [52], head-mounted cameras [53], chest-mounted RGB camera systems [54], and ultrasonic devices on the waist [55] to recognize either the distance between the user and obstacles or different terrain types such as stairs or uneven surfaces. This recently emerging category has shown promising results [52], [53]. Nonetheless, if implemented independently of other high level control categories, its usage is limited to modes that are only environment-dependent. This technique would not be able to provide high-level control knowledge in the absence of any objects or obstacles, detectable by these sensors.

1.1.4.1.4 Manual User Input: most common, have shown potential for improvements

While previous methods are still in the research phase, controlling the device through user input is being introduced as the basic underlying control method of the commercially available exoskeletons. This method works directly through user commands, either with switch buttons or

voice-control [20], [21], [41]. The push-button approach is less susceptible to errors than the speech-recognition technique because it is not affected by environmental noises and is the most currently implemented high-level approach for the end users [41]. It is simple to implement, and has the capacity for adding more modes. Figure 1.3 shows an example of a lower-limb exoskeleton equipped with a switch-button panel for performing different walking movements.

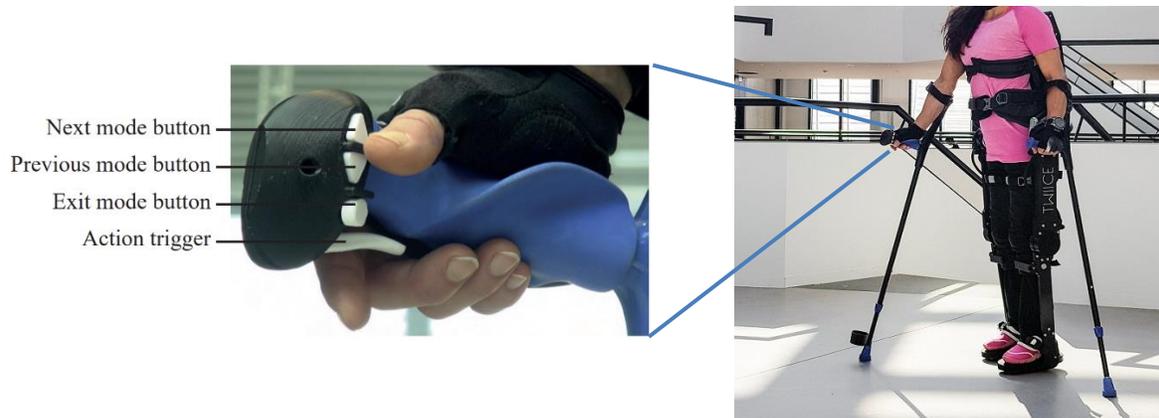


Figure 1.3 An example of an exoskeleton (TWIICE) utilizing a switch-button panel for switching between several operating modes, as: slow gait, fast gait, sitting down, stairs ascent, stairs descent and variable step lengths [18].

Manual User Inputs Drawbacks – what is the evidence?

Despite the benefits of using push-button switching systems for lower-limb exoskeletons, they have several drawbacks. Using a single switch and going through a list of operating modes requires both high transition times to toggle between modes and a high number of switching actions (hitting a switch button to get the desired mode) for a single switch system. Using a panel of switches (one for each mode) also limits the capacity of adding more modes. Therefore, these push-button strategies make the user feel uncomfortable, decrease the speed of performing tasks, and require a high degree of mental concentration, thus increasing the chance of errors [41].

A recent (2019) qualitative study involving prosthesis users and therapists in Europe [56], described the participants' opinions on using a prosthesis with several available operating modes

under two control approaches: 1) manual switching from a pre-defined list of modes, and 2) a pattern recognition approach using EMG signals from the residual limb. The study concluded that almost all of the participants found the manual switching approach (first method) as a major problem, describing it as “too time-consuming”, “taking lots of effort” and “mentally and physically exhausting,” leading to abandoning the device [56]. Here are some direct quotes from the participants, indicating their feelings after using the manual switching approach, according to Franzke et al. [56]. The participants’ id is mentioned after each quote as P_i and therapists’ id as T_i.

“The control. . . it always takes effort... And if it takes me a lot of effort to do a certain task...Then I do it once, then I do it twice, but I won’t do it a third time (P₂). It is more that you sometimes think ... this switching to another grip function ... That you have to use a trigger and then you continue with the next step ... Then it takes you a bit longer (P₁₂). I don’t get that under control. It’s just very difficult (P₁₁). Well, you just take the path of least resistance. And I won’t slow down my movements just to utilize my prosthesis (P₁₆). Actually it should do it faster... it has to switch faster (T₃).

The pattern recognition method (second approach) in the mentioned study is not applicable to lower-limb exoskeleton users who are unable to exert EMG activity in their lower limb muscles, including persons with SCI or stroke. Although some improvements over current switching systems were identified, the study participants were still not fully satisfied with this approach. Specifically, the system was unable to predict the correct movements in cases where the signals changed due to noise or unexpected situations, such as when a participant was carrying a bag of groceries shopping [56]. Other limitations of pattern recognition approaches are similar to those discussed the Movement Recognition section (Sec 1.1.4.1.2).

The same issues of manual switching are present when using lower-limb exoskeletons. A newly published qualitative study from the Netherlands interviewed 13 people with SCI with exoskeleton experience for their needs and wishes for the future lower-limb exoskeleton [57]. The most important need was making the exoskeleton systems easier to use and work with. The study reported a consensus to incorporate different modes such as step and/or speed adjustment controls to enhance the exoskeletons' utility. There were uncertainties on how to implement the controller though, suggesting control interfaces to be put on crutches, with functionality similar to the automatic gear transition mechanisms in cars [57].

Overall, what is evident is a need for a more intelligent approach for controlling the exoskeletons and adapting them to the users' need. A system that is intelligent enough to predict the users' next movement, while also not being dependent on signals from their limbs would be highly desirable.

Studies on enhancing control methods using a switch-button panel where switching is performed based on a pre-ordered list of available modes have been pursued in other areas such as upper-limb prostheses [58]–[61]. In those studies and also in this thesis, intelligent learning methods were utilized as the underlying control strategies of the devices. As a result, the following section is allocated to describing the most common learning methods that are used for controlling robotic devices.

1.2 Intelligent Learning Methods

The concept of artificial intelligence (AI) was introduced in 1950s and soon after, machine learning methods were described in 1980s. Since then, researchers, scientists, entrepreneurs and many others have begun to incorporate ideas from these two concepts to benefit the society, improve the quality of life, automate tasks and help people who are in need of help. AI enables

machines to think like humans and machine learning, which is a part of AI, was introduced as the science of getting computers to learn without being explicitly programmed [62]. Machine learning methods can be generally divided into three broad categories of unsupervised, supervised and reinforcement learning. In this section, the two latter fields currently used for enhancing the control of exoskeletons are described in detail. The unsupervised learning methods are usually utilized for clustering and/or reducing the dimensionality of data and are out of scope of this introduction.

1.2.1 Supervised Learning Methods

Supervised learning algorithms are the most widely used methods in machine learning, where pre-labelled data are incorporated to classify different groups or to predict a future numerical value. Inputs to the algorithms that are used as examples to train the model are known as the training dataset. They are used to optimize the values of initialized weights and biases of the learning system (if available) that drive the predictions. There is one output for each input, with each input being a class or a numerical value. The biological equivalent is concept learning, where past experiences are used to create associations, extract commonalities or to find distinctions and adapt that learned knowledge to respond to a novel stimulus or to classify objects [63], [64]. The idea of concept learning has been the topic of interest among many researchers exploring prefrontal cortex, parietal cortex and hippocampal activation patterns [63], [65]–[67]. Usually, a sufficient amount of training examples (training datasets) are needed to train the system to enable it to distinguish between different classes or different values whenever new examples are introduced, known as the test dataset.

Supervised learning methods can be divided into many different algorithms and some of the most applicable ones to locomotion generation and exoskeleton control are introduced here.

1.2.1.1 Regression Analysis

One of the most fundamental methods in supervised learning is regression analysis. Driven from statistical modeling, regression analysis uses statistical concepts to drive relationships between one or more independent variables (known as features, X_i) and one dependent variable (the outcome, y_i) [68]. In the simplest model, linear regression aims to fit a line (in 2 dimensional space) or a plane (in higher dimensional space) to the available dataset, using the following equation:

$$y_i = W_i X_i + b_i \quad (1.1)$$

Where y is the outcome and X is the vector of features. W and b are the weight vector and bias terms, respectively, which are optimized during the learning process using the gradient descent algorithm, an optimization algorithm that aims to find local minima of a goal function [69]. In case of a classification problem, the output (y_i) is sent to a softmax function. This is a function that calculates the probability of the input belonging to a class. The classification method is known as logistic regression and its softmax algorithm is provided in the following equation:

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{j=1}^k e^{y_j}} \quad (1.2)$$

where the output $\sigma(y)_i$ is a probability of belonging to the i th class among all k classes, calculated as a positive integer between 0 and 1.

Linear and logistic regression algorithms are widely used in prediction or classification of biological signals and/or different sensor values associated with exoskeletons. In Li et al. [70] a regression model was trained using ankle angle and contact forces at the toe and heel to detect gait events during walking with a powered ankle-foot orthosis. Their approach showed ~40%

enhancement in the accuracy of event detection compared to a threshold crossing algorithm. In Hahne et al. [71] different variations of regression methods (linear regression, mixture of linear experts and kernel ridge regression (KRR)) were applied on EMG signals recorded from both neurologically-intact people and a person with congenital upper limb deficiency. Their results showed that the KRR method outperformed other methods in its control accuracy, while the mixture of linear experts method, which is a physiologically inspired extension of linear regression, also showed great potential for further investigations as it required lower computational cost and hardware during calibration and prediction phases.

1.2.1.2 Support Vector Machines

Support vector machines (SVMs) is a very powerful algorithm for classification that not only aims to classify the dataset, but also aims to find the best possible decision boundary. The best possible boundary is the boundary that maintains the largest distance (margin) between the separation line and the points in each class, in proximity to that boundary (Fig. 1.4). This will

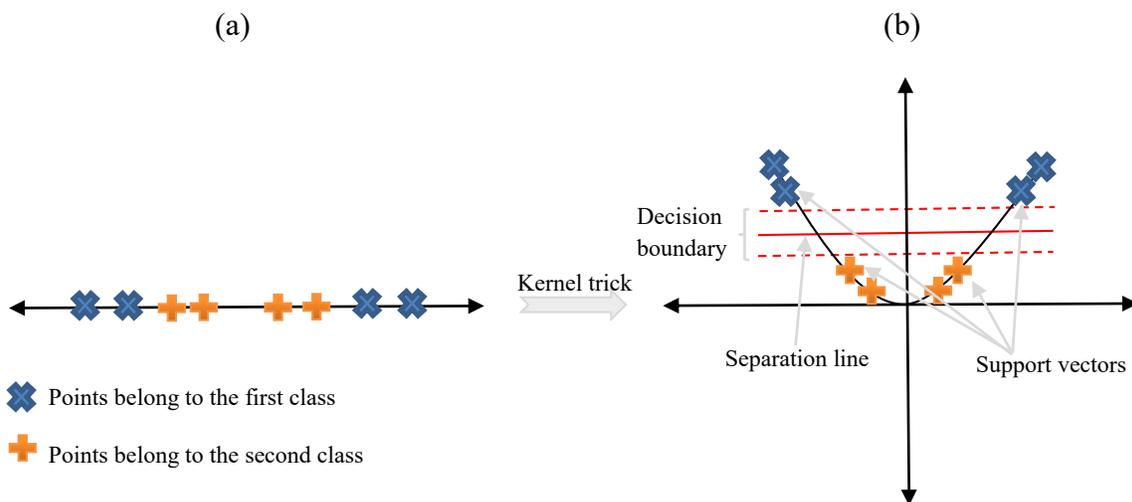


Figure 1.4 An example of the SVM decision process and its kernel trick. (a) shows a non-linear classification problem between two classes in one dimension. (b) shows the translated non-linear problem into a linear one in the 2 dimensional space using the kernel trick. The boundary points are called support vectors as they are fundamental to the decision boundary establishment. The goal is to provide the largest distance between the support vectors and the separation line (solid red line), using the boundary lines (red, dashed lines).

make the algorithm faster in classification than other algorithms when there is a clear margin of separation between points of different classes. It also uses only a subset of the training dataset that is in proximity to the boundary line, called support vectors (Fig. 1.4), which increases the efficiency in memory allocation. Another important advantage of SVM is known as its kernel tricks. Kernel is a function in SVM that translates a non-linear problem to a linear problem in one higher dimension; therefore, avoiding complexities of non-linear classifications [72]. An illustration of the SVM algorithm and its kernel trick is shown in Fig. 1.4.

SVMs have also been used widely for high-level control of robotic devices and specifically, lower-limb exoskeletons. In Xu et al. [73], an SVM was trained using IMU measurements from the knee joint of a unilateral knee assistive exoskeleton for real-time detection of gait events. IMU recordings from a bilateral HeSA (Hip Exoskeleton for Superior Augmentation) [47] were also incorporated for high-level control, by training an SVM to detect transitions between activities in real-time. Huang et al. [74] showed that an SVM can be trained with EMG recordings from gluteal and residual thigh muscles and also ground reaction forces as inputs. Features from EMG signals (mean absolute value, number of slope sign changes, waveform length, and number of zero crossings) were extracted to be passed to the classifier along with the ground reaction forces to classify gait states during stance (initial double-stance, single-limb stance, terminal double stance) and swing phases for prosthetic leg users with transfemoral amputation. Their trained SVM showed a 99% accuracy for the stance states and 95% accuracy for the swing state and predicted the correct transitions, 300-650 ms before the prosthetic foot left or touched the ground.

Although the mentioned supervised learning algorithms showed high accuracy (above 90%) for classification of the provided signals, the need for manual feature extraction/selection and their

lack of capturing complicated patterns and associations led research investigations towards more advanced algorithms.

1.2.1.3 Artificial Neural Networks

The manual process of feature extraction and/or feature selection is prone to losing useful information that is not captured by the selected features. Therefore, artificial neural networks were introduced to automate this process by leveraging different layers of connections. They were inspired by the human brain neuronal connectivity, where a neuron receives inputs from neighbour neurons through its dendrites and sums all of the received inputs. If the total sum is more than a certain threshold, that neuron fires a signal that is then sent to neighboring neurons through its axon and associated collaterals. The same procedure is seen in neural network model configurations, in which each node (neuron) in the network receives a number of inputs, adds them together and sends the sum to an activation function that decides the level of information to be sent for further processing. A comparison between the neural networks' system and its biological motivation can be seen in Fig 1.5.

There can be many layers and connections of neurons in an artificial neural network, each requiring the magnitude of its weight (w_i) to be optimized with the goal of making the output class of the network similar to the desired label, or the output numeric value closest to the desired value. The process of using the final loss function (comparison between output of the network and desired output) and moving backwards into the network to update the weights using gradient descent is called backpropagation [75], that uses the chain rule of derivations to optimize the weights of each layer with the goal of minimizing the final error. If the number of layers in a network exceeds three, the network becomes a deep neural network and the learning process is called deep learning. If all neurons from previous layers are connected to all neurons

in their next layers, the network is called a fully-connected network. Deep learning models began to be used widely in many domains after the introduction of convolutional neural networks (CNNs) in 2012, which were developed in an attempt to achieve high image classification accuracy in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [76]. CNNs are a specific type of deep networks specialized in pattern detection, and have attracted the attention of many model developers and engineers in different fields. CNNs utilize filters as opposed to weight vectors, where each filter is focused on detecting a specific pattern from the input data, and can be optimized in similar approach to the weight optimization process. Due to the presence of filters, each neuron receives inputs only from its own filter which noticeably reduces the required learned parameters, when compared to a fully-connected network.

Various versions of deep neural networks are widely used now in research with task-specific modifications to solve the problems of controlling robotic limbs or exoskeletons. Ren et al. [77]

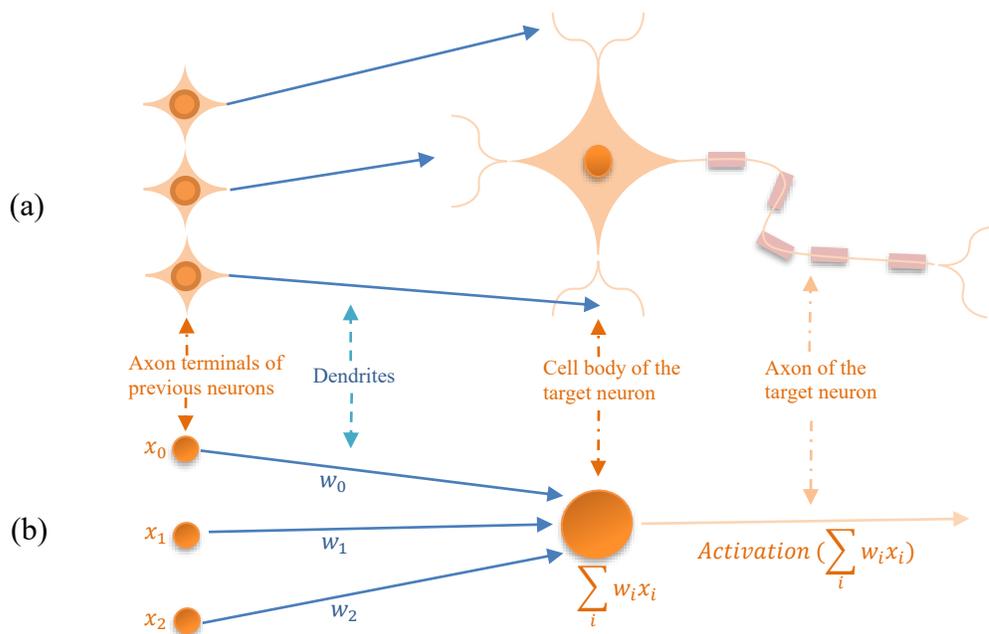


Figure 1.5 A comparison between neuronal connections in the central nervous system (a) and in the artificial neural networks (b). In both scenarios, inputs from previous neurons will be sent to the target neuron through its dendrites. The inputs are summed together in the cell body and sent through an activation unit, which determines the firing rate and the information to be sent to the next neuron.

trained a deep network using surface EMG signals from arm muscles and joint angles to predict users' motion and control an upper-limb exoskeleton. Their analysis showed that their designed deep-learned model performs better than traditional regression models in terms of the accuracy of predicting arm motion. In Laschowski et al. [78] a large dataset of images collected using their wearable camera during both indoor and outdoor walking scenarios was utilized to train a CNN. An environmental classification algorithm was designed and used for walking mode recognition with robotic prosthetic legs. Fang et al. [79] proposed a gait neural network (GNN) structure to optimize human-exoskeleton interactions. Their GNN consisted of an intermediate network, a target network, and a recognition and prediction model. The intermediate network was a sequence-learning algorithm called temporal convolutional network (proposed by Bai et al [80]) that used inertial sensor data to predict the intermediate sensor data. The output of the intermediate network was then passed to the target network that concatenated that output with the original input (to the intermediate network) and produced an encoded vector that contained both current and historical information. This vector was then passed to the recognition and prediction model (fully-connected networks) to predict three walking modes (standing, walking and running).

Although supervised learning methods are capable of generating precise predictions (high classification/regression accuracy), they need to be trained on previously collected and labeled data. Moreover, for people with complete SCI, EMG signals of limbs may not be detectable and cannot be utilized to train a model to predict the users' intent. As a result, there is a need for a method capable of online learning, with the ability to continually update the learned predictions and adapt to changes to the human-robot interactions without the need to be trained with a pre-labeled dataset.

1.2.2 Reinforcement Learning Methods

Reinforcement learning (RL) is the third branch of machine learning algorithms, with the first two branches being supervised and unsupervised learning methods. In RL, the goal is to maximize a cumulative future reward that an agent can receive by adapting its behaviour in its interactions with an environment. Many RL approaches, its computational perspectives and the terminology used in RL are similar to psychological concepts, namely theories of how different animals learn in laboratory environments [81]. These analogies will be discussed in the remainder of this introduction.

1.2.2.1 RL Framework

RL is a problem formulation for sequential decision making under uncertainty, where an agent (learner) is interacting with an environment (either virtual or physical), performing some actions at different states based on its policy and receiving rewards accordingly. Figure 1.6 depicts the fundamental elements of an RL framework and their connections with each other [81]. An agent can be a computer or a human experimenter, interacting in a virtual or physical environment,

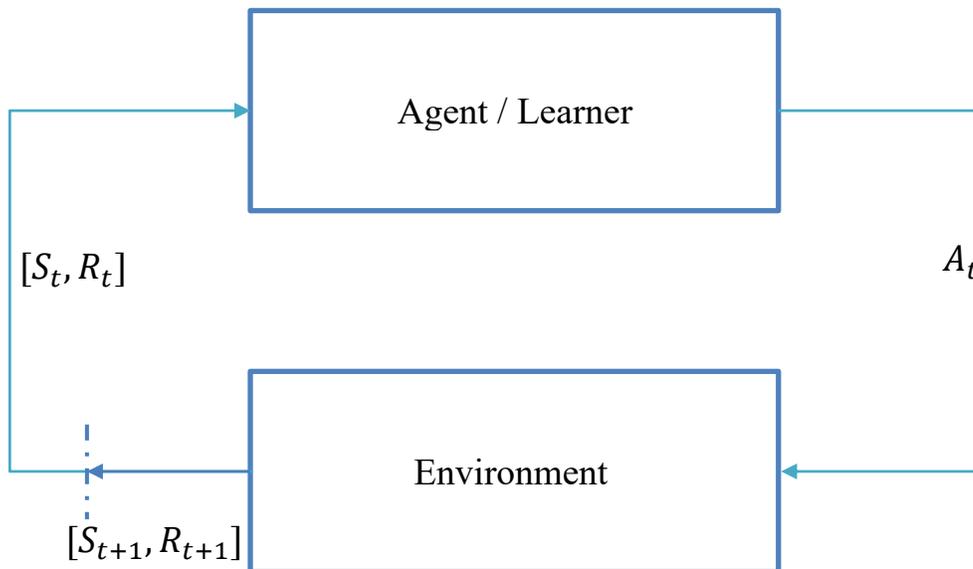


Figure 1.6 A general RL framework. A_t represents the action taken by the agent at time t , S_{t+1} represents the state of the environment at time $t+1$ and R_{t+1} represents the received reward at time $t+1$ based on the action A_t and the state S_t .

respectively. At each time step t , a state S_t of the environment is presented to the agent. The state contains an abstract of the most useful required information from the environment that the agent is present in. The sources of information (signals) presented in a state are defined before the start of the experiment. The agent then decides which action A_t to take among all the presented options according to its policy and the current state S_t . The policy calculates the probabilities of selecting different actions at each state. The chosen action at time t leads to receiving a reward R_{t+1} at the next time step. This behaviour continues until the experiment is terminated. The overall goal in an RL framework is to maximize the “expected return” over the long run. The true (actual) return G_t is defined as the discounted sum of rewards from $t + 1$ to the end:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \quad (1.3)$$

where t is the current time step and $0 \leq \gamma \leq 1$ is the discounting factor, used to discount the future rewards. Discounting means that closer rewards to the current time step will have a larger effect on the return value than the rewards collected in future. If $\gamma = 0$, only the immediate reward (the reward of the next time step) will be taken into account and no attention is given to future outcomes. Conversely, if $\gamma = 1$, all future rewards will have a similar influence to the immediate reward in decision making. As future rewards are unavailable, a value function indicates what is desired in the long run, by estimating the “expected return”. Value function $v_{\pi}(s)$ is a function of a state that estimates how good it is for the agent to be in a particular state, depending on the policy. The value function calculates the expected return using the following equation:

$$v_{\pi}(s) = \mathbb{E}_{\pi}[G_t | S_t = s] = \mathbb{E}_{\pi}[R_{t+1} + \gamma G_{t+1} | S_t = s] \quad (1.4)$$

1.2.2.2 Linear Function Approximation

In real-world situations where the states and their learned values are too large to be represented using tables, value functions need to be approximated using approximation methods. At each state, many sources of information can be available with wide ranges of values. To optimize the usage of all these sources there is a need to have a computationally efficient method to present this information as a state at each time step. Function approximation methods are methods of state representations that make this process happen. In the simplest form, a linear function approximation technique approximates the value of a state using the following linear equation:

$$v = w^T x(s) \tag{1.5}$$

where w is a weight vector, optimized during the learning process (with similar functionality to the weights discussed in Sec 1.2.1) and x is a binary representation of the state-space, called feature vector. The main advantage of using feature vectors instead of storing separate state values in a table can be seen in their ability to generalize across similar scenarios/states while discriminating based on the level of their granularity. In the following sections, two main common methods of generating feature vector representations in linear function approximators are discussed.

1.2.2.2.1 Tile Coding

As indicated by its name, tile coding divides the continuous state-space into discrete units of tiles. Each tile represents a range of sensor values and is activated (i.e., gets the value of 1) once the sensor values fall within the allocated ranges of that tile. All other blocks of tiles remain inactive (with the value of 0). While one layer of tiling provides a chessboard-like environment, in practice, different overlapping tilings with offset are put on top of each other (Fig 1.7 a). This improves the generalization abilities of the system while the number of tiles in a tiling improves

the discriminative capabilities of the system. Therefore, tile coding performs exhaustive partition of the state-space using overlapping square grids. Since the grids are uniform, it is easy to compute which cell the current state falls into which can be beneficial in low-dimensional environments. However, as the number of dimensions grows, the number of required tiles grows exponentially. This can cause challenges with regards to memory availability and computational speed [82].

1.2.2.2 Selective Kanerva Coding

Selective Kanerva coding (SKC) provides a method of activating nearest randomly distributed prototypes to the current state with minimal computation [82]. SKC is inspired by Kanerva [83], where he proposed that a high-dimensional state-space can be represented by a set of randomly distributed points, known as prototypes [64]. SKC attempts to find and activate the c closest prototypes to a state based on Euclidean distance, (Fig 1.7 b) using Hoare’s quickselect [84]. Similar to the tile coding, SKC assures that a unique number of prototypes is activated at each state. Moreover, it outperforms the tile coding approach with increasing the allocated prototypes

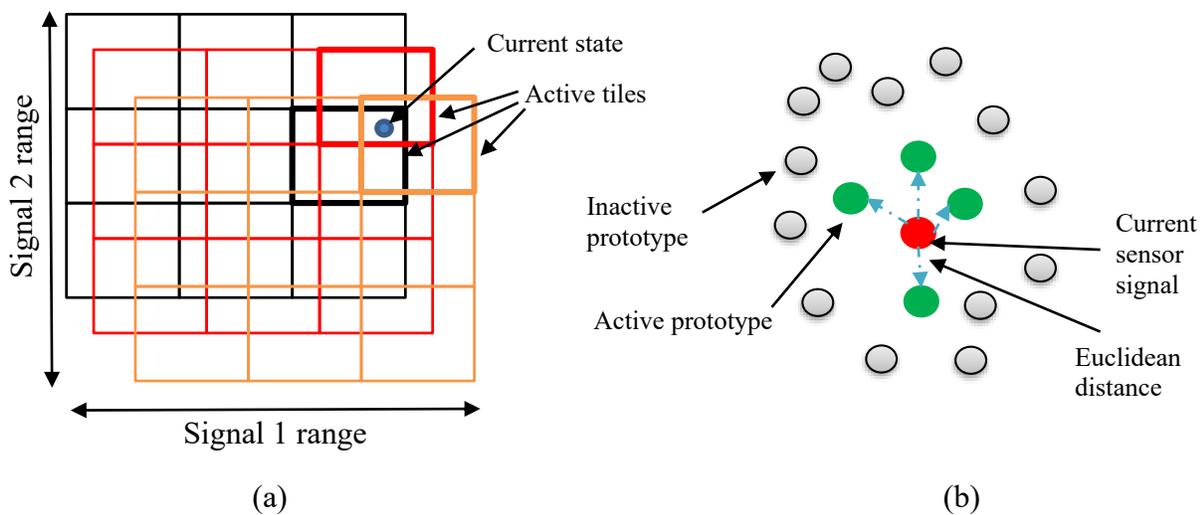


Figure 1.7 A general overview of two feature vector representation methods. (a) Tile coding for a 2D space, (b) Selective Kanerva coding for a single signal. In SKC, different random set of prototypes will be allocated for different signals.

linearly (not exponentially) through the addition of new sensors. This helps the system run faster and avoid dimensionality and memory issues.

1.2.2.3 Temporal-Difference Learning

Temporal-difference (TD) learning is one of the fundamental algorithms in the RL domain. TD learning attempts to update the estimate for the value function using previously learned estimates. In contrast to the Monte Carlo method, TD learning does not wait until the end of an episode/trial to update the visited states; instead, it updates the weight vector associated with the value function (expected return) by bootstrapping [81]. According to equation 1.6, a TD error term δ_t is defined using the difference between discounted future estimate of the value function and the prediction of the current value function, plus the reward term. This error term will be used to update the weight vector accordingly.

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \quad (1.6)$$

TD learning is among the RL methods highly associated with cognitive neuroscience and learning behaviour in animals. An important contribution to this association between TD learning and natural learning was the demonstration that dopaminergic neurons in the ventral tegmental area (VTA) of the midbrain tend to compute reward prediction errors which can be seen as the difference between the actual reward and the reward predicted by the amygdala, orbitofrontal cortex and the ventral striatum [85]–[87]. TD learning attempts to update the estimate of a value function using the mentioned reward prediction errors and similarly, these reward prediction errors are utilized by the dopaminergic reward system to alter the firing rate of the nucleus accumbens in the basal forebrain, resulting in a reward-seeking behaviour [64], [85]. It is worth mentioning that in spite of these similarities, a major difference between the traditional TD learning method and reward-seeking behaviour in animals is that the reward

function is fixed in the former. The reward functionality in animals, however, changes continuously as the state of the body changes. As an example, food can be considered as a triggering reward only in case of hunger [85]. Computational modifications and trials are ongoing to develop more similar-to-reality algorithms in the RL framework [85], [88], [89].

1.2.2.4 General Value Functions

While traditional RL methods attempt to maximize a future reward, in some situations the goal is to instead specifically predict the actual signals of interest. For this purpose, general value functions (GVFs) are introduced and their application for signal prediction have been investigated over the last decade [90]. GVFs can be introduced as value functions (or in the form of value functions [81]) with the capability of predicting unlimited arbitrary signals of interest over a window in the future [91], [92]. Therefore, the computational approaches for learning the approximate of the traditional value functions still can be applied to GVFs (equations 1.5 and 1.6). The difference is that instead of predicting summation of the future reward terms, each signal that is intended to be predicted is referred to as a cumulant (C), which is predicted in the form of a GVF. Similar to equation (1.3), the true return at time t can be computed using the future cumulant values:

$$G_t = \sum_{k=0}^{\infty} \gamma^k C_{t+k+1} \quad (1.7)$$

where γ here is known as the discounting factor or the termination signal [91], specifying the window in the future where the prediction will take place. Using γ , each prediction is a sum of exponentially decaying outcomes with a half-life of T time-steps [60], where T is calculated using the following equation:

$$T = \frac{1}{1-\gamma} \quad (1.8)$$

Each GVF prediction value can also be calculated (as the estimate of the expected return) using equation 1.5 for estimating a value function.

Similar to reward-based systems with linear function approximation, GVF predictions are also generated using weight vectors. Those weight vectors are updated (optimized) using the TD error. The prediction-driven behavior of the GVFs is analogous to the role of the cerebellum.

One of the most fundamental tasks of the nervous system is the ability to make predictions and test those predictions against actual sensory data, with the cerebellum playing an integral role in this behavior [93]. According to forward internal models [94], [95] the cerebellum predicts and modifies the sensory results of motor commands and participates in computing sensory prediction errors by evaluating the predictions against the sensory feedback. Forward internal models can be utilized to accurately predict the state of the environment and the body [96].

Climbing and mossy fibers are two important sources of inputs to the cerebellum and the Purkinje cells, which regulate the motor response, are the single output of the cerebellar cells. By adjusting the efficacy of the mossy-parallel fiber connection with the Purkinje cells through long-term depression, the input from the climbing fibers provides motor error signals [97], [98]. According to Shadmehr et al. [99], The primary motor cortex sends a copy of the motor commands to the cerebellum. With the knowledge of the motor command, the cerebellum can forecast the sensory effects of such motor orders, enabling the musculoskeletal system to execute a movement. Predicted and actual incoming sensations are then compared while moving. If they are aligned, the pattern is kept for the subsequent movement. A warning signal is delivered back to the motor cortex and subcortical regions in the absence of an alignment, activating feedback movement corrections and calibrating the forward model. The mentioned studies in general all confirm the role of the cerebellum in learning to relate the motor commands with the new

sensory outcomes (forward model), rather than learning to associate sensory goals to the new motor commands (inverse model) [100]. This is aligned with the use of GVFs in making forward predictions of the state of the system. For example, GVFs can be used to answer “how long will it take before my bump sensor turns on if I continue to drive?” or “which joint of a myoelectric arm a user will move next?” [61], [101].

Utilizing GVFs to predict signals rather than maximizing a single reward term is advantageous in circumstances where those GVF-predicted signals, arriving in advance before the actual signals, are used by the system for further inferences. GVFs have been incorporated in different applications so far and their contributions have been investigated. In Modayil et al. [92] GVFs were used in mobile robots to formalize the concept of nexting in robots, referred to as continuously predicting the immediate future about sensory inputs. The capability of real-time nexting in robots was tested in that study by predicting 53 sensor readings (e.g., electrical current, voltage, motor temperature, wheel rotational velocity, etc.) using GVFs. Building on that study, Modayil and Sutton [102] introduced GVF-based Pavlovian control in robots, inspired by Pavlovian conditioning experiments [103]. In Pavlovian conditioning experiments, animals’ behavior was adapted while learning to predict an event, even in the absence of any rewards or benefits. The animals initiated an unconditional fixed response to a learned prediction. Their results showed that utilizing a GVF of the over-current signal (an indicator of motor stall), the robot can anticipate wheel stalls and shut down the motors in advance. The reactive safety control was modified by the Pavlovian control, which shut off the motors when it anticipated that too much current will be drawn. This improved the efficacy of the robot and was beneficial for its maintenance. GVFs were also incorporated for walking restoration after SCI. Dalrymple et al. [64] used GVF predictions in the form of Pavlovian control for a model of incomplete SCI in

cats, where ground reaction forces and angular velocities of the cats' intact limbs were predicted and used to determine the next walking state. That information was then used to stimulate the spinal cord to produce the desired movements in the paralyzed limb. Their results showed the benefit of using the predictive strategy in reducing errors in gait pattern detection and also eliminating the need for threshold resetting which was often needed in the absence of the machine-learned strategy. In addition, myoelectric prosthetic hands have also benefited from GVs for addressing known issues such as active joint switching, grip slippage detection, identifying user intent and multi-joint movements [58]. In Pilarski et al. [58], EMG recordings from a participant's dominant arm in addition to the angular information of the elbow and hand joints of a robotic limb (controlled by the participant) were predicted using GVs for controlling a myoelectric training robot. They found that GVs were able to successfully predict the measured signals after short periods of online learning trials. The predictions preceded the actual signals by 0.5–2.0s. Moreover, they also verified the ability of GVs to predict sensorimotor signals when a person with upper-limb amputation interacted with a robotic training prosthesis. In other studies [59], [60], [104], joint angles of robotic devices were predicted with GVs to anticipate the next joint that the myoelectric prosthetic user intends to switch to. Their approach, which was referred to as “adaptive switching”, showed noticeable reduction both in the number of required switching actions that the users may perform to move their desired joint and in switching times. building on those studies, Edwards et al. [61] developed an autonomous switching algorithm using GVs, capable of switching autonomously to the next intended robot joint.

1.3 Thesis Outline

The aforementioned studies showed the potential of GVFs in speeding the tasks performed by both robots and their users, reducing the burden on the users and helping people with disabilities. These improvements formed the motivation for using GVFs in conducting this thesis research.

This work aimed to investigate whether GVFs can be utilized to predict the intention of lower-limb exoskeleton users for controlling the movements of these devices. Specifically, the question framed in this study was to predict “which walking mode, among five walking modes, an exoskeleton user wants to select next?” I hypothesized that “GVFs can be implemented in the control of lower-limb exoskeletons to reduce the number of required switching actions, when switching between modes”. GVFs, in this context, are expected to provide predictive knowledge of the users’ switching behavior and suggest the most likely mode(s) that the users may want to select next. To verify this hypothesis, an adaptive switching strategy using GVFs was designed and incorporated into a lower-limb exoskeleton system for switching between the five walking modes. The system was first tested in an offline setup, where a user walked with the exoskeleton in two experimental scenarios using a switching panel consisting of 5 switch buttons [105]. The collected sensory data and switching actions were then passed to the designed machine learning system post-hoc to verify the capabilities of the system in GVF prediction and learning. After successful implementation of the offline experiments, the more comprehensive online experiments were conducted using a single button for switching between modes. Three users each participated in three unique experimental scenarios, mimicking different real-world situations, to verify the capabilities of the system in real-time. Using the adaptive strategy, a switching list was updated at each time step in real-time, presenting the most likely walking modes that users may select as the first suggestions in the switching list. Therefore, users had to

hit the switch button just once (or twice in cases that uncertainties were present) instead of cycling through a fixed list. To examine the capabilities of the proposed method, the switching list order and the number of required switches (switching actions) under the adaptive strategy at each switching instance were investigated and compared to the lowest possible number of switches achieved using a fixed switching list. This study is an important contribution to the overall goal of restoring walking capacity after CNS injury by taking the users' intention into account and making the control of the lower-limb exoskeletons easier. Chapter 2 addressed the offline verification of the proposed adaptive switching method by providing the recorded data to the learning system post-hoc. Chapter 3 addressed the online verifications of the proposed method through real-time implementation of the learning system and testing across comprehensive experimental scenarios that mimicked real-world situations. Chapter 4 provided general conclusions, further discussion, alignment of the results with findings of systems neuroscience and future directions.

Chapter 2: Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control: Preliminary Results¹

2.1 Introduction

Powered lower-limb exoskeletons provide assistance to their users and have different active joints that can be controlled by the users, depending on the implemented control strategy for the device [106], [107]. What makes these exoskeletons beneficial for rehabilitation and gait restoration research is their capability of tracking the desired motions presented to them with high accuracy, collecting data of different joint motions with their built-in sensors that can be used as feedback, and providing different levels of assistance to users with a variety of conditions such as people with complete spinal cord injury (SCI) (no motor function) and incomplete SCI (limited motor function) [12].

The ultimate goals of many research avenues in this domain are: 1) taking into account users' intention 2) reducing the effort needed to perform tasks and 3) making the orthosis adaptive to the users' need in a safe manner. To this end, many control strategies have been designed and tested experimentally. Generally, there are three main areas of focus for designing control strategies for exoskeletons: high-level, mid-level and low-level control [24], [41]. The main focus of this work is on high-level control and as a result, the remainder of this introduction is allocated to the high-level control concept.

A high-level controller can be seen as a perception and motion planning layer [32] that characterizes the overall status or behavior of the robotic device. Both knowledge of the environment and user-dependent measurements (such as ground reaction forces, joint torques,

¹A version of this chapter was published as: P. Faridi, J. K. Mehr, D. Wilson, M. Sharifi, M. Tavakoli, P. M. Pilarski and V. K. Mushahwar "Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control: Preliminary Results," in 2022 International Conference on Rehabilitation Robotics (ICORR), Rotterdam, Netherlands, Jul. 2022, pp. 1–6. doi: 10.1109/ICORR55369.2022.9896611.

joint angles, etc.) can be used as the inputs to the high level controller, while the output is expected to be a specific mode of walking [41]. The modes available on a control system are predefined modes. These modes are designed based on the needs of the users. They can contain different speeds and lengths of walking [20], various tasks such as stair ascending/descending [20], [33], [34], sit-to-stand and stand-to-sit transitions [20], [35] and also different states of over-ground walking [36]. In this regard, selecting the user's intended next mode can be seen as the major concern of high-level controllers, especially when a variety of modes are present. In a recent review [41], high-level controllers were divided into 4 main categories as: brain-computer interfaces, movement recognitions, terrain detections and manual user inputs. The first category, which mostly uses electroencephalography (EEG) recordings from the brain [37]–[40], faces many practical issues. Artifact removal, requirement of high concentration by the user, the lengthy procedure of preparing the EEG electrodes, and losing accuracy over time are among those issues [41], [42].

Another group of strategies (movement recognitions) aims to determine the desired mode of walking by relying on the user's body movements and forces. Examples are studies focused on threshold crossing techniques [43], [44], and machine learning algorithms, mostly supervised learning, to classify sensor values for specific modes [46], [47]. These systems either require a precise threshold setting or a high amount of recorded data for their training sets [48].

The third category, using sensors or cameras such as infrared distance sensors [51], [52], head-mounted cameras [53], and chest-mounted RGB camera systems [54], has shown promising results. Nonetheless, if implemented independently of other high level control categories, its usage is limited to modes that are only environment-dependent.

The last category (manual user input) works directly through user commands, either with switch buttons or voice-control [20], [21], [41]. Using switch buttons is the most commonly used method of high-level control [41] because of the simplicity of its implementation, capacity for adding more modes and less susceptibility to errors. Despite the benefits of push-button systems, they have several drawbacks. Using a single switch requires high transition times to toggle between modes due to a high number of required switching actions for a switching instance. Using a panel of switches for each mode also limits the capacity of adding many modes. Therefore, both of these push-button strategies make the user feel uncomfortable, reduce the speed of tasks, and require a high degree of mental concentration, thus increasing the chance of errors [41].

Considering all the aforementioned control approaches, this work aimed to 1) reduce the switching-related problems in high-level control of lower-limb exoskeletons while using the switch-button method, and 2) increase the users' confidence in the device by employing reinforcement learning techniques and predicting users' intention. The goal was to design an adaptive switching controller that updates the order of modes in a pre-designed switching list at each time-step based on the user's previous activity and locational information. This information was used to predict the most probable next mode that the user would select, and suggest that mode as the first mode in the switching list. If successful, this will make the switching actions easier and faster, and improve the use of exoskeletons for upright mobility.

2.2 Methods

2.2.1 Robotic Platform

The powered orthosis used in this study was the Indego lower-limb exoskeleton (Parker Hannifin Corporation, Cleveland, Oh, USA) with powered hip and knee joints (by brushless DC motors)

[13]. Joints were also equipped with a potentiometer to provide the actual angles at each time step. The performance of the exoskeleton was controlled by on-board components, connected to a laptop with an Intel Core i7 CPU via USB. The control strategy was designed in Real-time Desktop Simulink environment of MATLAB (The MathWorks, Inc., Natick, MA, USA) and communicated with the exoskeleton system through the CAN interface (Vector VN1610). A walker was equipped with additional components. A 5-button switch panel was designed and mounted on the right side of the walker for switching purposes on the part of the user. To acquire locational information, 3 GARMIN LIDAR-Lite v4 LiDAR (Light Detection and Ranging) sensors were installed on 3 sides of the walker to provide distance to objects around. The system received the external signals and operated at 50Hz. The platform setup can be seen in Fig. 2.1.

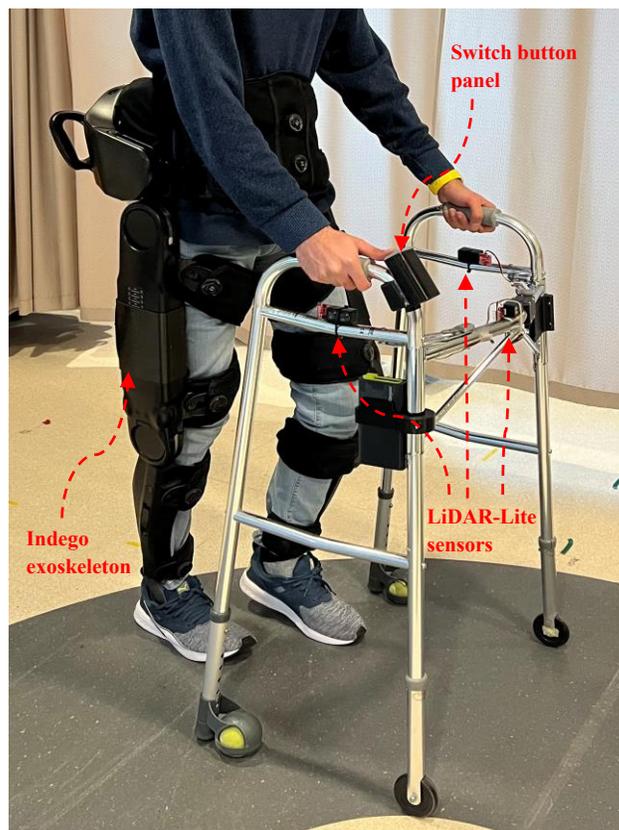


Figure 2.1 A study investigator wearing the exoskeleton with a walker equipped with a switch button panel and distance measurement sensors.

2.2.2 Experimental Procedures

The experiments were performed in 2 different scenarios. For each scenario, the user (neurologically-intact, male, 24 years old and experienced in walking with the exoskeleton) had the authority to switch between 5 available walking modes as: 3 different speeds (slow, normal and fast) and 2 turning directions (left and right) using the switch button panel while walking with the exoskeleton. For the purposes of this study, a switching panel consisting of 5 buttons was designed and used to assess the core capabilities of the machine learning algorithm on predicting the next walking mode, and avoid the delays upon mode switching while collecting experimental data for the machine learning algorithm. However, using a separate button for each mode is not a scalable solution when there is a larger number of modes to switch to.

After a mode was selected by the user, the desired trajectory of that mode was implemented by the designed controller using central pattern generator (CPG) concepts [44], [108] in which specific pre-defined frequency and amplitude of each mode were passed through the differential equations of motion, and a reference trajectory was updated for each joint to allow a smooth transition between walking modes. The built-in proportional-derivative (PD) tracking system of the Indego exoskeleton with modified gains [109] was used to track the desired trajectories. The characteristics of the designed walking modes are shown in Table 2.1. These were chosen based on the mean gait speed of people with SCI walking with the Indego exoskeleton [110].

In the first experimental scenario, the user walked through a rectangular path (4.5m * 5m) 6 times (rounds), using the walking modes as shown in Fig. 2.2. This scenario was designed to test the core machine learning capabilities in prediction and learning. In the second scenario, the user had the authority to select between two different paths, separated by an obstacle, at each round when he reached the starting position (Fig. 2.2). A total of 11 rounds (from starting position,

back to the starting position) were walked by the user, with an arbitrary order of choosing between the two available paths. This scenario was designed with the goal of testing the capability of the designed machine learning strategy in differentiating between different paths and providing reasonable mode suggestions upon approaching an obstacle.

2.2.3 Machine Learning Strategy

The machine learning strategy implemented in this work was based on an extension technique to conventional reinforcement learning (RL) called general value functions (GVFs) [90]. GVFs are value functions with the ability of representing temporally extended predictions of arbitrary signals [91], [92] and have been implemented to design adaptive and autonomous controllers in myoelectric prostheses [58], [59], [59], [60]. In this study, GVFs were used to provide anticipatory knowledge on the next possible walking mode to be selected by the user from a switching list in order to minimize the number of manual switches needed to be performed by the user. The proposed machine learning strategy was implemented on all of the collected data, in an offline setup, for the purpose of preliminary verifications.

Position and distance information was used to anticipate the levels of mode activities. Signals from the 3 LiDAR sensors in addition to the position of the user in the 2D x-y plane (computed mathematically by considering a reference coordinate system and the time spent in each mode

Walking mode	Walking characteristics	
	<i>Speed (m/s)</i>	<i>Stride length (m)</i>
Slow speed	0.23	0.875
Normal speed	0.31	1.050
Fast speed	0.39	1.125
Left/Right turn	0.15	0.725

Table 2.1 Characteristics of designed walking modes

❖ Colored arrows are indicative of the modes being used as: Slow speed, Normal speed, Fast speed, Left turn, Right turn

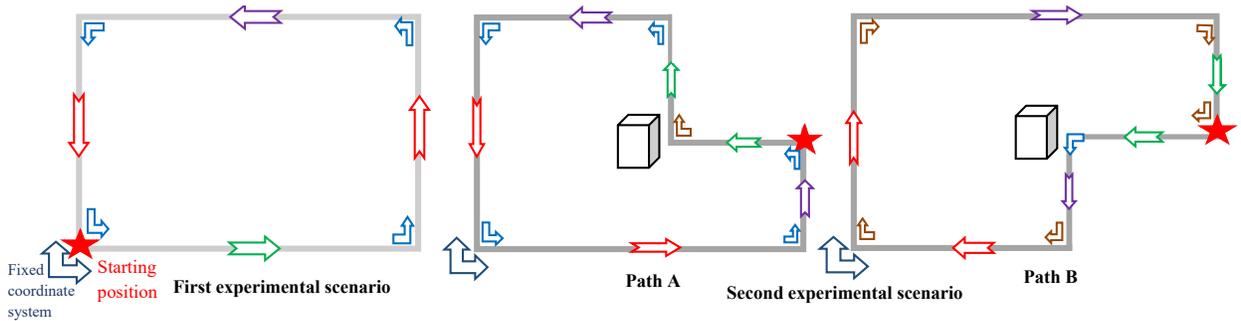


Figure 2.2 Experimental design scenarios. In the first scenario (left image) a rectangular path was walked by the user, using different walking modes in the directions specified on the image. This pattern was repeated for six times. In the second scenario, the user had the authority to select between two available paths (A,B) for each round of walking. A total of 11 rounds were walked in the order of: A,A,B,A,B,B,B,A,A,B,B. A fix coordinate system was used at the bottom left of the experimental areas to calculate the position of the user at each time-step based on the speed of walking and the time spent in a specific mode and specific direction. The red star indicates the starting position for each scenario.

along with the speed of that mode) formed the state-space (s) of the system (Fig. 2.3). LiDAR signals were able to provide information on reaching to an obstacle. These five signals were then passed through a function approximation method called Selective Kanerva coding (SKC) [82] to provide a binary vector. The resultant binary vector, called feature vector (x), contained 15000 elements in which the 650 closest elements to each state were active at all times. For algorithm details and parameters' selection please refer to Dalrymple et al. [111].

An activation level (c) was also defined for each mode. The walking mode selected by the user at each time-step was considered the active mode, given a value of $c_j=1$ while all other modes received a value of $c_j=0$ ($j \equiv$ number of modes). One GVF weight vector w_j was also initialized at the beginning of each experiment for each mode that was updated at each learning time step (Algorithm 2.1). The inner product of the weight vector and the feature vector (from SKC) was introduced as the GVF prediction value (p_j) for each mode. These GVF predictions were then ranked in the switching list based on their relative magnitude in a descending order, with the current active mode being ranked last, regardless of its prediction value.

GVF predictions (p_j) and their weight vectors were then updated at each time-step using the temporal-difference learning method (TD(λ)) presented in Algorithm 2.1, in which a TD error signal (δ) was formed as the difference between the discounted future prediction and the prediction for the current state, plus the current mode activation signal (c_j). Replacing eligibility traces (e_j) were then used [112] with TD error (δ) to update the weight vectors. For more information on TD learning please see [60], [81]. The discounting factor used in updating the TD error (δ) was set to $\gamma=0.992$ for all modes and $\alpha=0.001$ was used for weight vector updates as the step-size parameter, based on a comprehensive trial and error. The bootstrapping parameter in the replacing eligibility traces update was set to $\lambda=0.9$, as is often standard [111].

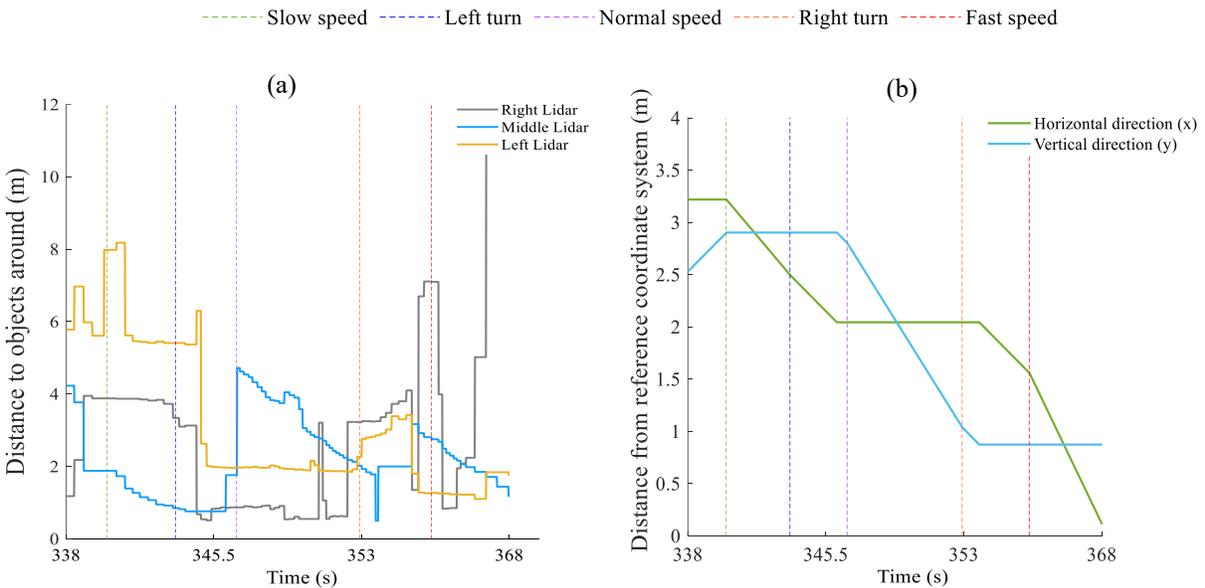


Figure 2.3 Signals used in the state-space of the system for a sample of recorded data during the second experimental scenario (path B). The selected modes by the user are specified with dashed lines at the switching instances for both images. (a) The signals from the LiDAR sensors. These 3 signals were used to monitor the distance of the user from obstacles. (b) The absolute position of the user in the 2D space, computed mathematically at each time step (0.02 sec), using a reference fixed coordinate system and related mathematical relationships. The signals from the LiDAR sensors and the computed absolute space were used for identifying similarities in the modes being selected at specific positions. The horizontal axis for (a) and (b) represents the specific time these signals were taken during walking.

Algorithm 2.1 GVF prediction and learning with TD(λ)

Initialize w, e, s, x

Repeat every time step:

Observe next state s

$x' \leftarrow SKC(s)$

For all modes j **do**:

Observe mode activity signal c_j

$\delta \leftarrow c_j + \gamma w_j^T x' - w_j^T x$

$e_j \leftarrow \min(\lambda e_j + \delta, 1)$

$w_j \leftarrow w_j + \alpha \delta e_j$

$p_j \leftarrow w_j^T x'$

$x \leftarrow x'$

Rank the modes in the switching list

2.3 Results

A comparison of the number of instances in which the next selected mode by the user was ranked first, second, third and fourth in the switching list for both experimental scenarios under adaptive and best possible non-adaptive control is shown in Fig. 2.4a. The comparison is based on the use of a single switch button to transition to the desired mode. In the adaptive controller strategy, the system was able to quickly adjust the switching list based on the user preferences and with regards to locational and positional information provided to the system. In the first experimental scenario (Fig. 2.4a, left), it can be seen that the system was able to predict correctly the next mode at the times of mode switching by the user and rank it as the first in the switching list for 82.98% (39 out of 47) of times, while all other switches that were not ranked first were limited to the first and second rounds of walking along the rectangular path. However, using the best non-adaptive strategy, as computed separately for each experimental scenario, showed that in only 18 out of 47 switching instances (38.3% of times) one switching action was required from user and the remaining selections required two or more switching actions. Switching

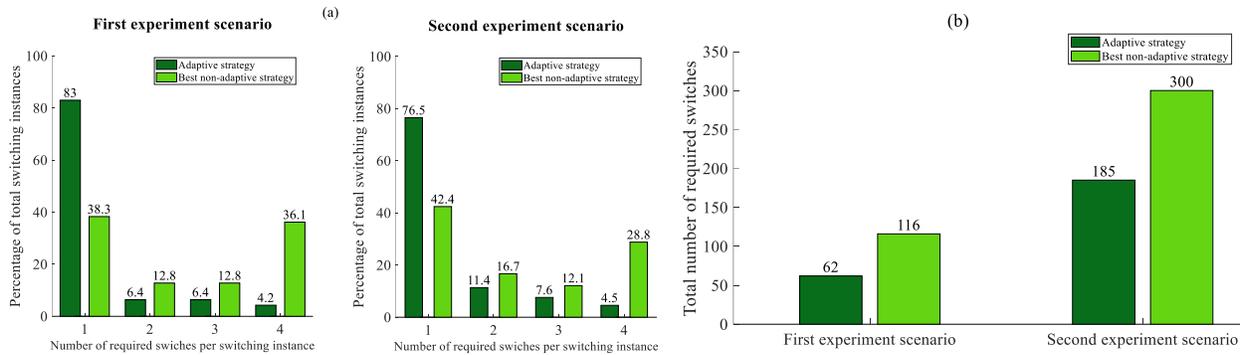


Figure 2.4 Number of switches required using a single switch button under the adaptive and the best possible non-adaptive strategy, computed post-hoc. (a) The percentage of times where 1,2,3 or 4 switches were needed to reach to the user’s desired mode at each switching instance for the first (left) and second (right) experimental scenarios. (b) Total number of required switches using the two strategies under the two experimental scenarios.

numbers for the second experimental scenario (Fig. 2.4a, right), which was more complicated and had two different paths, showed that in 76.52% of instances (101 out of total 132 switching instances) the next selected mode was ranked first, while this number for the fixed-list strategy was 34.1% lower (42.42% of instances, 56 out of 132).

The total number of required switches to perform the tasks under each strategy is shown in Fig. 4b. For the first experimental scenario, the total number of required switches decreased by 46.55% for the adaptive strategy in comparison to the best computed non-adaptive case. Also, the second experimental scenario showed a 38.33% decrease in the number of total switches upon using the adaptive strategy relative to the best non-adaptive strategy which was also task-specific. These advantages are also expected to be more appreciated upon increasing the number of available modes in the switching list.

An example of the GVF’s predictions as an indication of the expected mode activation levels for a subset of collected data from the second experimental scenario is depicted in Fig. 2.5 after four rounds of walking through paths A,A,B and A (Fig. 2.2). Solid lines present the normalized prediction values for each mode while the dashed lines indicate the user switches and transitions

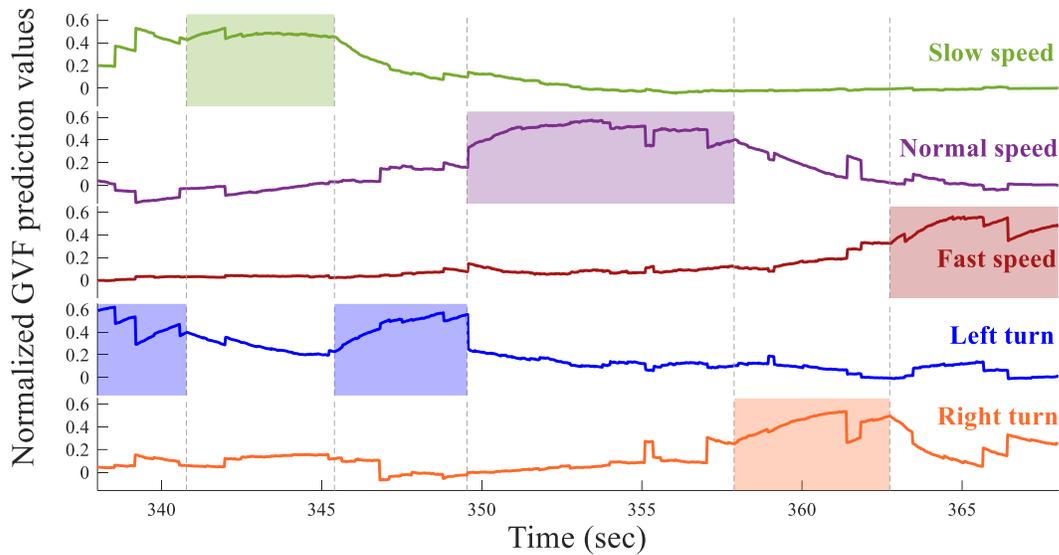


Figure 2.5 Normalized GVF prediction values for a sample of recorded data during the second experimental scenario (path B) after 4 rounds of walking in the order of A, A, B and A. Solid lines show the predictions for each mode activation level and dashed lines represent the user's switching actions from the previous active mode (left side of the dashed lines) to the next intended mode (right side of the dashed lines). Modes in the switching list were ranked at each time-step (0.02 sec) based on their GVF prediction value in descending order, except for the active mode that always was ranked least, regardless of its GVF value.

from one mode to the next. It can be seen that upon transitions, the next intended mode had the higher prediction value than other modes, except for the current active mode, and that the trend of that prediction was ascending a few time-steps before the switching action, with some fluctuations.

2.4 Conclusions and Discussion

This study demonstrated, for the first time, a proof of concept of GVF learning and prediction in lower-limb exoskeletons. Specifically, we demonstrated the application of a machine learning approach to reduce the burden on the users for manually switching between different available walking modes. Considering the two experimental scenarios in this work, an average decrease of 42.44% was seen in the total number of required switches, using the adaptive strategy in comparison to the best possible non-adaptive strategy. The techniques implemented in this work demonstrate a great potential for continuous real-time implementation of adaptive switching

algorithms in lower-limb exoskeletons. The results showed that the proposed method was able to reduce the required switching actions noticeably in comparison to the best possible fixed switching list for each task, while using a single switch button. Using the adaptive switching approach, the target population can not only contain people with SCI (either complete or incomplete), but also be beneficial for people with other conditions such as stroke, multiple sclerosis or other groups who need assistance during walking. Moreover, the core machine learning technology has also the capability and potential to be applied to other domains where generalities and similarities in the movements being performed are present.

There were some limitations in this study. The experiments were limited to the lab environment so the LiDAR sensors received noiseless signals. Also, the method used for determining the location of the user (although was only used offline and for verification purposes) cannot be applied to the real world, and high precision GPS systems are needed. Moreover, although limited walking modes were designed due to the restrictions of the lab environment, the system has the ability of predicting unlimited number of GVFs and re-ordering their respective walking modes in the switching list.

Future goals and next steps involve assessing the online capabilities of the machine learning system, designing less predictable experimental scenarios, utilizing more users and the addition of autonomous features to the system.

Chapter 3: Continual Machine Learning and Unlearning in Lower-limb Exoskeletons to Modify Switching-based Interfaces²

3.1 Introduction

Research on robotic lower-limb exoskeletons as a method for both rehabilitation and walking assistance continues to evolve. In clinical settings with research focused on rehabilitation, these devices are used as a training tool for improving walking capacity, and for overcoming secondary conditions after neural injury or disease such as reducing spasticity and enhancing bowel and bladder function [113]–[115]. However, the main goal of introducing these devices in the medical field is to assist users to walk again independently, both at home and in the community [116].

Lower-limb exoskeletons currently on the market include Indego [117], ReWalk [118], Hal [19], Exo H3 [119], REX [120]; several other prototypes aim to enter the market in the upcoming years [20], [121]–[123]. Manufacturers of these exoskeletons are currently competing to overcome the existing structural and control-related limitations of the devices to provide systems that are widely accepted by the community of users. Structural limitations include the generally low number of degrees of freedom with movements confined to the sagittal plane, the high weight of the devices, expensive electrical motors that are prone to overheating and have limited thrust force, and the need for continuous power supply [18], [23]. Controlling the movements of the available joints and being responsive to users' commands is another equally important challenge in exoskeletons, hindering their wide use in the community [10], [57], [124]. The ultimate goal of many research efforts in the control domain is to take the user intention into

²A version of this chapter is submitted as: P. Faridi, J. K. Mehr, D. Wilson, F. Gauthier, M. Tavakoli, P. M. Pilarski and V. K. Mushahwar “Continual Machine Learning and Unlearning in Lower-limb Exoskeletons to Modify Switching-based Interfaces,” to a peer-reviewed journal.

account while reducing the effort needed to perform tasks, and eventually making the orthosis adaptive to the user's needs in a safe manner.

To this end, many control strategies have been designed and tested experimentally, with focus on one of three primary areas of controlling robotic devices, known as 1) high-level 2) mid-level and 3) low-level control [10], [125]. In general, high-level controllers determine the walking mode, mid-level controllers shape the desired joint trajectories or torques [126]–[128], and low-level controllers aim to track the desired joint trajectories or joint torques through position [129], [130] or torque/force [131], [132] controllers. The work in the present study focused on designing an intelligent high-level controller, and the remainder of this section summarizes work related to high-level control concepts.

As a planning and perception layer [32], the high-level controller determines the robotic device's general state or behavior. Various walking-related behaviors can be considered in the high-level aspect of the lower-limb exoskeletons, depending on the purpose and target users of the device. Exoskeletons in industry are mostly targeted towards supporting users' stability, safety and reducing body joint torques in sitting and standing positions while carrying heavy loads. Medical lower-limb exoskeletons, on the other hand, target the behaviors seen in daily walking patterns in the community. These can range from walking with different speeds and step lengths, turning left/right, ascending/descending stairs, transitioning from sit-to-stand and stand-to-sit, and avoiding obstacles [33]–[35]. When several walking/operating modes are available in the underlying control strategy of the device, the high-level controller needs to propose the most desired mode to the user. This can be achieved through either direct user commands or the intelligence of the high-level controller. Researchers have investigated various methods to provide an acceptable strategy for this purpose.

Deep learning methods are utilized widely for high-level control. In Liu et al. [133], a vision-assisted autonomous strategy was implemented on a custom-built exoskeleton to detect small, low-height obstacles in the walking area using an RGB camera. In Laschowski et al. [78], a large dataset of images collected using a wearable camera during both indoor and outdoor walking scenarios was utilized to train a deep-learned model. An environmental classification algorithm was designed and used for walking mode recognition with robotic prosthetic legs. Sharifi et al. [108] proposed an autonomous locomotion trajectory shaping controller by estimating the human-exerted interaction torques using a deep neural network. Although deep learning methods are capable of generating precise predictions, they need to be trained on previously collected and labeled data. Moreover, people with complete spinal cord injury (SCI) are not able to exert forces detectable by the system. As a result, there is a need for a method capable of real-time learning, without the need to be trained with a pre-labeled dataset and independent of the users' lower limb forces.

In the simplest and most common control form, various walking modes are pre-ordered into a switching list. The users need to switch between those modes using a switching panel. The panel in general can consist of either a separate button for selecting each walking mode or one switch button for switching through a list and another button for selecting that mode [20], [134]–[137]. In the presence of many operating modes, however, this switching regime can be tedious and less desirable for the users. The ability to add more modes is restricted when using one button for each mode. While a single-switch system can be used, transitioning between operating modes with a single switch needs many switching actions (pressing a switch button repeatedly to select the desired mode), which leads to long transition periods for each switching instance [10]. Work in our lab found that it takes on average 1 sec for each switching action, which includes the time

the notification of the selected mode is received (0.5 sec) and the user's confirmation of the mode or switching (0.5 sec) to the next mode in the switching list (unpublished results).

A recent qualitative study involving prosthesis users and therapists in Europe described the participants' opinions on using a prosthesis with several operating modes and manually switching from a pre-defined list of those modes [138]. The study concluded that almost all of the participants found the manual switching approach as a major problem, describing it as “too time-consuming”, “taking lots of effort when switching many times” and “mentally and physically exhausting”. The same issues are present when using lower-limb exoskeletons. A newly published qualitative study from the Netherlands interviewed 13 people with SCI with exoskeleton experience for their needs and wishes for the future lower-limb exoskeletons [57]. The most important need was making the exoskeleton systems easier to use and work with. The study reported a consensus to incorporate different modes such as step and/or speed adjustment controls to enhance the exoskeletons' utility. There were uncertainties on how to implement the controller though, suggesting control interfaces to be put on crutches, with functionality similar to the automatic gear transition mechanisms in cars [57].

To address the critical issue of switching between modes, a real-time strategy, called adaptive switching, was designed in the present work using General Value Functions (GVFs) and reinforcement learning (RL) methods. The goal was to learn the exoskeleton users' intention as they walk with the exoskeleton, predict the walking modes that they will utilize in future steps and propose that mode to them at the right time. This promises to reduce switching-related issues in the high-level control of exoskeletons while using a single switch button, without a need to be periodically trained.

We previously demonstrated the potential of this adaptive switching system in predicting a user's intention [105]. That work showed the capabilities of the learning system in an offline setup, where data from a person walking with a lower-limb exoskeleton were passed to the learning system post-hoc. The present work extends the previous work in 1) assessing the real-time capabilities of the machine learning system, 2) testing the method in more advanced scenarios in the presence of uncertainties and randomness, 3) modifying the function approximation method to generalize across different scenarios, and 4) examining the unlearning capabilities of the system.

3.2 Methods

3.2.1 System Configuration

The Indego lower-limb exoskeleton (Parker Hannifin Corporation, Cleveland, OH, USA), a powered orthosis with motorized hip and knee joints, was utilized in this study [117], [139]. The movements of the exoskeleton joints were controlled by the Real-time Desktop Simulink environment in MATLAB (The MathWorks, Inc., Natick, MA, USA).

The exoskeleton was used along with a walker that had custom-made 2 button switch (one button for switching between modes and another for confirmation) and three GARMIN LIDAR-Lite v4 LiDAR (Light Detection and Ranging) sensors mounted on its three sides to detect objects. Moreover, the position of the users was determined in 2D by using 8 motion capture cameras (Vicon Motion Systems Ltd, Oxford, UK) tracking a reflective marker, mounted on one side of the walker. The setup can be seen in Fig. 3.1.

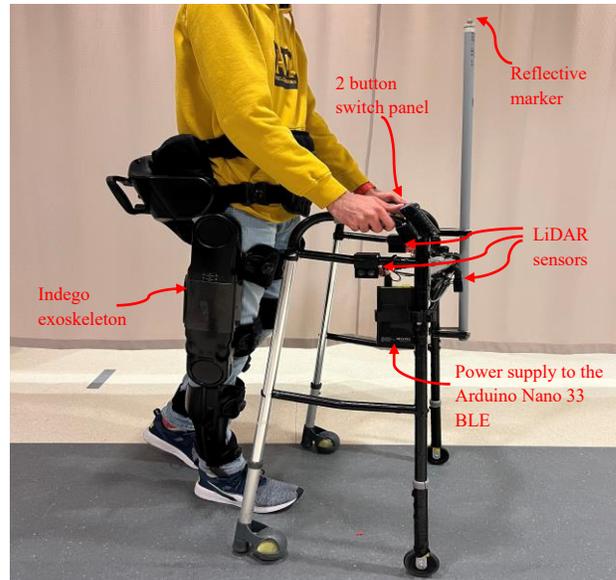


Figure 3.1 An experimenter (user) wearing the exoskeleton with a walker equipped with 2 button switch panel, distance measurement LiDAR sensors and a reflective marker.

3.2.2 Communication Methods

The exoskeleton was connected to a computer with an Intel Core i9 CPU via USB. The software communicated with the exoskeleton robot through the CAN interface (Vector VN1610). The main machine learning code was run in a separate MATLAB environment and the walking mode (determined by the high-level controller) was sent in real-time to the Simulink environment of the exoskeleton at each time step through Universal Datagram Protocol (UDP). Switch button hits and LiDAR data transmissions to the MATLAB environment were established through an Arduino Nano 33 BLE (Bluetooth Low Energy) board that was wirelessly powered through a power bank. The positional information was passed from the Vicon system to the MATLAB program in real-time using the Vicon's DataStream SDK. The system received the external signals and operated at ~12 Hz.

3.2.3 Experimental Procedures

The experiments were performed by three neurologically intact experimenters (male, age: 25 ± 2). Each user (u_i) completed experimental scenarios consisting of Static (a static arrangement of

objects in the environment) and Dynamic (a changing arrangement of objects) settings. Specifically, three unique experimental scenarios were designed for each of the users: 1) Static-Random pathways, 2) Static-Learning and unlearning, and 3) Dynamic. For each scenario, five walking modes were available to switch to, namely, a) Slow speed, b) Normal speed, c) Fast speed, d) Left turn, and e) Right turn. The speeds were determined using the average gait speed of people with SCI using the Indego exoskeleton [105], [110]. A switch button was allocated for switching between these modes. Upon initiating a switching action, an audio cue was used to present the mode suggested by the system to the user. Users had the authority to either confirm the selection of that mode or hit the switch button again to get the next suggested mode. A separate button was used for confirmation purposes. The adaptive switching list froze when a switching action started, and remained the same until a selection was confirmed.

A reference trajectory was updated for each joint to enable a smooth transition between walking modes after the user had confirmed the selection of a mode. A mid-level controller then implemented those updated trajectories for that mode using central pattern generator concepts [44], where the specific pre-defined frequency and amplitude of each mode were passed through the differential equations of motion to update the reference trajectory correctly. The Indego exoskeleton's integrated proportional-derivative (PD) low-level position controller with adjusted gains [109] was used to track the desired trajectories.

3.2.4 Experimental Setup

The three experimental scenarios are depicted in Fig. 3.2 for all three users. Each experiment had three sets, where the learned weights were transferred from one set to the next set, and each set was performed at a different day to mimic real-world day to day usage of the exoskeletons. Static scenarios were performed for 24 rounds (8 rounds for each set) and the Dynamic scenarios were

performed in 30 setups (10 setups at each set). The first set for each scenario was considered as the initialization set, where the machine learning system had no previous knowledge of the tasks and the environment (early learning). The details of each experimental scenario are described below:

1) Static-Random pathways: Blue (B) and green (G) arrows show the available pathways in Fig. 3.2 (top row) for reaching from a starting position (shown as model wearing the exoskeleton) to an end point. Yellow (Y) and red (R) arrows show the pathways to return to the starting point. Starting from the start point and returning back to that point was called a round. The order of traversing the colored arrows was determined randomly but was the same for all three users as:

Set 1 (B, Y, B, Y, B, Y, G, R, G, R, G, R, G, Y, B, R);

Set 2 (G, R, G, Y, B, Y, B, R, B, Y, G, R, B, R, G, R);

Set 3 (G, Y, B, Y, G, Y, B, R, G, R, B, R, B, Y, G, Y).

The goal of this scenario was to assess the capabilities of the system in learning the modes used in different representations and providing the most likely walking mode(s) that the user may select, as the first (and second) ranking in the switching list. As an example, user 1 (u_1) had the authority to choose either left or right when facing the obstacle. The expectation from the system was to rank the Left and Right turning modes at the top of the switching list, while in other situations where only one mode should be selected, that mode was expected to be at the top of the list.

2) Static-Learning and unlearning: Users traversed the experimental area (Fig. 3.2, middle row) using the modes in the green and yellow arrows for half of the length of the experiment (phase 1,

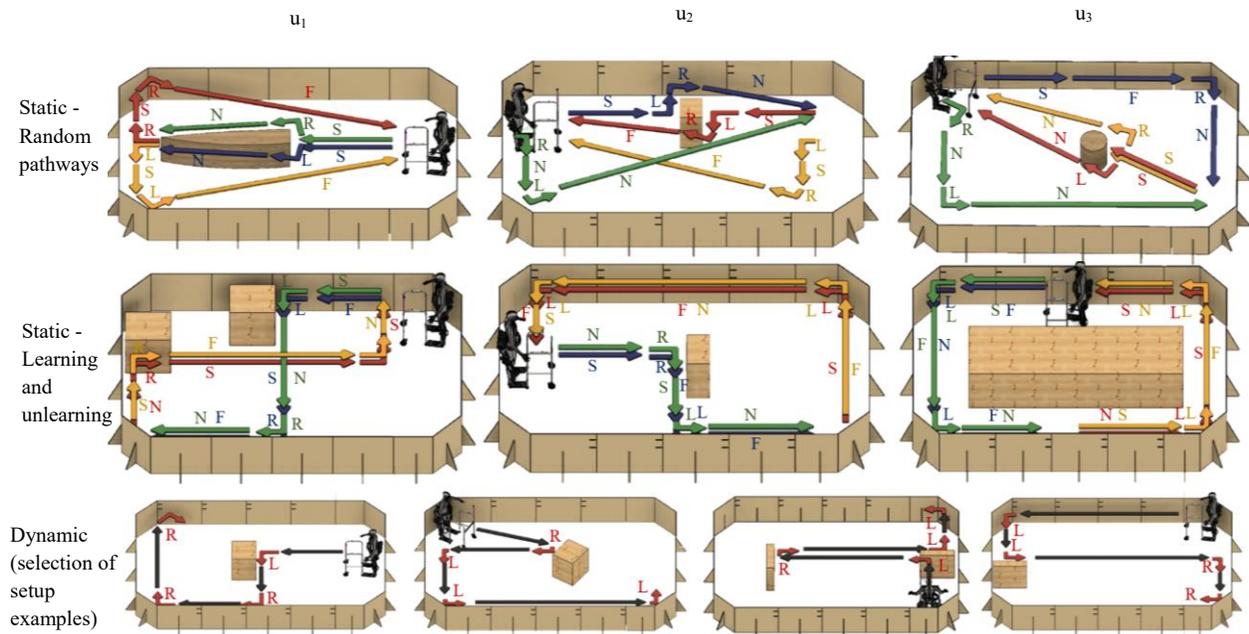


Figure 3.2 Three experimental scenarios for three users (u_i) are shown. Top row: Static-Random pathways. Middle row: Static-Learning and unlearning. Bottom row: Dynamic. The five available modes in each scenario were Slow (S), Normal (N), Fast (F) walking speeds and turning Left/Right (L/R). Only 4 examples (out of 90 unique setups) are shown for the Dynamic scenario, demonstrating how the environment changed in 4 consecutive setups for user u_1 .

first 12 rounds) and then changed the walking modes (based on blue and red arrows) for the remaining half of the experiment (phase 2, following 12 rounds). The goal in this scenario was to verify the ability of the learning system to unlearn the modes used in the first phase and learn the new modes when phase 2 started.

3) Dynamic: Each setup had a unique arrangement with different numbers, types and locations of obstacles (Fig. 3.2, bottom row). Some obstacles just blocked the pathway in front of the users, while others blocked the front and either the left or right side. The goal was to assess the capabilities of the system in suggesting both the Left/Right turning modes when facing the front-only obstacles and suggesting the correct mode (only Left or Right) while facing the front-side obstacles, regardless of their location. This verified that the system does not learn by memorizing the pathways, but learns the users' behavior based on the specific representations to the system.

We also verified the generalization abilities of the system by placing obstacles in different

locations, where previously users walked straight using one of the speed modes. Each setup had 4 turning instances (shown as red turning arrows in Fig. 3.2), and the system accuracy was tested at those instances. Note that as opposed to the “rounds” in the Static scenarios, “setups” in the Dynamic scenario did not have the same starting and ending locations.

3.2.5 Learning Strategy

To learn real-time predictions of walking mode utilization, an extension to the conventional RL value functions, known as GVFs [140] was utilized. A GVF is a temporary extended prediction of an arbitrary signal. GVFs generalize the conventional RL value functions in two aspects: 1) they can predict any non-reward signal of interest (called cumulant, c), rather than just predict the reward, and 2) GVFs allow transition-dependent discounting, meaning that the discounting factor ($\gamma \in [0,1]$) can be a function of transitions (states and actions) [141]. The discounting factor determines the prediction timescale over a window in the future. In this study however, a constant discounting factor was determined to be sufficient. GVFs have been incorporated in different applications so far and their contributions have been investigated in speeding the tasks performed by robots and their users, reducing the control burden on the users, removing the need for manual resetting the control parameters and helping people with disabilities [92], [111], [142]–[145].

The cumulants reflected the active walking mode. This means that upon confirmation of a mode selection on the part of the user, the selected (active) walking mode received a cumulant of $c_j=1$ (j = number of modes), and all other modes received a cumulant of $c_j=0$. The active mode was assigned $c_j=1$ until another mode was selected. These cumulants then participated indirectly in forming the GVF predictions (p_j) for each walking mode. The predicted magnitudes of the GVFs were then used to rank the modes in the switching list in a descending order at each time step

(i.e., the mode with the highest GVF prediction (p) was ranked first in the list). The current active mode was always ranked last in the list to allow for other modes to be presented first in the switching instances. If none of the mode GVFs (except for the current active mode) was above a learning threshold at the start of a switching action (mostly during early learning), the adaptive switching list was considered unreliable, and a fixed default list of Slow, Normal, Fast, Left, Right was introduced to the users. This means that the controller only updated the switching list when it was certain about the next mode to be selected. The learning threshold was set to 5.50 based on preliminary tests. The fixed default list was the list that the experiments started with and was determined based on the switching lists incorporated into the exoskeleton systems, putting the speed modes in an ascending order at the top of the list [20].

Algorithm 3.1 shows the adaptive strategy, containing the prediction method of GVFs and the update procedure. GVF predictions were learned using temporal-difference ($TD(\lambda)$) learning [81], [142]. $TD(\lambda)$ learning updated a weight vector associated with a GVF in question using a TD error (δ). TD error was the difference between current estimate and a sampled bootstrapped return. The sampled bootstrapped return was the current cumulant plus the discount of the next state value. The TD error formation can be seen in Algorithm 3.1, line 7. GVF predictions were formed using the inner product of a weight vector (w_j) and a feature vector (x). The weight vector for each mode was initialized for each scenario and updated at each time step using a learning rate (α), the TD error, and a replacing eligibility trace (e_j). Eligibility traces are among the basic mechanisms of RL, seen as a short-term memory vector parallelizing the weight vector [81]. When a weight (w_j) component contributes to the creation of a prediction, the accompanying e_j component is raised and then starts to fade away. If a nonzero δ occurs before the eligibility trace returns to zero, learning update takes place in that component of w_j . The trace-decay

(bootstrapping) parameter ($\lambda \in [0,1]$) controls how quickly the trace decays. Based on several preliminary tests, the learning parameters were designed as follows:

$\alpha=0.0001$, $\gamma=0.98$ and $\lambda=0.9$.

Algorithm 3.1	Adaptive switching algorithm
Initialize w, e, s, x	
Repeat every time step:	
Record next state s	
$x' \leftarrow SKC(s, k, p_V, p_L, \vec{m})$	
For all modes j do :	
Record mode activity signal c_j	
$\delta \leftarrow c_j + \gamma w_j^\top x' - w_j^\top x$	
$e_j \leftarrow \min(\lambda e_j + x, 1)$	
$w_j \leftarrow w_j + \alpha \delta e_j$	
$p_j \leftarrow w_j^\top x'$	
$x \leftarrow x'$	
Rank the modes in the switching list	
Send the current active mode to the end of the list	
If switching is started:	
Freeze the switching list and use it until a mode is selected	

3.2.6 Function Approximation Method

Feature vector (x) is an important component in shaping the GVF predictions. A function approximator takes the representations (sensory data) of the environment and produces a feature vector that can participate in prediction and learning. Here, the sources of representations were LiDAR signals (3 independent values) and the 2D positional information (from Vicon motion capture system). These five signals were normalized and introduced to the algorithm as the current state (s) at each time step. Selective Kanerva Coding (SKC) [82], [83] was the function approximator used in this study. SKC distributes a set of random points (known as prototypes) in space and attempts to find and activate the m closest prototypes to a state based on Euclidean

distance (d), using Hoare’s quickselect [84]. The goal was to design the SKC structure in a way that it can both discriminate between different, separated states and generalize over similar situations. This design process has been a major challenge in the RL literature, referred to as feature selection [146].

To this end, two unique groups of $k=10,000$ normalized prototypes (p) were randomly distributed for each experimental scenario. One of the groups in 2D space (for the 2D Vicon signals, $p_V=rand(k,2)$) and the other group in 3D space (for the 3D LiDAR signals, $p_L=rand(k,3)$). For each group and at each time step, the m closest prototypes were activated (i.e., set to 1 in the feature vector) three times, each time with unique values of $\vec{m} = [500, 300, 100]$. These values were chosen based on preliminary trial and error and with respect to the total number of k prototypes. Therefore, each group in total provided a vector of length $3k$ with $\sum \vec{m}=900$ values equal to one, and the remaining values set to zero. The resulting Vicon and LiDAR-related vectors were then concatenated, forming the feature vector with $2\sum \vec{m} = 1800$ active prototypes from the total of $6k = 60,000$ elements. The activated elements were therefore the ones that allowed their corresponding weight elements to participate in producing a GVF prediction value. Algorithm 3.2 shows the details of how SKC generated the feature vector. Using 3 m values instead of one m value resulted in an increased granularity and resolution of the representations and as a result, an improved prediction performance without adding substantial additional memory or processing needs [146]. It was determined through trial and error that increasing the number of m values (to more than 3) would not add additional resolution benefits while increasing the computational load. An example of the total and activated prototypes in a state (the representations of Vicon and LiDAR values at a single time step) for each of the two groups is shown in Fig. 3.3.

Algorithm 3.2 2D+3D Selective Kanerva Coding (SKC)

Parameters provided s, k, p_V, p_L, \vec{m}

Reset $D_V, D_L, x' = \text{zeros}(k,1), \text{zeros}(k,1), \text{zeros}(6k,1)$

$D_V \leftarrow d(p_V, s_i)$ **For** i in $[1,2]$ $d = \text{Euclidean distance}$

$D_L \leftarrow d(p_L, s_i)$ **For** i in $[3, 4, 5]$

$I_V \leftarrow \text{Quickselect}(D_V, \vec{m})$ Selecting the indices of

$I_L \leftarrow \text{Quickselect}(D_L, \vec{m})$ m lowest values

$x'(I_V) \leftarrow 1$

$x'(I_L+3k) \leftarrow 1$

Output x'

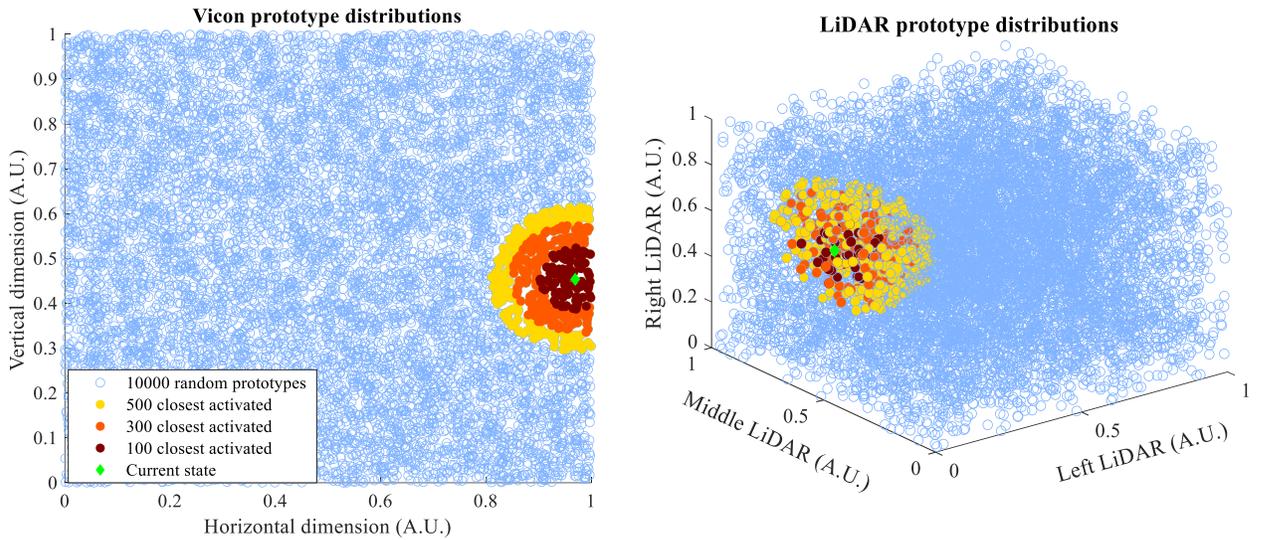


Figure 3.3 Vicor and LiDAR prototype distributions for a time step during the experiments. The 500 closest prototypes contained the 300 and 100 closest prototypes. The 100 closest prototypes were also included in the 300 closest ones.

3.2.7 Statistical Analysis

Statistical comparison between the required number of switches using the adaptive strategy and using the best possible fixed list at each round (Static scenarios) or each setup (Dynamic scenario) was performed using non-parametric Friedman's repeated measures analysis of variance (ANOVA) with Bonferroni correction across 2 strategies (adaptive and the best fixed list) and 3 sets (IBM SPSS, Build 1.0.0. 1447; IBM Corp., Armonk, N.Y., USA). As a reminder,

each set of experiments was performed on a different day and the learned weights were transferred from one set to the next. The best possible fixed list was determined for each experiment post-hoc, from the 120 possible orders, as the one that showed the lowest number of required switches (5 modes can be ordered in a switching list in $5!=120$ different orders). Pairwise comparisons were performed using Wilcoxon rank-sum test for each user separately and comparisons with $p \leq 0.05$ were considered significant.

The total number of required switches under each of the three experimental scenarios were averaged across all users. A t-test was used to compare the total number of required switches using the adaptive strategy and the best possible fixed list for each scenario. Comparisons with $p \leq 0.05$ were considered significant.

3.3 Results

An example of GVF prediction values during the experiments in the Static-Learning and unlearning scenario is shown in Fig. 3.4. An excerpt of the GVF values for all modes is shown in Fig. 3.4a; for the times that the user initiated a switching action (vertical red-dashed lines), the next intended mode (filled area) had the highest prediction value among all other available modes, and began to rise before the switching action took place (the ongoing active mode was not considered in the comparisons as it was always ranked last in the list). In the case when Left turn (purple), for instance, was the desired mode after using the Fast mode (orange), its GVF value rose in advance and was the highest among all other modes, before the switching action took place. An example of GVF predictions at relevant switching instances for two modes in the Static-Learning and unlearning scenario is provided in Fig. 3.4b. The GVFs of the Normal speed mode used regularly in phase 1 increased (learned) and then decreased (unlearned) in phase 2 during which the Fast mode was used instead. The GVFs for the Fast mode increased (new

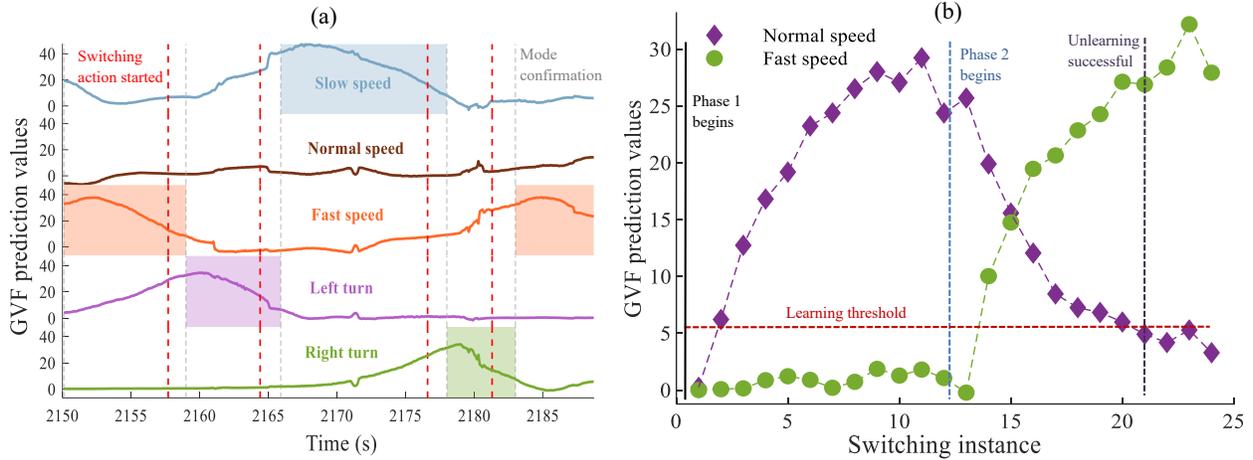


Figure 3.4 Examples of GVF predictions during the Static-Learning and unlearning scenario. (a) User u_1 traversing through the blue arrows in Fig. 2 (middle row). The vertical red dashed lines signify the initiation of switching instances. The switching list after initiation of each switching instance remained the same until a mode was confirmed (vertical grey lines). (b) User u_2 using the Fast mode during phase 2 after using Normal mode for the first 12 rounds (phase 1) during a specific path.

learning) during phase 2. After 4 times of using the Fast speed mode instead of the Normal speed in phase 2, the Fast speed became the dominant mode and its GVF value was higher than that for the Normal speed. Also, after 8 times of not selecting the Normal speed in the instances that the system expected the Normal speed to be selected, its GVF values dropped below the learning threshold, indicating that the Normal speed was unlearned.

Samples of the learned weights (user u_2 , Left turn) at the end of the experiments for all three scenarios are shown in Fig. 3.5. The Static scenarios showed generally higher values for the Vicon-related weights, while the Dynamic scenario emphasized more the LiDAR-related weights. This was aligned with our expectations, as the Static scenarios were more dependent on the position of the users, but the Dynamic scenarios were completely LiDAR-dependant and independent of the 2D positions of the user. The gradual decrease seen in the weight values is due to using different m values in the SKC algorithm (Section II-F), because the associated weight values change in accordance to the m values.

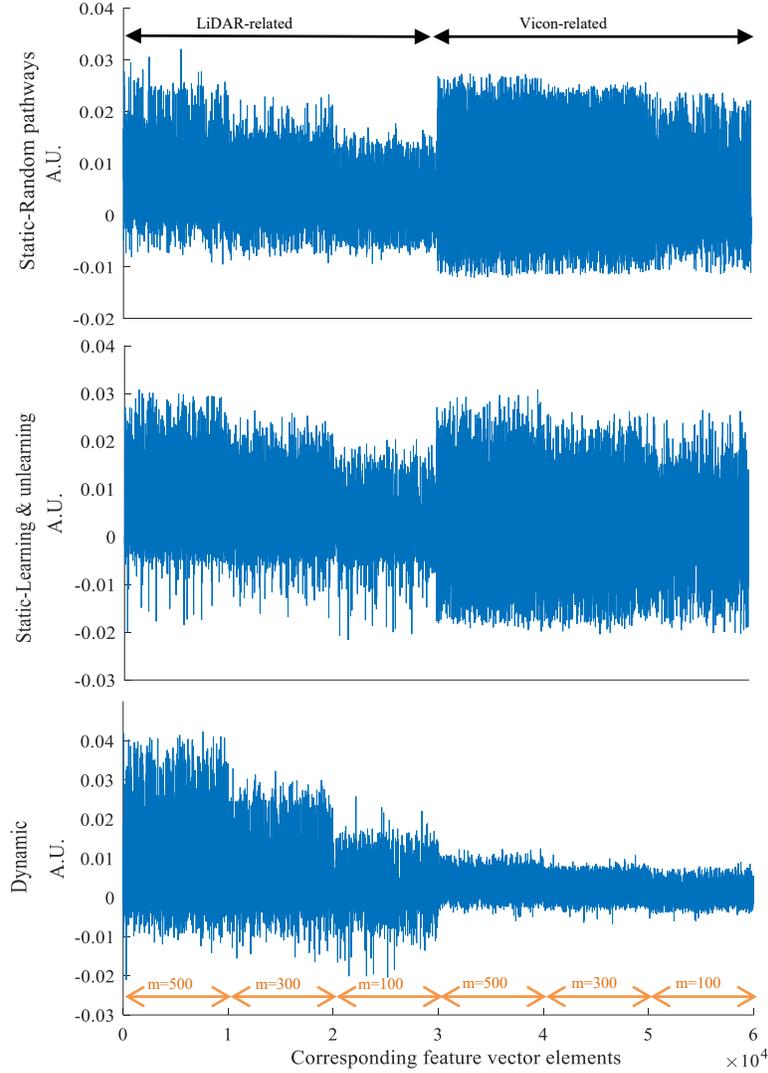


Figure 3.5 Weight values for the 2nd user (u_2) experiments. First row shows the weights for the Static-Random pathways scenario while the second and third rows show values for the Static-Learning and unlearning and Dynamic scenarios, respectively. The first 30,000 elements are LiDAR-related weights while the second 30,000 elements are Vicon-related weights. The first 10,000 elements of the LiDAR and Vicon groups correspond to $m=500$ active elements in the SKC function approximation method. The 2nd 10,000 elements correspond to $m=300$ active elements and the last 10,000 elements correspond to $m=100$ active elements of the SKC method.

To verify that the GVF signals were learned and predicted correctly, their true (ideal) return was calculated post-hoc. The true return G_t at each time step t is defined as the discounted sum of cumulants from $t + 1$ to the termination state, T , using the following equation: $G_t =$

$$\sum_{k=0}^T \gamma^k c_{t+k+1},$$

where γ is the discounting factor and c is the cumulant, as described in Section

II-E. GVF predictions were compared to their ideal return by calculating the mean binned

absolute error between them for all three users. An example of this comparison for the Slow mode in the Static-Learning and unlearning scenario can be seen in Fig. 3.6a. For all users, the mean error continually decreased to the end of phase 1, increased noticeably by the change in the modes' utilization at the beginning of phase 2, and decreased again afterwards. Moreover, comparing the mean error in the first bin of phase 1 and the bin following phase 2 shows the effect of learning, as the latter had larger error, meaning that it had to learn new values for weights while also trying to overcome the values learned during phase 1. Fig. 3.6b shows the prediction signals, their true return and the raw activity level (cumulant values, $c_j=1$ for the current active mode and $c_j=0$ for other modes) for the bins at the beginning of phase 1 (bin 1), before phase 2 (bin 5 for the 1st user; bin 6 for the 2nd and 3rd users), just after phase 2 (bin 6 for the 1st user; bin 7 for the 2nd and 3rd users), and the bin at the end of phase 2 (bin 11). In addition to the alignment of the prediction signals with their true return, both rose in advance of the cumulant raw activity level. This is important because the early rise placed the desired modes on top of the switching list, before the user hit the switch button.

The number of required switches at each switching instance using the adaptive strategy is shown in Fig. 3.7a for all users and all experimental scenarios. The sum of potential learning system errors in each round (Static scenarios) or each setup (Dynamic scenario) is plotted in Fig. 3.7b. Potential error was defined as the instances where the system did not present a desired mode (or two modes) at the top of the switching list. A potential error was calculated as the number of switching actions minus 1, except for the instances where more than one mode could be presented to the user; e.g., when facing a front-only obstacle where both turning Left and Right are rational suggestions based on the user's previous activities. In these exception instances, two required switches also resulted in a zero potential error if the two first modes in the list were the

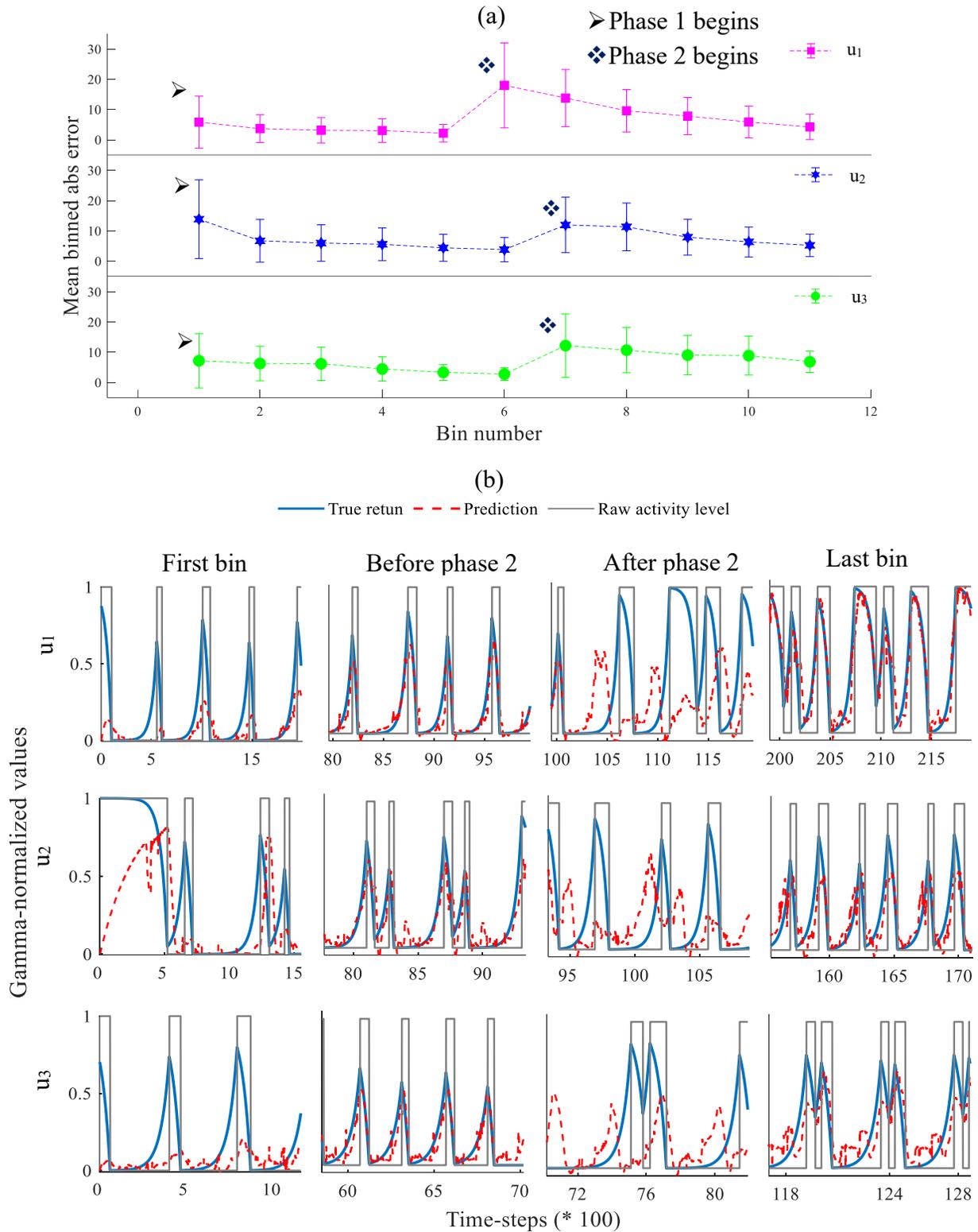


Figure 3.6 A comparison between the prediction values, their true return and the raw activity levels (cumulants) for the Static-Learning and unlearning scenario. (a) The absolute errors between the prediction values and their true return (for the Slow mode, as an example) were divided into 11 bins for each user (u) and the mean values along with the standard deviations of each bin were plotted. The number of time steps at each bin were determined as 1990, 1555, 1170 for users u_1 , u_2 and u_3 , respectively. These bin numbers were chosen to best separate the occurrence of phase 2 from phase 1. (b) The prediction, true return and raw activity values plotted for four different bins.

expected modes. For instances where the first mode in the list was chosen by the user (one

required switch) but there were two possible modes to switch to, one potential error was recorded for that instance if the second mode in the list was not the expected mode. Therefore, ideally after an initial early learning period, only one switch should have been required at each instance unless an uncertainty is present (the exception instances). For the Static-Random pathways scenario (Fig. 3.7b, left), after traversing the green and yellow arrows for the first three rounds (Fig. 3.2, top row), the potential errors were zero and only one switch was needed at the 3rd round for all users. For the next three rounds (4-6), a new set of arrows (blue & red, Fig 3.2, top row) were traversed. After an initial rise in the errors and in the number of switches, there were no errors and no more than two switches from the 6th round on for all users except for two instances where there was one switching error and three required switches for the 3rd user (u_3). The Static-Learning and unlearning scenario showed zero error and one required switch after an initial learning period during phase 1 (Fig. 3.7b, middle). However, it took the system between 3-5 rounds to replace the old fully learned modes in the switching list with the new modes in phase 2, while only 1-3 rounds were needed for the initial learning in phase 1 to become error-free. This was consistent with the finding in Fig. 3.6a (learning new values of weights while also trying to overcome the learned values during phase 1). No error was seen after that initial learning part of phase 2. In the Dynamic scenario, the number of switches and potential errors for Left turn and Right turn modes are plotted in Fig. 3.7, right column. They were the main modes of interest in this scenario as this scenario was specifically targeting the capabilities of the system in predicting Left and Right turns based on the LiDAR-provided information (Vicon-related weights were expected to gain lower weights, as observed in Fig. 3.5, third row). After the first set (first 40 switches, 10 setups, seen as the initial learning set; Section II-D), the system was able to generalize the turning behavior to all the obstacles regardless of their 2D position in

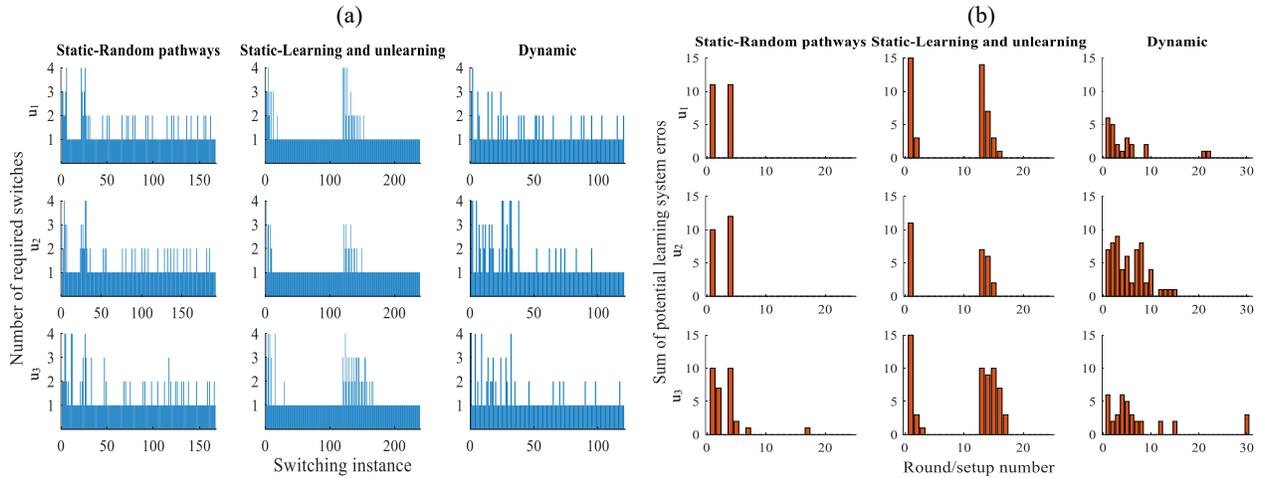


Figure 3.7 (a) Number of required switching actions at each switching instance using the adaptive strategy for all scenarios and all users. (b) Sum of potential system errors using the adaptive strategy for each round (Static scenarios) or setup (Dynamic scenario) for all scenarios and all users.

the environment, where the 2nd and 3rd sets showed an error only in 2-4 instances (out of 80) and no more than 2 switches were required at each instance of the 2nd and 3rd sets.

The average number of required switches for each round (Static scenarios) or each setup (Dynamic scenario) was compared for all 3 sets (3 consecutive sets for each scenario, each performed on a different day), under the adaptive and the best possible fixed list strategies. The results of the comparison are depicted in Fig. 3.8. For the Static-Random pathways scenario, the adaptive strategy required significantly lower number of switches at each round during the 2nd and 3rd sets for all 3 users. The Static-learning and unlearning scenario also showed a significantly decreased number of switches for the adaptive strategy during set 1 (for 2nd and 3rd users) and sets 2 and 3 (for all 3 users). For the Dynamic scenario, a significantly decreased number of switches for the adaptive strategy during set 1 (for 1st and 3rd users) and sets 2 and 3 (for all 3 users) was seen. Moreover, the sets under the adaptive strategy itself showed a significantly lower number in set 2 (2nd and 3rd users) and set 3 (all 3 users) in comparison to set 1, which demonstrates the effect of learning.

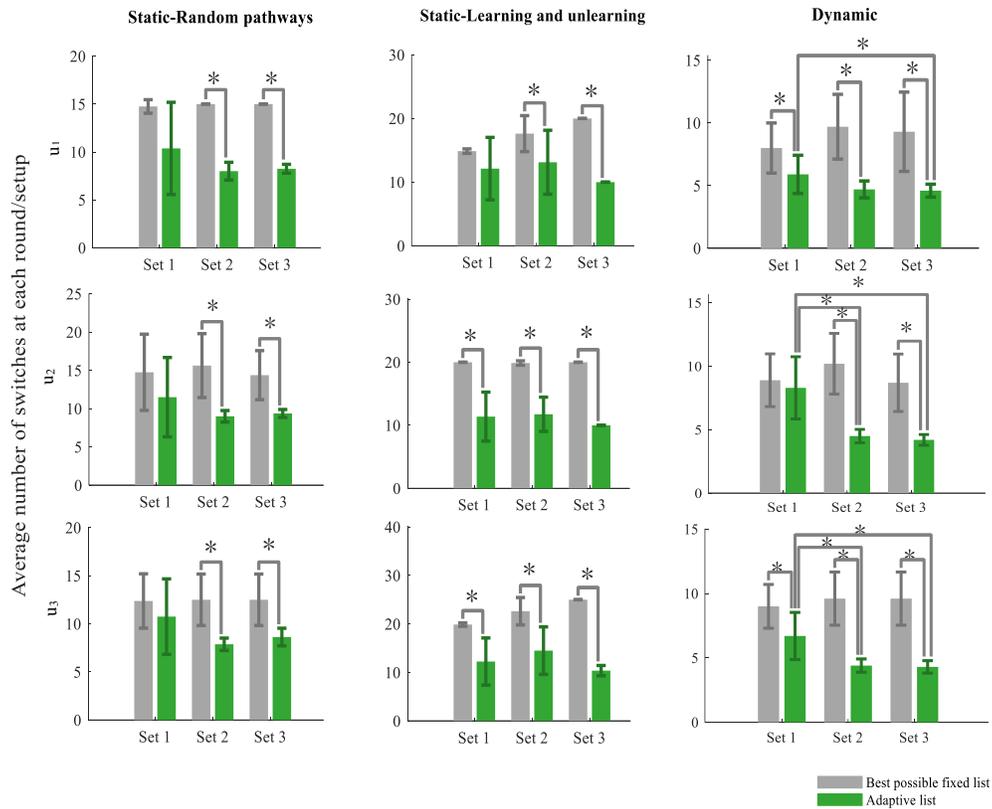


Figure 3.8 Statistical comparison (2 x 3 repeated measures analysis of variance) between the average number of required switches based on a best possible fixed list (grey; mean \pm SD) and the adaptive list (green; mean \pm SD) for all experimental scenarios. Left column: Static-Random pathways scenario for all users (u_1 - u_3). Middle column: Static-Learning and unlearning scenario for all users. Right column: Dynamic scenario for all users. The number of required switches were averaged for all rounds (Static scenarios, 8 rounds at each set) and for all setups (Dynamic scenario, 10 setups for each set) at each set. * $p < 0.05$.

A comparison between the average of total number of required switches in each experimental scenario across all users under the adaptive and the best possible lists is shown in Fig. 3.9. For all scenarios, the total number of switches was significantly lower under the adaptive strategy ($p = 0.025, 0.024, 0.003$ for the Static-Random pathways, Static-Learning and unlearning and Dynamic, respectively). Moreover, the number of switching hits under the adaptive list was close to the minimal possible number of switches (considering one switch for each switching instance; horizontal lines in Fig. 3.9), demonstrating the success of the proposed method.

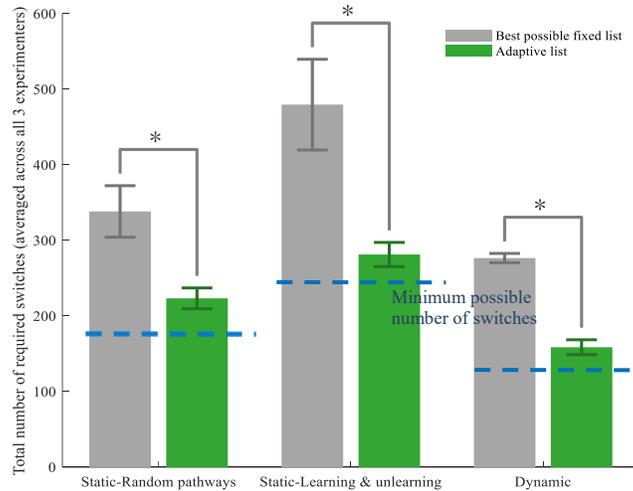


Figure 3.9 Statistical comparison (t-test) between the total number of required switches for each experimental scenario using the best possible fixed list (grey) and the adaptive list (green), averaged across all three users (mean \pm SD). The dashed horizontal blue lines show the minimal possible number of switching hits for each scenario. * $p < 0.05$.

3.4 Discussion and Conclusions

This study represents an important contribution to the overall goal of restoring walking after paralysis by making it possible to take users' personalized intentions into account during the control of lower-limb exoskeletons. This promises to make control of future exoskeletons easier and less tedious. Using the adaptive strategy, the users' preferences were, for the first time, learned online (continually during ongoing experience) and presented to them in order of expected relevance during their manual control interactions. Under the adaptive strategy, the total number of required switches were significantly lower, and reduced by nearly 50% in comparison to the best traditional non-adaptive method. The system was generalizable across different scenarios and was precise in predicting the users' needs at different occasions, where randomness or uncertainties were present. No pre-training was needed and the system updated its decisions by interacting with the users during walking.

Importantly, the proposed system, unlike many high-level controllers, would not force the users to select a mode; instead, it enhances its response to their intention by proposing a walking

mode. The users always would have the authority to switch again and select the next mode(s) in the switching list. The system continually corrected its predictions to adapt to its users' walking behaviors. We expect that the adaptive strategy contributed in this work can benefit persons with mobility impairments due to conditions such as SCI (either complete or incomplete), stroke or multiple sclerosis who require aid when walking.

The findings are consistent with those in upper limb prostheses, where the control interface for moving 4 joints (shoulder, elbow, wrist and hand) was improved using real-time adaptive switching [144], [145], and a significant decrease in the total number of required switches was seen in comparison to the non-adaptive method. This study substantially advanced the work in the previous studies by enhancing the function approximator, taking environmental information into account and testing in more advanced and less predictable (Random pathways and Dynamic) scenarios.

This study had a few limitations. 1) Only 5 walking modes were used; however, the system theoretically can anticipate an infinite number of GVFs (and operating modes accordingly) and rearrange the associated modes in the switching list, as each GVF is calculated independently. 2) The LiDAR sensors received noiseless signals in the lab environment; additional filtering will be necessary in outdoor environments. 3) A Vicon motion capture system was used for determining the users' locations in the lab; this can be replaced by high precision GPS systems in home or outdoor environments. 4) Testing was performed on only three users (experimenters); nonetheless, the consistency in learning across these users suggests that the same will be accomplished across all potential users of lower-limb exoskeletons.

Designing an autonomous strategy which eliminates the need for even a single switching action will be the focus of future work. Moreover, recruiting participants with walking impairments and conducting qualitative studies on the mental load experienced by the participants when using the adaptive strategy is also an essential step towards translating this system to real-world applications.

Chapter 4: General Discussion

4.1 Summary and Significance

4.1.1 Thesis Summary and Significance

The overall goal of this work was to design an intelligent method to reduce the cognitive and physical burdens associated with walking using lower-limb exoskeletons after paralysis.

Specifically, the work outlined in this thesis demonstrates the first use of a biologically inspired predictive approach, called General Value Functions (GVFs) [90], in an applied setting for predicting the lower-limb exoskeletons users' intention in switching between walking modes.

GVFs are value functions with the ability of predicting unlimited non-reward signals over a window in the future. The prediction-driven behavior of the GVFs is inspired from the role of the cerebellum in the brain by making predictions and verifying those predictions against actual sensory data [93]–[95]. Accurate and proper predictions will enhance the ability to control the upcoming situation and prepare an appropriate response [58]. GVFs were used in this work to produce predictive knowledge that can benefit the exoskeleton users by sharing the control burden between the user and the device. State-of-the-art lower-limb exoskeletons offer various walking/operating modes than can be chosen by their users. However, determining an efficient way to choose or switch between these incorporated modes has been identified as a major problem among the end users of robotic devices [57], [138]. To this end, a GVF-based strategy was designed and tested across users (experimenters) walking with the exoskeleton. The developed strategy, called adaptive switching controller, learned the users' preferences in choosing modes at specific states, adapted to the users' walking patterns and behaviours, and proposed in advance the most likely mode(s) that the users may select next. This resulted in a significantly lower number of required switching actions on the part of the users by using a

switch button, in comparison to manual switching between modes through a pre-ordered list. Figure 4.1 shows a comparison of the traditional non-adaptive switching list and the adaptive switching list used in this study. Real-time prediction learning and adaptation were applied in this work, for the first time, to enhance the control interface of a lower-limb exoskeleton during uninterrupted and continuous use. The adaptive switching method developed in this work is expected to be used to benefit persons with mobility impairments who require assistance when walking. This work resulted in a ground breaking accomplishment and a promising deliverable that is geared towards real-world application of the smart lower-limb exoskeleton robots.

4.1.2 A Machine-learned System for Lower-limb Exoskeletons' Control – Offline Verification

The goal of the work in chapter 2 was to assess the capabilities of a machine-learned adaptive algorithm in predicting the most likely walking modes chosen by a user in an offline setup [147]. As the developed learning system controlled the general motion of the exoskeleton (high-level control), the usability of the method had to be verified first in an offline setup. This helped to debug the learning algorithms and provide a safe experimental environment for the users. A user

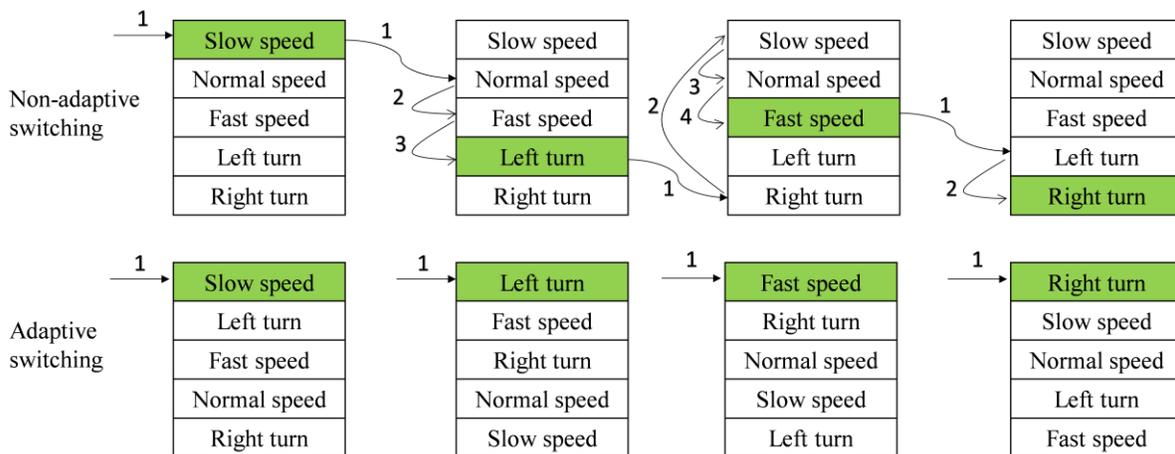


Figure 4.1 A comparison between the traditional non-adaptive switching method (top row) and the adaptive switching method (bottom row). The desired walking mode at each switching instance is highlighted as green. It can be seen that the adaptive method reorders the switching list to optimize the number of required switches (shown by numbers on top of each arrow) at each switching instance.

(experimenter) walked with the lower-limb exoskeleton in different experimental scenarios using a switching panel. Five walking modes (three walking speeds and turning left/right) were available to switch to, and a separate switch button was allocated for each walking mode. Three LiDAR sensors mounted on three sides of a walker recorded the distance of the user to the objects, located in the experimental environment. Position of the user was determined mathematically in a 2D plane post-hoc using a reference coordinate system and the time spent in each mode along with the speed of that mode. The LiDAR and position-related information in addition to the user's switching choices at each time step were passed to the designed learning system post-hoc. The system showed its potential in learning and adapting to the user preferences, predicting the next mode that was chosen by the user. The system was able to re-order a switching list containing the mentioned five modes based on the likelihood of the modes being used next (e.g., the mode with the highest prediction value of being used next was ranked first). Comparing the machine-learned adaptive method to the best possible non-adaptive strategy (fixed list) showed an average of 42.44% reduction in the total number of required switching actions under the adaptive strategy. This work formed the framework for the real-time, online application of this system using a single button for switching from one mode to a different mode.

4.1.3 A Machine-learned System for the Control of Lower-limb Exoskeletons – Real-time Verification

Chapter 3 aimed to assess the online capabilities of the designed GVF-based adaptive system in predicting users' future selection of walking modes in real-time, when walking with a lower-limb exoskeleton. Three neurologically-intact users (experimenters) walked with the exoskeleton each in three unique experimental scenarios, mimicking real-world situations. Five walking modes were pre-ordered in a switching list, where the order of the modes in the list was updated at each time step, based on their likelihood of being used next. One switch button was allocated for

switching between modes. Using Selective Kanerva Coding (SKC) [146], the system decoded the LiDAR-provided information on the objects inside the experimental environment and the Vicon motion capture system-provided information on users' position. Using GVs and temporal-difference learning (TD) [148], the system was able to 1) learn the users' preferences at each specific state, 2) unlearn the mode selections that were not preferred anymore and 3) generalize across similar situations, while discriminating the choices across separated situations. Overall, an adaptive controller was designed with the capability of adapting to the users' walking patterns and choices. The modes were suggested to the users according to their ranking in the adaptive switching list. Users had the authority to either select the suggested mode or switch to the next mode(s) in the list. The adaptive controller did not require any previous training dataset and was able to quickly adapt to the users' behavior. The adaptive controller did not force the users to select a mode, but it enhanced its response to their intention by proposing a walking mode that it predicted the users wanted to switch to. This study showed that it is possible to both learn and unlearn users' switching behaviors in real-time. In practice, both humans and machines can make mistakes, or humans can change their approach after a while. In the case of exoskeletons, users can get used to the device and tend to walk with higher speeds or the rehabilitation regime can change after some observations. As a result, it was important to show that not only the system can learn behaviors, but also it can allow for a learned behavior to be unlearned. This project combined the advanced function approximation, prediction and learning methods with the lower-limb exoskeletons' technology to propose a personalized approach to restore walking after paralysis and make the control of lower-limb exoskeletons one step smarter and closer to their utilization in real-world communities.

4.2 Limitations

The experiments were conducted in the laboratory setup, where LiDAR sensors provided noiseless and accurate signals in short distances (less than 10 meters). In outdoor settings, more filtering will be required. Moreover, position of the users was determined with high accuracy using 8 Vicon motion capture system cameras. In both outdoor or indoor settings, high precision GPS systems can take the place of the cameras.

The machine learning controller was developed and tested in MATLAB and sent the selected mode to the Real-time Desktop Simulink environment of MATLAB to be commanded to the exoskeleton joints. The exoskeleton was connected to a computer with an Intel Core i9 CPU via USB. However, deploying these codes into the exoskeleton system processors can be challenging due to the limitations of compiling. Additional verifications and continuous maintenance based on the available resources would be required for translating a machine-learned prototype into production.

Five walking modes, consisting of three speeds and two turning modes were utilized in this study. However, the system theoretically has the ability to predict the utility of unlimited number of modes as each GVF prediction value is calculated independently. As a result, the system can be tested by adding other operating modes such as ascending/descending stairs, sit-to-stand and stand-to-sit transitions.

Moreover, the study utilized three neurologically-intact users (experimenter) for testing the system's effectiveness, due to the limitation on recruiting study participants and the end users of the exoskeletons. However, the three users' shared consistency in their learning pattern suggests that all potential users of lower-limb exoskeletons can benefit from the proposed strategy in the same way.

4.3 Future Directions

This thesis was a research and development study, containing an initial validation of an adaptive controller for lower-limb exoskeletons. While this work showed the effectiveness of the adaptive controller in terms of reducing the number of required switching actions, a qualitative study should follow. In addition to adapting the system to the users, the amount of users' adaptation to the system should also be investigated. This can be achieved by 1) recruiting a large number of study participants with different neurological conditions affecting their walking abilities, and 2) conducting a subjective mental workload assessment of the participants while using both the adaptive and the non-adaptive traditional controllers. It is important to know whether the proposed approach and the fact that the modes are being re-ordered in the switching list adds or reduces the cognitive load and whether they can see the proposed approach as an applicable method for their daily usage.

Moreover, while the adaptive system under ideal situations requires only a single switching action to switch between modes, introducing autonomous transitions can be seen as an enhancement [61]. Autonomous controller uses GVF's in real-time to predict when is the right time to switch, based on users' previous activities. As the autonomous controller tries to optimize a policy of switching, rather than the policy of using an operating mode, it requires another realm of TD learning methods, called off-policy gradient TD learning [61], [149].

If a function approximation technique like SKC is employed, as it did in this work, the addition of sensors (e.g., more LiDAR sensors, eye gaze trackers) is possible since it is unaffected by the exponential rise in dimensions that regular tile coding suffers from [82]. While the LiDAR sensors in this study were able to detect objects with a certain minimum amplitude, addition of sensors facing the ground in front of the users to detect ground objects seems important. It can

help the system to identify and prepare for a desired response (e.g., increasing the step amplitude to bypass the object) when facing those objects. Eye gaze trackers can also be an interesting addition to the system. The system can track the users' attention and predict a response accordingly.

Finally, the adaptive switching technique developed in this work can be translated to exoskeletons in other fields as well. The exoskeletons used in industrial applications, both for upper and lower-limb assistance, can benefit from this optimized human-robot interaction. Supernumerary robotic limbs can also be equipped with the adaptive and eventually autonomous switching controllers to achieve their ultimate goal of enhancing human capabilities.

4.4 Links to Neuroscience

The learning methods used in this study were inspired from and/or linked to the functions of the nervous systems in animals and humans. These connections are summarized below.

4.4.1 Dopamine and TD Error

Nearly every area of the brain exhibits neural activity linked to reward processing. In RL algorithms, the reward, R_t , summarizes the main problem that the RL agent is trying to solve. We might think of that in brain as the result of multiple systems that are related to pleasure or punishment. These systems generate neural signals that contribute to the overall perception of reward [81].

The largest similarity between this work and systems neuroscience lies in the TD error correction behavior used in this work. Dopamine is one of the neurotransmitters crucial for reward processing in the brain, with the cell bodies mainly residing in the ventral tegmental area (VTA) and the substantia nigra pars compacta of the midbrain. According to the “reward prediction

error hypothesis of dopamine neuron activity,” one of the purposes of the phasic activity of dopamine-producing neurons in the midbrain of animals is to transmit an error between a previous and a present estimate of projected future reward to specific brain regions [81], [150]. In this regard, TD error formation is highly similar to the phasic activity of dopamine-producing neurons. Sutton and Barto in [81] identified 4 main similarities between these two concepts according to the experiments conducted by Schultz’s group [151], [152]. The 4 similarities and their connections with the findings in this thesis work are summarized below:

1) A dopaminergic neuron only exhibits the phasic response when a rewarding experience is unexpected (not fully predicted). This is the same as the procedure in TD algorithm; a TD error ($\delta_t = C_{t+1} + \gamma V(S_{t+1}) - V(S_t)$) is non-zero only when the GVF prediction ($V(S_{t+1})$) is not fully predicted, regardless of the reward being received (or in this study, the cumulant C being either 0 or 1). Meaning that if $V(S_{t+1}) = V(S_t) = \frac{C_{t+1}}{1-\gamma}$, then the TD error δ_t is zero and no additional update will occur. In this study, cumulants were either 0 or 1. In case of a cumulant being 0, then $V(S_{t+1}) = V(S_t) = 0$ means that the system has not yet experienced the utilization of the cumulant (walking mode) whose GVF is in question. Therefore, it is predicting “undoubtedly” that the considered cumulant (walking mode) is not going to be used in the next time step. This was seen in the GVF values during early learning, for the walking modes that were not utilized yet. In case of a cumulant being 1, then $V(S_{t+1}) = V(S_t) = \frac{1}{1-\gamma}$ shows that the GVF prediction has been saturated to its maximum value. The system is again predicating “without any uncertainty” that the current active mode is being used in the next time step as well. There would be no update (phasic dopaminergic response) in both cases. Figure 4.2 shows an example of an unutilized GVF cumulant (mode) remaining in 0, and a utilized GVF cumulant

(mode) which is saturated to its maximum value ($\frac{1}{1-0.98}=50$) during a post-hoc test using the learning parameters in this study ($\gamma = 0.98, \alpha=0.0001$).

2) Neutral cues that come before a reward initially do not trigger significant phasic dopamine responses, but as learning progresses, these signals acquire predictive significance and start to trigger phasic dopaminergic responses. This was the same as the observation in this work, where, as an example, Left/Right turn GVFs were not triggered (did not rise) during early learning while facing an obstacle (the cue) and before the users selected those modes (the reward), but facing the obstacles gained predictive significance during continual online learning using the TD update rule, and made the Left/Right turn GVFs rise in advance of the users' switching action.

3) If an even earlier stimulus comes before the one that has already developed predictive knowledge, the phasic dopaminergic response switches to the earlier cue and stops for the later cue. This parallels the bootstrapping effect in TD learning, where update priority is given to an earlier predictive stimulus over a later predictive stimulus (closer rewards/cumulants to the current time step will have a larger effect on the return value than the later rewards/cumulants).

4) After learning, a dopaminergic neuron's response falls below its baseline level shortly if the predicted rewarding event is omitted. This was consistent with the Learning and unlearning

scenario results, where the GVF of modes in phase 1 dropped below the learning threshold (became unlearned) after several rounds of not being used ($C=0$) due to a negative TD error.

4.4.2 Cerebellum and General Value Functions

The prediction task in this study was to predict which walking mode of the exoskeleton a user will want to select next. This question was answered by using GVFs and updating their associated learning weights using TD learning. Using GVFs to predict future human-machine interactions is parallel with the brain making “forward” motor predictions of its own [96], [104]. Motor learning entails behavioral adaptations of the brain (or the body itself) in controlling the body due to interactions with the environment, especially when there are unexpected changes in the tasks or the environment, and can be distinct from maturation and innate motor behaviors. As these changes are unpredictable, a pre-specified control system is unable to respond accordingly and therefore, flexibility in the control system is needed [96]. In this regards, the cerebellum is hypothesized to predict the outcomes of a motor or cognitive command that comes from the

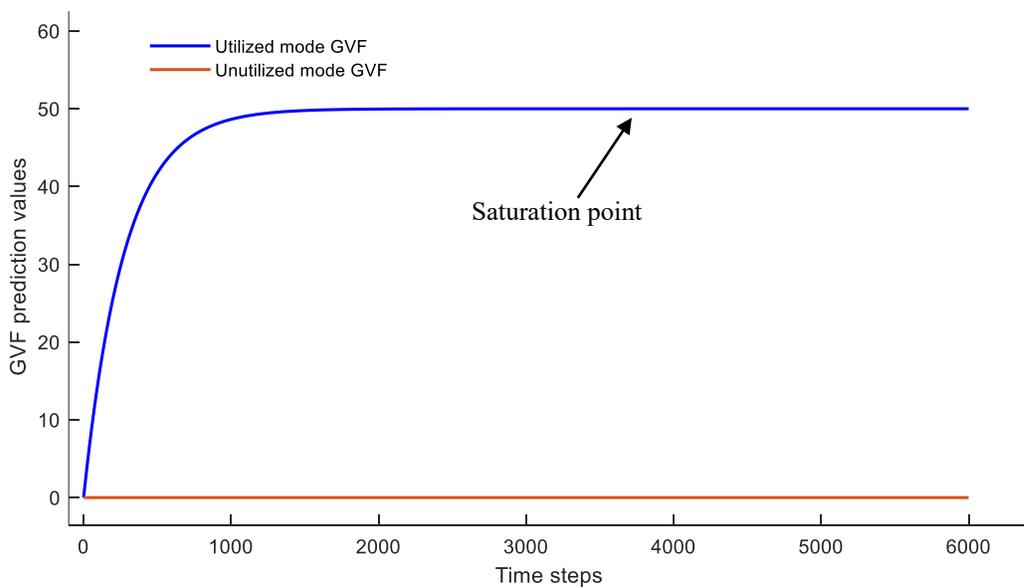


Figure 4.2 GVF values for 2 arbitrary cumulants in a post-hoc test. One mode (blue) was being activated ($C=1$) for the duration of the test (6000 time steps), while the other mode (red) was not utilized ($C=0$) at all. The activated mode reached to its maximum GVF value of 50 and remained at that level after 3830 time steps. The unutilized mode GVF remained zero during the whole duration of the test.

cerebral cortex in order to get the musculoskeletal system ready to handle ongoing changes [100]. Morton and Bastian [153] showed that damage to the cerebellum reduces the ability to adapt to predictable (feedforward) tasks but not to the unpredictable changes that require reactive control. This is aligned with forward internal models [94], [95], stating that the cerebellum predicts and modifies the sensory results of motor commands and participates in computing sensory prediction errors by evaluating the predictions against the sensory feedback. Use of GVFs in making forward predictions of the state of the system is parallel with these findings. Specifically, in this work, GVFs were used to provide a predictive response to the question of “which walking mode, among five walking modes, an exoskeleton user wants to select next?”

Overall, GVFs tackle one of the most difficult issues in AI/ML: how does an agent create a picture/representation of the world based on its own experience? This has been studied for more than a decade [84], [101] and the underlying motivations and correlations to the field of Neuroscience have yet to be completely uncovered.

Bibliography

- [1] P. Assinck *et al.*, “Myelinogenic Plasticity of Oligodendrocyte Precursor Cells following Spinal Cord Contusion Injury,” *J. Neurosci.*, vol. 37, no. 36, pp. 8635–8654, Sep. 2017, doi: 10.1523/JNEUROSCI.2409-16.2017.
- [2] R. Jian *et al.*, “Repair of spinal cord injury by chitosan scaffold with glioma ECM and SB216763 implantation in adult rats: Effect of Scaffold on the Differentiation of Neural Stem Cell,” *J. Biomed. Mater. Res.*, vol. 103, no. 10, pp. 3259–3272, Oct. 2015, doi: 10.1002/jbm.a.35466.
- [3] Y.-H. Kim, K.-Y. Ha, and S.-I. Kim, “Spinal Cord Injury and Related Clinical Trials,” *Clin Orthop Surg*, vol. 9, no. 1, p. 1, 2017, doi: 10.4055/cios.2017.9.1.1.
- [4] G. Renzenbrink, J. Buurke, A. Nene, A. Geurts, G. Kwakkel, and J. Rietman, “Improving walking capacity by surgical correction of equinovarus foot deformity in adult patients with stroke or traumatic brain injury: A systematic review,” *J Rehabil Med*, vol. 44, no. 8, pp. 614–623, 2012, doi: 10.2340/16501977-1012.
- [5] G.-H. Wang *et al.*, “Free-Radical Scavenger Edaravone Treatment Confers Neuroprotection Against Traumatic Brain Injury in Rats,” *Journal of Neurotrauma*, vol. 28, no. 10, pp. 2123–2134, Oct. 2011, doi: 10.1089/neu.2011.1939.
- [6] R. Saigal, C. Renzi, and V. K. Mushahwar, “Intraspinal microstimulation generates functional movements after spinal-cord injury,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 12, no. 4, pp. 430–440, Dec. 2004, doi: 10.1109/TNSRE.2004.837754.
- [7] U. S. Hofstoetter *et al.*, “Augmentation of Voluntary Locomotor Activity by Transcutaneous Spinal Cord Stimulation in Motor-Incomplete Spinal Cord-Injured Individuals: Augmentation of Locomotion by tSCS in Incomplete SCI,” *Artificial Organs*, vol. 39, no. 10, pp. E176–E186, Oct. 2015, doi: 10.1111/aor.12615.
- [8] F. B. Wagner *et al.*, “Targeted neurotechnology restores walking in humans with spinal cord injury,” *Nature*, vol. 563, no. 7729, pp. 65–71, Nov. 2018, doi: 10.1038/s41586-018-0649-2.
- [9] S. Mohammed, Y. Amirat, and H. Rifai, “Lower-Limb Movement Assistance through Wearable Robots: State of the Art and Challenges,” *Advanced Robotics*, vol. 26, no. 1–2, pp. 1–22, Jan. 2012, doi: 10.1163/016918611X607356.
- [10] R. Baud, A. R. Manzoori, A. Ijspeert, and M. Bouri, “Review of control strategies for lower-limb exoskeletons to assist gait,” *J NeuroEngineering Rehabil*, vol. 18, no. 1, p. 119, Dec. 2021, doi: 10.1186/s12984-021-00906-3.
- [11] A. Esquenazi, M. Talaty, and A. Jayaraman, “Powered Exoskeletons for Walking Assistance in Persons with Central Nervous System Injuries: A Narrative Review,” *PM&R*, vol. 9, no. 1, pp. 46–62, Jan. 2017, doi: 10.1016/j.pmrj.2016.07.534.
- [12] L. Zhou, W. Chen, J. Wang, S. Bai, H. Yu, and Y. Zhang, “A Novel Precision Measuring Parallel Mechanism for the Closed-Loop Control of a Biologically Inspired Lower Limb Exoskeleton,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 6, pp. 2693–2703, Dec. 2018, doi: 10.1109/TMECH.2018.2872011.
- [13] H. Quintero, R. Farris, C. Hartigan, I. Clesson, and M. Goldfarb, “A Powered Lower Limb Orthosis for Providing Legged Mobility in Paraplegic Individuals,” *Topics in Spinal Cord Injury Rehabilitation*, vol. 17, no. 1, pp. 25–33, Jul. 2011, doi: 10.1310/sci1701-25.

- [14] S. Jezernik, G. Colombo, and M. Morari, “Automatic Gait-Pattern Adaptation Algorithms for Rehabilitation With a 4-DOF Robotic Orthosis,” *IEEE Trans. Robot. Automat.*, vol. 20, no. 3, pp. 574–582, Jun. 2004, doi: 10.1109/TRA.2004.825515.
- [15] G. Zeilig, H. Weingarden, M. Zwecker, I. Dudkiewicz, A. Bloch, and A. Esquenazi, “Safety and tolerance of the ReWalk™ exoskeleton suit for ambulation by people with complete spinal cord injury: A pilot study,” *The Journal of Spinal Cord Medicine*, vol. 35, no. 2, pp. 96–101, Mar. 2012, doi: 10.1179/2045772312Y.0000000003.
- [16] K. A. Inkol and J. McPhee, “Assessing Control of Fixed-Support Balance Recovery in Wearable Lower-Limb Exoskeletons Using Multibody Dynamic Modelling,” in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, New York City, NY, USA, Nov. 2020, pp. 54–60, doi: 10.1109/BioRob49111.2020.9224430 [Online]. Available: <https://ieeexplore.ieee.org/document/9224430/>. [Accessed: Jun. 29, 2022]
- [17] H. Kawamoto *et al.*, “Voluntary motion support control of Robot Suit HAL triggered by bioelectrical signal for hemiplegia,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, Buenos Aires, Aug. 2010, pp. 462–466, doi: 10.1109/IEMBS.2010.5626191 [Online]. Available: <http://ieeexplore.ieee.org/document/5626191/>. [Accessed: Jun. 29, 2022]
- [18] A. Rodríguez-Fernández, J. Lobo-Prat, and J. M. Font-Llagunes, “Systematic review on wearable lower-limb exoskeletons for gait training in neuromuscular impairments,” *J NeuroEngineering Rehabil*, vol. 18, no. 1, p. 22, Dec. 2021, doi: 10.1186/s12984-021-00815-5.
- [19] H. Kawamoto, T. Hayashi, T. Sakurai, K. Eguchi, and Y. Sankai, “Development of single leg version of HAL for hemiplegia,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, MN, Sep. 2009, pp. 5038–5043, doi: 10.1109/IEMBS.2009.5333698 [Online]. Available: <http://ieeexplore.ieee.org/document/5333698/>. [Accessed: Jun. 30, 2022]
- [20] T. Vouga, R. Baud, J. Fasola, M. Bouri, and H. Bleuler, “TWIICE — A lightweight lower-limb exoskeleton for complete paraplegics,” in *2017 International Conference on Rehabilitation Robotics (ICORR)*, London, Jul. 2017, pp. 1639–1645, doi: 10.1109/ICORR.2017.8009483 [Online]. Available: <https://ieeexplore.ieee.org/document/8009483/>. [Accessed: Jan. 14, 2022]
- [21] R. J. Farris, H. A. Quintero, and M. Goldfarb, “Preliminary Evaluation of a Powered Lower Limb Orthosis to Aid Walking in Paraplegic Individuals,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 6, pp. 652–659, Dec. 2011, doi: 10.1109/TNSRE.2011.2163083.
- [22] N. Birch *et al.*, “Results of the first interim analysis of the RAPPER II trial in patients with spinal cord injury: ambulation and functional exercise programs in the REX powered walking aid,” *J NeuroEngineering Rehabil*, vol. 14, no. 1, p. 60, Dec. 2017, doi: 10.1186/s12984-017-0274-6.
- [23] Y. Tanaka, S. Sakama, K. Nakano, and H. Kosodo, “Comparative Study on Dynamic Characteristics of Hydraulic, Pneumatic and Electric Motors,” in *ASME/BATH 2013 Symposium on Fluid Power and Motion Control*, Sarasota, Florida, USA, Oct. 2013, p. V001T01A037, doi: 10.1115/FPMC2013-4459 [Online]. Available: <https://asmedigitalcollection.asme.org/FPST/proceedings/FPMC2013/56086/Sarasota,%20Florida,%20USA/231642>. [Accessed: Jun. 30, 2022]

- [24] M. R. Tucker *et al.*, “Control strategies for active lower extremity prosthetics and orthotics: a review,” *J NeuroEngineering Rehabil*, vol. 12, no. 1, p. 1, 2015, doi: 10.1186/1743-0003-12-1.
- [25] T. Yan *et al.*, “A novel adaptive oscillators-based control for a powered multi-joint lower-limb orthosis,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Singapore, Singapore, Aug. 2015, pp. 386–391, doi: 10.1109/ICORR.2015.7281230 [Online]. Available: <http://ieeexplore.ieee.org/document/7281230/>. [Accessed: Jul. 04, 2022]
- [26] Keehong Seo, SeungYong Hyung, Byung Kwon Choi, Younbaek Lee, and Youngbo Shim, “A new adaptive frequency oscillator for gait assistance,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015, pp. 5565–5571, doi: 10.1109/ICRA.2015.7139977 [Online]. Available: <http://ieeexplore.ieee.org/document/7139977/>. [Accessed: Jul. 04, 2022]
- [27] I. Kang, P. Kunapuli, and A. J. Young, “Real-Time Neural Network-Based Gait Phase Estimation Using a Robotic Hip Exoskeleton,” *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 28–37, Feb. 2020, doi: 10.1109/TMRB.2019.2961749.
- [28] P. Beyl *et al.*, “Safe and Compliant Guidance by a Powered Knee Exoskeleton for Robot-Assisted Rehabilitation of Gait,” *Advanced Robotics*, vol. 25, no. 5, pp. 513–535, Jan. 2011, doi: 10.1163/016918611X558225.
- [29] J. Bae and M. Tomizuka, “A gait rehabilitation strategy inspired by an iterative learning algorithm,” *Mechatronics*, vol. 22, no. 2, pp. 213–221, Mar. 2012, doi: 10.1016/j.mechatronics.2012.01.009.
- [30] J. Zhang, C. C. Cheah, and S. H. Collins, “Torque Control in Legged Locomotion *
*Supplementary document of this chapter is located at https://www.andrew.cmu.edu/user/shc17/Zhang_2016_BLL—SuppMat.pdf,” in *Bioinspired Legged Locomotion*, Elsevier, 2017, pp. 347–400 [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/B9780128037669000075>. [Accessed: Jul. 04, 2022]
- [31] S. Oh and K. Kong, “High-Precision Robust Force Control of a Series Elastic Actuator,” *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 1, pp. 71–80, Feb. 2017, doi: 10.1109/TMECH.2016.2614503.
- [32] S. Qiu, W. Guo, D. Caldwell, and F. Chen, “Exoskeleton Online Learning and Estimation of Human Walking Intention Based on Dynamical Movement Primitives,” *IEEE Trans. Cogn. Dev. Syst.*, vol. 13, no. 1, pp. 67–79, Mar. 2021, doi: 10.1109/TCDS.2020.2968845.
- [33] X. Liu and Q. Wang, “Real-Time Locomotion Mode Recognition and Assistive Torque Control for Unilateral Knee Exoskeleton on Different Terrains,” *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 6, pp. 2722–2732, Dec. 2020, doi: 10.1109/TMECH.2020.2990668.
- [34] F. Xu, X. Lin, H. Cheng, R. Huang, and Q. Chen, “Adaptive stair-ascending and stair-descending strategies for powered lower limb exoskeleton,” in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, Takamatsu, Japan, Aug. 2017, pp. 1579–1584, doi: 10.1109/ICMA.2017.8016052 [Online]. Available: <http://ieeexplore.ieee.org/document/8016052/>. [Accessed: Jan. 14, 2022]
- [35] H. A. Varol, F. Sup, and M. Goldfarb, “Powered sit-to-stand and assistive stand-to-sit framework for a powered transfemoral prosthesis,” in *2009 IEEE International Conference on Rehabilitation Robotics*, Kyoto, Jun. 2009, pp. 645–651, doi:

- 10.1109/ICORR.2009.5209582 [Online]. Available: <http://ieeexplore.ieee.org/document/5209582/>. [Accessed: Jan. 14, 2022]
- [36] X. Tan, B. Zhang, G. Liu, X. Zhao, and Y. Zhao, “Cadence-Insensitive Soft Exoskeleton Design With Adaptive Gait State Detection and Iterative Force Control,” *IEEE Trans. Automat. Sci. Eng.*, pp. 1–14, 2021, doi: 10.1109/TASE.2021.3066403.
- [37] A. J. McDaid, Song Xing, and S. Q. Xie, “Brain controlled robotic exoskeleton for neurorehabilitation,” in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, NSW, Jul. 2013, pp. 1039–1044, doi: 10.1109/AIM.2013.6584231 [Online]. Available: <http://ieeexplore.ieee.org/document/6584231/>. [Accessed: Jan. 14, 2022]
- [38] J. Choi, K.-T. Kim, J. Lee, S. J. Lee, and H. Kim, “Robust Semi-synchronous BCI Controller for Brain-Actuated Exoskeleton System,” in *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, Gangwon, Korea (South), Feb. 2020, pp. 1–3, doi: 10.1109/BCI48061.2020.9061658 [Online]. Available: <https://ieeexplore.ieee.org/document/9061658/>. [Accessed: Jan. 14, 2022]
- [39] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, “High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton,” in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, Jul. 2013, pp. 5606–5609, doi: 10.1109/EMBC.2013.6610821 [Online]. Available: <http://ieeexplore.ieee.org/document/6610821/>. [Accessed: Jan. 14, 2022]
- [40] J. Choi and H. Kim, “Real-time Decoding of EEG Gait Intention for Controlling a Lower-limb Exoskeleton System,” in *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*, Gangwon, Korea (South), Feb. 2019, pp. 1–3, doi: 10.1109/IWW-BCI.2019.8737311 [Online]. Available: <https://ieeexplore.ieee.org/document/8737311/>. [Accessed: Jan. 14, 2022]
- [41] R. Baud, A. R. Manzoori, A. Ijspeert, and M. Bouri, “Review of control strategies for lower-limb exoskeletons to assist gait,” *J NeuroEngineering Rehabil*, vol. 18, no. 1, p. 119, Dec. 2021, doi: 10.1186/s12984-021-00906-3.
- [42] Y. He, D. Eguren, J. M. Azorín, R. G. Grossman, T. P. Luu, and J. L. Contreras-Vidal, “Brain-machine interfaces for controlling lower-limb powered robotic systems,” *J. Neural Eng.*, vol. 15, no. 2, p. 021004, Apr. 2018, doi: 10.1088/1741-2552/aaa8c0.
- [43] W. Svensson and U. Holmberg, “Ankle-Foot-Orthosis Control in Inclinations and Stairs,” in *2008 IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, Sep. 2008, pp. 301–306, doi: 10.1109/RAMECH.2008.4681479 [Online]. Available: <http://ieeexplore.ieee.org/document/4681479/>. [Accessed: Jan. 18, 2022]
- [44] M. Sharifi, J. K. Mehr, V. K. Mushahwar, and M. Tavakoli, “Adaptive CPG-Based Gait Planning With Learning-Based Torque Estimation and Control for Exoskeletons,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8261–8268, Oct. 2021, doi: 10.1109/LRA.2021.3105996.
- [45] S. C. Kirshblum *et al.*, “International standards for neurological classification of spinal cord injury (Revised 2011),” *The Journal of Spinal Cord Medicine*, vol. 34, no. 6, pp. 535–546, Nov. 2011, doi: 10.1179/204577211X13207446293695.
- [46] A. C. Villa-Parra *et al.*, “Control of a robotic knee exoskeleton for assistance and rehabilitation based on motion intention from sEMG,” *Res. Biomed. Eng.*, vol. 34, no. 3, pp. 198–210, Jul. 2018, doi: 10.1590/2446-4740.07417.

- [47] P. T. Chinimilli, S. C. Subramanian, S. Redkar, and T. Sugar, “Human Locomotion Assistance using Two-Dimensional Features Based Adaptive Oscillator,” in *2019 Wearable Robotics Association Conference (WearRAcon)*, Scottsdale, AZ, USA, Mar. 2019, pp. 92–98, doi: 10.1109/WEARRACON.2019.8719628 [Online]. Available: <https://ieeexplore.ieee.org/document/8719628/>. [Accessed: Jan. 18, 2022]
- [48] F. Cordella *et al.*, “Literature Review on Needs of Upper Limb Prosthesis Users,” *Front. Neurosci.*, vol. 10, May 2016, doi: 10.3389/fnins.2016.00209. [Online]. Available: <http://journal.frontiersin.org/Article/10.3389/fnins.2016.00209/abstract>. [Accessed: Feb. 09, 2022]
- [49] I. Farkhatdinov, N. Roehri, and E. Burdet, “Anticipatory detection of turning in humans for intuitive control of robotic mobility assistance,” *Bioinspir. Biomim.*, vol. 12, no. 5, p. 055004, Sep. 2017, doi: 10.1088/1748-3190/aa80ad.
- [50] C. Tefertiller *et al.*, “Initial Outcomes from a Multicenter Study Utilizing the Indego Powered Exoskeleton in Spinal Cord Injury,” *Topics in Spinal Cord Injury Rehabilitation*, vol. 24, no. 1, pp. 78–85, Jan. 2018, doi: 10.1310/sci17-00014.
- [51] M. Liu, D. Wang, and H. Huang, “Development of an Environment-Aware Locomotion Mode Recognition System for Powered Lower Limb Prostheses,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 4, pp. 434–443, Apr. 2016, doi: 10.1109/TNSRE.2015.2420539.
- [52] S. Carvalho, J. Figueiredo, and C. P. Santos, “Environment-Aware Locomotion Mode Transition Prediction System,” in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, Porto, Portugal, Apr. 2019, pp. 1–6, doi: 10.1109/ICARSC.2019.8733658 [Online]. Available: <https://ieeexplore.ieee.org/document/8733658/>. [Accessed: Jan. 16, 2022]
- [53] N. E. Krausz and L. J. Hargrove, “Recognition of ascending stairs from 2D images for control of powered lower limb prostheses,” in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, Montpellier, France, Apr. 2015, pp. 615–618, doi: 10.1109/NER.2015.7146698 [Online]. Available: <http://ieeexplore.ieee.org/document/7146698/>. [Accessed: Jan. 18, 2022]
- [54] B. Laschowski, W. McNally, A. Wong, and J. McPhee, “Preliminary Design of an Environment Recognition System for Controlling Robotic Lower-Limb Prostheses and Exoskeletons,” in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*, Toronto, ON, Canada, Jun. 2019, pp. 868–873, doi: 10.1109/ICORR.2019.8779540 [Online]. Available: <https://ieeexplore.ieee.org/document/8779540/>. [Accessed: Jan. 18, 2022]
- [55] S. S. Bhatlawande, J. Mukhopadhyay, and M. Mahadevappa, “Ultrasonic spectacles and waist-belt for visually impaired and blind person,” in *2012 National Conference on Communications (NCC)*, Kharagpur, India, Feb. 2012, pp. 1–4, doi: 10.1109/NCC.2012.6176765 [Online]. Available: <http://ieeexplore.ieee.org/document/6176765/>. [Accessed: Jan. 18, 2022]
- [56] A. W. Franzke *et al.*, “Users’ and therapists’ perceptions of myoelectric multi-function upper limb prostheses with conventional and pattern recognition control,” *PLoS ONE*, vol. 14, no. 8, p. e0220899, Aug. 2019, doi: 10.1371/journal.pone.0220899.
- [57] R. B. van Dijksseldonk, J. E. Vriezেকolk, N. L. W. Keijsers, A. C. H. Geurts, and I. J. W. van Nes, “Needs and wishes for the future lower limb exoskeleton: an interview study

- among people with spinal cord injury with community-based exoskeleton experience,” *Disability and Rehabilitation*, pp. 1–8, Mar. 2022, doi: 10.1080/09638288.2022.2055158.
- [58] P. M. Pilarski *et al.*, “Adaptive artificial limbs: a real-time approach to prediction and anticipation,” *IEEE Robot. Automat. Mag.*, vol. 20, no. 1, pp. 53–64, Mar. 2013, doi: 10.1109/MRA.2012.2229948.
- [59] A. L. Edwards *et al.*, “Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching,” *Prosthetics and Orthotics International*, p. 9.
- [60] 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 *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Rome, Italy, Jun. 2012, pp. 296–302, doi: 10.1109/BioRob.2012.6290309 [Online]. Available: <http://ieeexplore.ieee.org/document/6290309/>. [Accessed: Jun. 29, 2022]
- [61] A. L. Edwards, J. S. Hebert, and P. M. Pilarski, “Machine learning and unlearning to autonomously switch between the functions of a myoelectric arm,” in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Singapore, Singapore, Jun. 2016, pp. 514–521, doi: 10.1109/BIOROB.2016.7523678 [Online]. Available: <http://ieeexplore.ieee.org/document/7523678/>. [Accessed: Jun. 29, 2022]
- [62] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers,” *IBM J. Res. & Dev.*, vol. 3, no. 3, pp. 210–229, Jul. 1959, doi: 10.1147/rd.33.0210.
- [63] D. Zeithamova *et al.*, “Brain Mechanisms of Concept Learning,” *J. Neurosci.*, vol. 39, no. 42, pp. 8259–8266, Oct. 2019, doi: 10.1523/JNEUROSCI.1166-19.2019.
- [64] A. Dalrymple, “Machine Learning to Characterize Motor Patterns and Restore Walking after Neural Injury,” 2019, doi: 10.7939/R3-6Q2S-S362. [Online]. Available: <https://era.library.ualberta.ca/items/983976d2-e315-4bdc-8cf2-ae29e847f841>. [Accessed: Jul. 15, 2022]
- [65] A. A. Fenton *et al.*, “Attention-Like Modulation of Hippocampus Place Cell Discharge,” *Journal of Neuroscience*, vol. 30, no. 13, pp. 4613–4625, Mar. 2010, doi: 10.1523/JNEUROSCI.5576-09.2010.
- [66] D. Badre and M. D’Esposito, “Is the rostro-caudal axis of the frontal lobe hierarchical?,” *Nat Rev Neurosci*, vol. 10, no. 9, pp. 659–669, Sep. 2009, doi: 10.1038/nrn2667.
- [67] T. J. Buschman and E. K. Miller, “Top-Down Versus Bottom-Up Control of Attention in the Prefrontal and Posterior Parietal Cortices,” *Science*, vol. 315, no. 5820, pp. 1860–1862, Mar. 2007, doi: 10.1126/science.1138071.
- [68] R. D. Cook and S. Weisberg, “Criticism and Influence Analysis in Regression,” *Sociological Methodology*, vol. 13, p. 313, 1982, doi: 10.2307/270724.
- [69] S. Ruder, “An overview of gradient descent optimization algorithms,” 2016, doi: 10.48550/ARXIV.1609.04747. [Online]. Available: <https://arxiv.org/abs/1609.04747>. [Accessed: Jul. 19, 2022]
- [70] D. Y. Li, A. Becker, K. A. Shorter, T. Bretl, and E. T. Hsiao-Wecksler, “Estimating System State During Human Walking With a Powered Ankle-Foot Orthosis,” *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 5, pp. 835–844, Oct. 2011, doi: 10.1109/TMECH.2011.2161769.

- [71] J. M. Hahne *et al.*, “Linear and Nonlinear Regression Techniques for Simultaneous and Proportional Myoelectric Control,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 2, pp. 269–279, Mar. 2014, doi: 10.1109/TNSRE.2014.2305520.
- [72] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.
- [73] D. Xu, X. Liu, and Q. Wang, “Knee Exoskeleton Assistive Torque Control Based on Real-Time Gait Event Detection,” *IEEE Trans. Med. Robot. Bionics*, vol. 1, no. 3, pp. 158–168, Aug. 2019, doi: 10.1109/TMRB.2019.2930352.
- [74] He Huang, Fan Zhang, L. J. Hargrove, Zhi Dou, D. R. Rogers, and K. B. Englehart, “Continuous Locomotion-Mode Identification for Prosthetic Legs Based on Neuromuscular–Mechanical Fusion,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 10, pp. 2867–2875, Oct. 2011, doi: 10.1109/TBME.2011.2161671.
- [75] E. Mizutani, S. E. Dreyfus, and K. Nishio, “On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, 2000, pp. 167–172 vol.2, doi: 10.1109/IJCNN.2000.857892 [Online]. Available: <http://ieeexplore.ieee.org/document/857892/>. [Accessed: Jul. 17, 2022]
- [76] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [77] J.-L. Ren, Y.-H. Chien, E.-Y. Chia, L.-C. Fu, and J.-S. Lai, “Deep Learning based Motion Prediction for Exoskeleton Robot Control in Upper Limb Rehabilitation,” in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, May 2019, pp. 5076–5082, doi: 10.1109/ICRA.2019.8794187 [Online]. Available: <https://ieeexplore.ieee.org/document/8794187/>. [Accessed: Jul. 20, 2022]
- [78] B. Laschowski, W. McNally, A. Wong, and J. McPhee, “Environment Classification for Robotic Leg Prostheses and Exoskeletons Using Deep Convolutional Neural Networks,” *Front. Neurobot.*, vol. 15, p. 730965, Feb. 2022, doi: 10.3389/fnbot.2021.730965.
- [79] B. Fang *et al.*, “Gait Neural Network for Human-Exoskeleton Interaction,” *Front. Neurobot.*, vol. 14, p. 58, Oct. 2020, doi: 10.3389/fnbot.2020.00058.
- [80] S. Bai, J. Z. Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.” arXiv, Apr. 19, 2018 [Online]. Available: <http://arxiv.org/abs/1803.01271>. [Accessed: Jul. 21, 2022]
- [81] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.
- [82] J. B. Travnik and P. M. Pilarski, “Representing high-dimensional data to intelligent prostheses and other wearable assistive robots: A first comparison of tile coding and selective Kanerva coding,” in *2017 International Conference on Rehabilitation Robotics (ICORR)*, London, Jul. 2017, pp. 1443–1450, doi: 10.1109/ICORR.2017.8009451 [Online]. Available: <https://ieeexplore.ieee.org/document/8009451/>. [Accessed: Jan. 21, 2022]
- [83] P. Kanerva, *Sparse distributed memory*. Cambridge, Mass: MIT Press, 1988.
- [84] C. A. R. Hoare, “Algorithm 65: find,” *Commun. ACM*, vol. 4, no. 7, pp. 321–322, Jul. 1961, doi: 10.1145/366622.366647.

- [85] B. Millidge, M. Walton, and R. Bogacz, “Reward Bases: Instantaneous reward revaluation with temporal difference learning,” *Neuroscience*, preprint, Apr. 2022 [Online]. Available: <http://biorxiv.org/lookup/doi/10.1101/2022.04.14.488361>. [Accessed: Aug. 04, 2022]
- [86] K. J. Friston, G. Tononi, G. N. Reeke, O. Sporns, and G. M. Edelman, “Value-dependent selection in the brain: Simulation in a synthetic neural model,” *Neuroscience*, vol. 59, no. 2, pp. 229–243, Mar. 1994, doi: 10.1016/0306-4522(94)90592-4.
- [87] W. Schultz, P. Dayan, and P. R. Montague, “A Neural Substrate of Prediction and Reward,” *Science*, vol. 275, no. 5306, pp. 1593–1599, Mar. 1997, doi: 10.1126/science.275.5306.1593.
- [88] E. A. Ludvig, R. S. Sutton, and E. J. Kehoe, “Evaluating the TD model of classical conditioning,” *Learn Behav*, vol. 40, no. 3, pp. 305–319, Sep. 2012, doi: 10.3758/s13420-012-0082-6.
- [89] E. A. Ludvig, R. S. Sutton, and E. J. Kehoe, “Stimulus Representation and the Timing of Reward-Prediction Errors in Models of the Dopamine System,” *Neural Computation*, vol. 20, no. 12, pp. 3034–3054, Dec. 2008, doi: 10.1162/neco.2008.11-07-654.
- [90] S. Richard *et al.*, “Horde : A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction,” *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pp. 761–768, 2011.
- [91] A. White, “Developing a Predictive Approach to Knowledge,” 2015, doi: 10.7939/R3FF3M75H. [Online]. Available: <https://era.library.ualberta.ca/items/7f973a64-35c9-4109-9a79-d87edb44ae52>. [Accessed: Feb. 15, 2022]
- [92] J. Modayil, A. White, and R. S. Sutton, “Multi-timescale Nexting in a Reinforcement Learning Robot,” *arXiv:1112.1133 [cs]*, Jun. 2012 [Online]. Available: <http://arxiv.org/abs/1112.1133>. [Accessed: Feb. 15, 2022]
- [93] M. U. Manto and A. G. Shaikh, “Editorial: Predictive Mechanisms of the Cerebello-Cerebral Networks,” *Front. Cell. Neurosci.*, vol. 13, p. 549, Dec. 2019, doi: 10.3389/fncel.2019.00549.
- [94] A. A. Sokolov, R. C. Miall, and R. B. Ivry, “The Cerebellum: Adaptive Prediction for Movement and Cognition,” *Trends in Cognitive Sciences*, vol. 21, no. 5, pp. 313–332, May 2017, doi: 10.1016/j.tics.2017.02.005.
- [95] H. Tanaka, T. Ishikawa, J. Lee, and S. Kakei, “The Cerebro-Cerebellum as a Locus of Forward Model: A Review,” *Front. Syst. Neurosci.*, vol. 14, p. 19, Apr. 2020, doi: 10.3389/fnsys.2020.00019.
- [96] D. M. Wolpert, Z. Ghahramani, and J. R. Flanagan, “Perspectives and problems in motor learning,” *Trends in Cognitive Sciences*, vol. 5, no. 11, pp. 487–494, Nov. 2001, doi: 10.1016/S1364-6613(00)01773-3.
- [97] M. L. Streng, L. S. Popa, and T. J. Ebner, “Climbing fibers predict movement kinematics and performance errors,” *Journal of Neurophysiology*, vol. 118, no. 3, pp. 1888–1902, Sep. 2017, doi: 10.1152/jn.00266.2017.
- [98] D. Purves and S. M. Williams, Eds., *Neuroscience*, 2nd ed. Sunderland, Mass: Sinauer Associates, 2001.
- [99] R. Shadmehr, M. A. Smith, and J. W. Krakauer, “Error Correction, Sensory Prediction, and Adaptation in Motor Control,” *Annu. Rev. Neurosci.*, vol. 33, no. 1, pp. 89–108, Jun. 2010, doi: 10.1146/annurev-neuro-060909-153135.
- [100] I. Pisotta and M. Molinari, “Cerebellar contribution to feedforward control of locomotion,” *Front. Hum. Neurosci.*, vol. 8, Jun. 2014, doi: 10.3389/fnhum.2014.00475.

- [Online]. Available:
<http://journal.frontiersin.org/article/10.3389/fnhum.2014.00475/abstract>. [Accessed: Dec. 27, 2022]
- [101] C. Sherstan, “Representation and General Value Functions,” 2020, doi: 10.7939/R3-8BEV-AP57. [Online]. Available: <https://era.library.ualberta.ca/items/241a73f6-1f2c-4ba2-9fa8-da1d6fca7713>. [Accessed: Dec. 27, 2022]
- [102] J. Modayil and R. S. Sutton, “Prediction Driven Behavior: Learning Predictions that Drive Fixed Responses,” p. 7.
- [103] S. Revusky, “Animal Learning: *The Psychology of Animal Learning*. N. J. Mackintosh. Academic Press, New York, 1974. xiv, 730 pp., illus. \$18.50.,” *Science*, vol. 189, no. 4197, pp. 131–131, Jul. 1975, doi: 10.1126/science.189.4197.131.a.
- [104] A. L. Edwards, “Adaptive and Autonomous Switching: Shared Control of Powered Prosthetic Arms Using Reinforcement Learning,” 2016, doi: 10.7939/R35Q4RX49. [Online]. Available: <https://era.library.ualberta.ca/items/bab4a49e-d4b7-4def-9399-d8fb176e2c18>. [Accessed: Dec. 22, 2022]
- [105] P. Faridi *et al.*, “Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control: Preliminary Results,” in *2022 International Conference on Rehabilitation Robotics (ICORR)*, Rotterdam, Netherlands, Jul. 2022, pp. 1–6, doi: 10.1109/ICORR55369.2022.9896611 [Online]. Available: <https://ieeexplore.ieee.org/document/9896611/>. [Accessed: Nov. 09, 2022]
- [106] W.-Z. Li, G.-Z. Cao, and A.-B. Zhu, “Review on Control Strategies for Lower Limb Rehabilitation Exoskeletons,” *IEEE Access*, vol. 9, pp. 123040–123060, 2021, doi: 10.1109/ACCESS.2021.3110595.
- [107] J. de la Tejera, R. Bustamante-Bello, R. A. Ramirez-Mendoza, and J. Izquierdo-Reyes, “Systematic Review of Exoskeletons towards a General Categorization Model Proposal,” *Applied Sciences*, vol. 11, pp. 1–25, Dec. 2020, doi: 10.3390/app11010076.
- [108] Mojtaba Sharifi, Javad K mehr, Vivian K Mushahwar, and Mahdi Tavakoli, “Autonomous locomotion trajectory shaping and nonlinear control for lower-limb exoskeletons,” *IEEE/ASME Transactions on Mechatronics*, Published Online 2022.
- [109] J. K. Mehr, M. Sharifi, V. K. Mushahwar, and M. Tavakoli, “Intelligent Locomotion Planning With Enhanced Postural Stability for Lower-Limb Exoskeletons,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7588–7595, Oct. 2021, doi: 10.1109/LRA.2021.3098915.
- [110] Spinal Cord Injury Research Evidence (SCIRE) Research Team, D. R. Louie, J. J. Eng, and T. Lam, “Gait speed using powered robotic exoskeletons after spinal cord injury: a systematic review and correlational study,” *J NeuroEngineering Rehabil*, vol. 12, no. 1, p. 82, Dec. 2015, doi: 10.1186/s12984-015-0074-9.
- [111] A. N. Dalrymple, D. A. Roszko, R. S. Sutton, and V. K. Mushahwar, “Pavlovian control of intraspinal microstimulation to produce over-ground walking,” *J. Neural Eng.*, vol. 17, no. 3, p. 036002, Jun. 2020, doi: 10.1088/1741-2552/ab8e8e.
- [112] S. P. Singh and R. S. Sutton, “Reinforcement Learning with Replacing Eligibility Traces,” *Machine Learning*, vol. 22, no. 1/2/3, pp. 123–158, 1996, doi: 10.1023/A:1018012322525.
- [113] C. Baunsgaard *et al.*, “Exoskeleton gait training after spinal cord injury: An exploratory study on secondary health conditions,” *J Rehabil Med*, vol. 50, no. 9, pp. 806–813, 2018, doi: 10.2340/16501977-2372.

- [114] L. Miller, A. Zimmermann, and W. Herbert, “Clinical effectiveness and safety of powered exoskeleton-assisted walking in patients with spinal cord injury: systematic review with meta-analysis,” *MDER*, p. 455, Mar. 2016, doi: 10.2147/MDER.S103102.
- [115] A. Chun *et al.*, “Changes in bowel function following exoskeletal-assisted walking in persons with spinal cord injury: an observational pilot study,” *Spinal Cord*, vol. 58, no. 4, pp. 459–466, Apr. 2020, doi: 10.1038/s41393-019-0392-z.
- [116] B. S. Rupal, S. Rafique, A. Singla, E. Singla, M. Isaksson, and G. S. Virk, “Lower-limb exoskeletons: Research trends and regulatory guidelines in medical and non-medical applications,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 172988141774355, Nov. 2017, doi: 10.1177/1729881417743554.
- [117] C. Hartigan *et al.*, “Mobility Outcomes Following Five Training Sessions with a Powered Exoskeleton,” *Top Spinal Cord Inj Rehabil*, vol. 21, no. 2, pp. 93–99, 2015, doi: 10.1310/sci2102-93.
- [118] G. Zeilig, H. Weingarden, M. Zwecker, I. Dudkiewicz, A. Bloch, and A. Esquenazi, “Safety and tolerance of the ReWalk™ exoskeleton suit for ambulation by people with complete spinal cord injury: a pilot study,” *J Spinal Cord Med*, vol. 35, no. 2, pp. 96–101, Mar. 2012, doi: 10.1179/2045772312Y.0000000003.
- [119] K. A. Inkol and J. McPhee, “Assessing Control of Fixed-Support Balance Recovery in Wearable Lower-Limb Exoskeletons Using Multibody Dynamic Modelling,” in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechanics (BioRob)*, Nov. 2020, pp. 54–60, doi: 10.1109/BioRob49111.2020.9224430.
- [120] G. Barbareschi, R. Richards, M. Thornton, T. Carlson, and C. Holloway, “Statically vs dynamically balanced gait: Analysis of a robotic exoskeleton compared with a human,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Aug. 2015, pp. 6728–6731, doi: 10.1109/EMBC.2015.7319937 [Online]. Available: <http://ieeexplore.ieee.org/document/7319937/>. [Accessed: Nov. 07, 2022]
- [121] K. Kreamer-Tonin, “Analysis of the User Requirements and Product Specifications for Home-Use of the ABLE Exoskeleton,” (*TRITA-CBH-GRU*). Available from: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-303993>, 2021.
- [122] S. K. Banala, S. H. Kim, S. K. Agrawal, and J. P. Scholz, “Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX),” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 1, pp. 2–8, Feb. 2009, doi: 10.1109/TNSRE.2008.2008280.
- [123] C. A. McGibbon *et al.*, “Evaluation of the Keeogo exoskeleton for assisting ambulatory activities in people with multiple sclerosis: an open-label, randomized, cross-over trial,” *J NeuroEngineering Rehabil*, vol. 15, no. 1, p. 117, Dec. 2018, doi: 10.1186/s12984-018-0468-6.
- [124] J. M. P. Poritz, H. B. Taylor, G. Francisco, and S.-H. Chang, “User satisfaction with lower limb wearable robotic exoskeletons,” *Disability and Rehabilitation: Assistive Technology*, vol. 15, no. 3, pp. 322–327, Apr. 2020, doi: 10.1080/17483107.2019.1574917.
- [125] M. R. Tucker *et al.*, “Control strategies for active lower extremity prosthetics and orthotics: a review,” *J NeuroEngineering Rehabil*, vol. 12, no. 1, p. 1, 2015, doi: 10.1186/1743-0003-12-1.
- [126] T. Yan *et al.*, “A novel adaptive oscillators-based control for a powered multi-joint lower-limb orthosis,” in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Singapore, Singapore, Aug. 2015, pp. 386–391, doi:

- 10.1109/ICORR.2015.7281230 [Online]. Available: <http://ieeexplore.ieee.org/document/7281230/>. [Accessed: Jan. 09, 2022]
- [127] Keehong Seo, SeungYong Hyung, Byung Kwon Choi, Younbaek Lee, and Youngbo Shim, “A new adaptive frequency oscillator for gait assistance,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015, pp. 5565–5571, doi: 10.1109/ICRA.2015.7139977 [Online]. Available: <http://ieeexplore.ieee.org/document/7139977/>. [Accessed: Jan. 09, 2022]
- [128] I. Kang, P. Kunapuli, and A. J. Young, “Real-Time Neural Network-Based Gait Phase Estimation Using a Robotic Hip Exoskeleton,” *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 28–37, Feb. 2020, doi: 10.1109/TMRB.2019.2961749.
- [129] P. Beyl *et al.*, “Safe and Compliant Guidance by a Powered Knee Exoskeleton for Robot-Assisted Rehabilitation of Gait,” *Advanced Robotics*, vol. 25, no. 5, pp. 513–535, Jan. 2011, doi: 10.1163/016918611X558225.
- [130] J. Bae and M. Tomizuka, “A gait rehabilitation strategy inspired by an iterative learning algorithm,” *Mechatronics*, vol. 22, no. 2, pp. 213–221, Mar. 2012, doi: 10.1016/j.mechatronics.2012.01.009.
- [131] J. Zhang, C. C. Cheah, and S. H. Collins, “Torque Control in Legged Locomotion *
*Supplementary document of this chapter is located at https://www.andrew.cmu.edu/user/shc17/Zhang_2016_BLL—SuppMat.pdf,” in *Bioinspired Legged Locomotion*, Elsevier, 2017, pp. 347–400 [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/B9780128037669000075>. [Accessed: Jan. 09, 2022]
- [132] S. Oh and K. Kong, “High-Precision Robust Force Control of a Series Elastic Actuator,” *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 1, pp. 71–80, Feb. 2017, doi: 10.1109/TMECH.2016.2614503.
- [133] D.-X. Liu, J. Xu, C. Chen, X. Long, D. Tao, and X. Wu, “Vision-Assisted Autonomous Lower-Limb Exoskeleton Robot,” *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 51, no. 6, pp. 3759–3770, Jun. 2021, doi: 10.1109/TSMC.2019.2932892.
- [134] S. Tanabe, S. Hirano, and E. Saitoh, “Wearable Power-Assist Locomotor (WPAL) for supporting upright walking in persons with paraplegia,” *NRE*, vol. 33, no. 1, pp. 99–106, Oct. 2013, doi: 10.3233/NRE-130932.
- [135] M. Bortole *et al.*, “The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study,” *J NeuroEngineering Rehabil*, vol. 12, no. 1, p. 54, Dec. 2015, doi: 10.1186/s12984-015-0048-y.
- [136] C.-H. Wu, H.-F. Mao, J.-S. Hu, T.-Y. Wang, Y.-J. Tsai, and W.-L. Hsu, “The effects of gait training using powered lower limb exoskeleton robot on individuals with complete spinal cord injury,” *J NeuroEngineering Rehabil*, vol. 15, no. 1, p. 14, Dec. 2018, doi: 10.1186/s12984-018-0355-1.
- [137] S. R. Chang *et al.*, “A muscle-driven approach to restore stepping with an exoskeleton for individuals with paraplegia,” *J NeuroEngineering Rehabil*, vol. 14, no. 1, p. 48, Dec. 2017, doi: 10.1186/s12984-017-0258-6.
- [138] A. W. Franzke *et al.*, “Users’ and therapists’ perceptions of myoelectric multi-function upper limb prostheses with conventional and pattern recognition control,” *PLoS ONE*, vol. 14, no. 8, p. e0220899, Aug. 2019, doi: 10.1371/journal.pone.0220899.

- [139] H. A. Quintero, R. J. Farris, C. Hartigan, I. Clesson, and M. Goldfarb, “A Powered Lower Limb Orthosis for Providing Legged Mobility in Paraplegic Individuals,” *Top Spinal Cord Inj Rehabil*, vol. 17, no. 1, pp. 25–33, 2011, doi: 10.1310/sci1701-25.
- [140] R. Sutton *et al.*, “Horde : A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction Categories and Subject Descriptors,” presented at the Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, Jan. 2011, vol. 2.
- [141] M. White, “Unifying task specification in reinforcement learning,” *ICML ’17: Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3742–3750.
- [142] 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 *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Rome, Italy, Jun. 2012, pp. 296–302, doi: 10.1109/BioRob.2012.6290309 [Online]. Available: <http://ieeexplore.ieee.org/document/6290309/>. [Accessed: Jan. 18, 2022]
- [143] P. M. Pilarski *et al.*, “Adaptive artificial limbs: a real-time approach to prediction and anticipation,” *IEEE Robot. Automat. Mag.*, vol. 20, no. 1, pp. 53–64, Mar. 2013, doi: 10.1109/MRA.2012.2229948.
- [144] A. L. Edwards, J. S. Hebert, and P. M. Pilarski, “Machine learning and unlearning to autonomously switch between the functions of a myoelectric arm,” in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Singapore, Singapore, Jun. 2016, pp. 514–521, doi: 10.1109/BIOROB.2016.7523678 [Online]. Available: <http://ieeexplore.ieee.org/document/7523678/>. [Accessed: Jan. 18, 2022]
- [145] A. L. Edwards *et al.*, “Application of real-time machine learning to myoelectric prosthesis control: A case series in adaptive switching,” *Prosthetics & Orthotics International*, vol. 40, no. 5, pp. 573–581, Oct. 2016, doi: 10.1177/0309364615605373.
- [146] J. Travník, “Reinforcement Learning on Resource Bounded Systems,” *MSc Thesis, University of Alberta*, 2018, doi: 10.7939/R39G5GV5S. [Online]. Available: <https://era.library.ualberta.ca/items/cbbd0fa7-48f8-4118-babd-d5833c7d9ee4>. [Accessed: Nov. 13, 2022]
- [147] P. Faridi *et al.*, “Machine-learned Adaptive Switching in Voluntary Lower-limb Exoskeleton Control: Preliminary Results,” *Accepted for publication in the Proceedings of 2022 IEEE 17th International Conference on Rehabilitation Robotics (ICORR)*, 2022.
- [148] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Mach Learn*, vol. 3, no. 1, pp. 9–44, Aug. 1988, doi: 10.1007/BF00115009.
- [149] H. R. Maei, “Gradient Temporal-Difference Learning Algorithms,” 2011, doi: 10.7939/R30Q50. [Online]. Available: <https://era.library.ualberta.ca/items/fd55edcb-ce47-4f84-84e2-be281d27b16a>. [Accessed: Dec. 06, 2022]
- [150] P. Montague, P. Dayan, and T. Sejnowski, “A framework for mesencephalic dopamine systems based on predictive Hebbian learning,” *J. Neurosci.*, vol. 16, no. 5, pp. 1936–1947, Mar. 1996, doi: 10.1523/JNEUROSCI.16-05-01936.1996.
- [151] T. Ljungberg, P. Apicella, and W. Schultz, “Responses of monkey dopamine neurons during learning of behavioral reactions,” *Journal of Neurophysiology*, vol. 67, no. 1, pp. 145–163, Jan. 1992, doi: 10.1152/jn.1992.67.1.145.

- [152] W. Schultz, P. Apicella, and T. Ljungberg, “Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task,” *J. Neurosci.*, vol. 13, no. 3, pp. 900–913, Mar. 1993, doi: 10.1523/JNEUROSCI.13-03-00900.1993.
- [153] S. M. Morton and A. J. Bastian, “Cerebellar Contributions to Locomotor Adaptations during Splitbelt Treadmill Walking,” *Journal of Neuroscience*, vol. 26, no. 36, pp. 9107–9116, Sep. 2006, doi: 10.1523/JNEUROSCI.2622-06.2006.