Reverse engineering the motor control system

Thèse N° 9599

Présentée le 28 août 2019

à la Centres pour la recherche
Programme doctoral en neurosciences

pour l'obtention du grade de Docteur ès Sciences

par

Berat DENIZDURDURAN

Acceptée sur proposition du jury

Prof. C. Petersen, président du jury
Prof. H. Markram, Dr M.-O. Gewaltig, directeurs de thèse
Prof. E. Falotico, rapporteur
Prof. Y. Sandamirskaya, rapporteuse
Prof. A. Ijspeert, rapporteur

2019

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

To my family…

Acknowledgements

First of all, I would like to thank Henry Markram, my supervisor, for giving me the oppprtunity to pursue my Ph.D. studies at the Blue Brain Project, for his encouragement, support and guidance. I cannot thank enough for my co-supervisor, Marc-Oliver Gewaltig, for his endless efforts to help me go through at every single step of my studies, for allowing me to study all my ideas and hypotheses. I also would like thank my lifelong mentor, Neslihan Serap Sengor, who help me to shape my mind whenever I need it since the day we met.

I also would like to thank all the current and past members of the Neurorobotics group at Blue Brain Project, Athanassia Chalimourda (Nancy), Willem Wybo, Daniel Peppicelli, Yves Dornbierer, Csaba Ero, Ihor Kuras, Dimitri Rodarie, Luc Guyot, Cladio Sousa, Susie Murphy, Bianca Schmiedle. I also would like to thank my friends at the Blue Brain Project for their wonderful company during my journey, Ahmet Bilgili, Danny Dyer, Jay Coggan, Alina Busuioc Jimenez, Bruno Magalhaes, Tristan Maquart-Bothorel, Xavier Piffard, Cyrille Favreau, James Gonzalo King, Nicolas Antille, Genrich Ivaska, Juan Hernando and Mike. Special thanks to Lida Kanari and Elefterios Zisis for always interesting discussions.

I'd like to thank my close friends just because of their existence in my life, Tuna Yekeler, Ozgun Cirakman, Omer Deniz Akyildiz, Busra Topal, Zeynep Derelioglu, Ramiz Erdem Aykac, Dilara Yetik Aykac, Ozgur Yonkuc, Ahmet Altuntas, Bilge Karamehmet, Manos Agelidis, Burcu Tepekule, Emec Ercelik, Florin Dzeladini.

And Elzbieta Szulc for becoming the color of my life.

Finally, I would like to thank my beautiful mother, Gunay, for every single moment that we have been sharing together, my beloved father, Ramazan, for his biggest heart for all of us, and my sister Gamze, my brother-in-law Okkes and our new, tiny little member of the family, Ege, who gave all of us another reason to live.

Geneva, 30 July 2019 B. D.

Abstract

Mammalian motor control is implemented by a combination of different networks and systems, working coherently to plan the movement of the body or a limb and to execute this movement in a dynamic environment. While it is believed that complex, voluntary movements are planned in the motor areas of the cortex, the execution of the movement is controlled by a combination of cortical and cerebellar networks together with central pattern generators and reflex circuits in the spinal cord. In this thesis, we propose an abstract model that captures the basic properties of mammalian motor control, using the example of two different movements: arm movement as well as locomotion. Our model consists of three parts: first a high-level network that learns movements using a combination of actor-critic based reinforcement learning and Optimal Control theory. The second part corresponds to spinal reflex circuits that execute basic movements of the musculo-skeletal system. They are modeled based on a simple neural network, that learns the dynamic properties of the musculoskeletal system by the mechanism of spontaneous motor activity (muscle twitching) combined with a Hebbian learning rule. We demonstrate that this network can learn the antagonistic control of joint movements. The third and final part of the model is the cerebellar network, that translates a complex movement trajectory, such as reaching, and activates the spinal reflex circuits to execute the movement. The mapping between the cerebellar neurons and the spinal reflex circuits are trained with artificial neural networks. Using a musculoskeletal arm model, we will demonstrate that the proposed neural motor control model can generate the movement to the arbitrary goal position.

Key words: Closed-loop Motor Control, Optimal Control, Nonlinear Programming, Reinforcement Learning, Hebbian Learning, Neural Networks, Arm Control, Biped Locomotion

Résumé

Le contrôle moteur des mammifères est mis en œuvre par une combinaison de différents réseaux et systèmes, travaillant de manière cohérente pour planifier le mouvement du corps ou d'un de ses membres afin d'exécuter ce mouvement dans un environnement dynamique. Bien que l'on pense que des mouvements volontaires complexes sont prévus dans les zones motrices du cortex, l'exécution du mouvement est contrôlée par une combinaison de réseaux corticaux et cérébelleux d'un part et de circuits réflexes et générateurs de motif centraux dans la moelle épinière. Dans cette thèse, nous proposons un modèle abstrait qui capture les propriétés de base du contrôle moteur des mammifères, en utilisant l'exemple de deux mouvements différents : les mouvements des bras et la locomotion. Notre modèle se divise en trois parties : tout d'abord un réseau de haut niveau, qui apprend les mouvements, en utilisant une combinaison d'apprentissage par renforcement basé sur la méthode d'acteur-critique et de contrôle optimal. La deuxième partie correspond aux circuits réflexes spinaux qui exécutent les mouvements de base du système musculo-squelettique. Les circuits spinaux sont modélisés par un réseau neuronal simple, qui apprend les propriétés dynamiques du système musculo-squelettique par un mécanisme combinant des contractions musculaires spontanées et une règle d'apprentissage hébbien. Nous démontrons que ce réseau peut apprendre le contrôle antagoniste des mouvements articulaires. La dernière partie du modèle est un réseau cérébelleux qui traduit une trajectoire de movement complexe, comme atteindre une position donnée, et active les circuits réflexes spinaux pour exécuter le mouvement. La cartographie entre les neurones cérébelleux et les circuits réflexes spinaux est également entraînée avec des réseaux neuronaux artificiels. À l'aide d'un modèle de bras musculo-squelettique, nous démontrons que le modèle de contrôle neuronal proposé peut générer un mouvement vers une position arbitraire.

Mots clefs : Contrôle Moteur en Boucle Fermé, Contrôle Optimal, Apprentissage par Renforcement, Apprentissage Hébbien, Réseaux De Neurones, Mouvement des membres, Locomotion bipède

Contents

List of Figures

Acronyms

CNS Central nervous system.

CPG Central Pattern Generator.

EMG Electromyography.

EPH Equilibrium-point hypothesis.

HJB Hamilton-Jacobi-Bellman Equation.

KKT Karush-Kuhn-Tucker (KKT) conditions.

MTU muscle-tendon units.

NLP Nonlinear Programming.

OCT Optimal control theory.

PMC Primary Motor Cortex.

PMP Pontryagin's Maximum Principle.

RL Reinforcement Learning.

UMT Uncontrolled manifold theorem.

1 Introduction

The ability of perceiving changes in the environment and acting accordingly with motor skills underlie the fundamental role of Central nervous system (CNS). Driven by the desire of amusement, survival or curiosity, motor skills comprise the range of all possible actions that can be executed to achieve any of the goals set by itself. A significant feature of CNS is that a new motor skill can be added on top of the existing ones, by which it re-organizes the sequence of learned actions to enrich the capabilities of motor control (Hikosaka et al., 2002; Luft and Buitrago, 2005; Shmuelof et al., 2012; Willingham, 1998). These learned motor skills can be possessed and last throughout the entire lifetime (Park et al., 2013; Romano et al., 2010) which indicates that CNS has the ability to store long-term motor skills based on a long-term retained plasticity mechanism (Dayan and Cohen, 2011; Ungerleider et al., 2002). Yet, the division of labor in acquiring new motor skills or execution of already integrated ones and the details of the role of perception/sensory integration among all distributed motor areas are still the interest of active research (Kawai et al., 2015).

Either coordinated or involuntary, all motor behavior requires active control of tens to hundreds of muscles. There are several connected loops in motor control, including hierarchical networks from muscles and sensory organs, over the spinal cord to high-level motor circuits in the cortex and finally back to the muscles. Each loop consists of several complex sub-units (Rosenbaum, 2009). Therefore revealing the function of the motor control circuit requires us to understand each of these subunits and their respective role within the hierarchy.

1.1 Neural Motor Control

Any motor behavior - grasping an object, lifting weights, walking or jogging is a result of multiple muscle contractions, acting on different skeletal bones. This interaction between muscles and skeletal bones generated-movements is controlled by the motor control circuits. To resolve this control problem, the motor control circuits recruit only those muscles that take part in the corresponding movement that is planned while controlling them in a timely fashion. These muscle contractions are triggered when an α-motoneuron innervates an extrafusal muscle fiber. Therefore, it can be stated that a motor behavior in fact starts when an α-motoneuron is recruited by motor control circuits to innervate the related muscle fiber, which generates force to act on joints, hence initializing the movement (Kernell, 2006). To enable movement, the motor control circuits are implemented by a combination of different networks, working coherently. What is accepted today is that the sensory integration, planning and execution of movement are organized as a hierarchy of distributed motor control networks with three main levels. The highest level of the hierarchy is regarded as the areas of neocortex and forebrain, for instance the parietal cortex, where coordinated motor programs are planned and associated with sensory information. Then based on the desired movement, the recruitment of the low level motor areas is generated mostly in premotor and motor areas as well as in the cerebellum, which is known as middle level of neural motor control circuit (Bear et al., 2007). The timing and accuracy of redundant muscle control is achieved by descending commands of these circuits to the low-level circuits in order to obtain smooth and complex desired movements. Lastly, the movement itself is executed at the low level of the hierarchy by the peripheral spinal and neuromuscular system, composed of the spinal cord and muscles (Donoghue and Sanes, 1994) while recruiting $alpha$-motoneurons and interneuron pools coherently (Monti et al., 2001). Therefore, the motor control circuit can be divided into three different parts based on their individual roles in movement, see also Figure 1.1;

- High-level cortical areas - Neocortex and forebrain: Movement strategy and goal setup
- Middle-level cortical areas - Premotor, motor areas and cerebellum: Recruitment of low level motor circuits related to desired goal
- The spinal cord - muscle closed-loop circuit: Execution of the movement with redundant muscle contractions

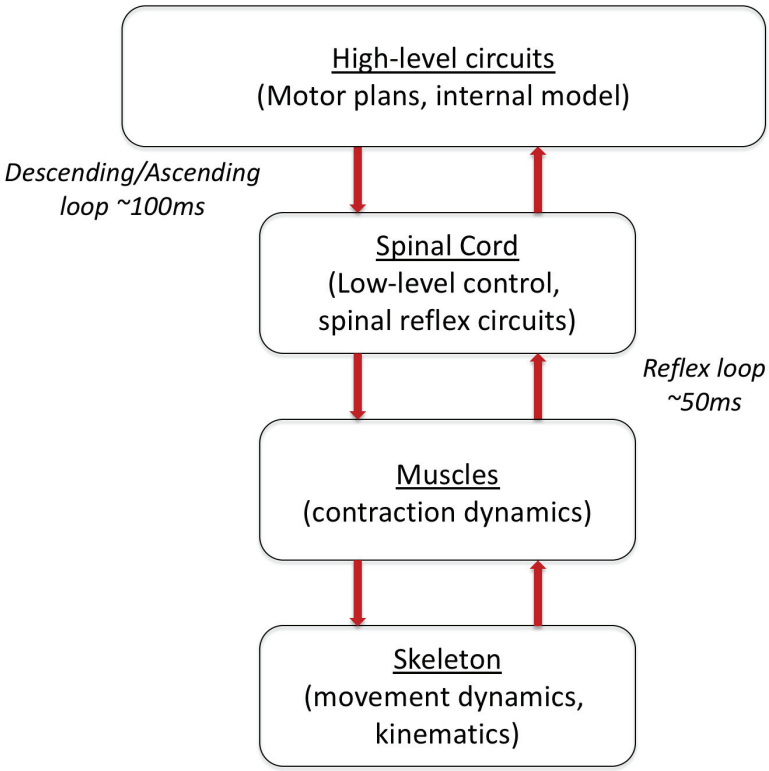


Figure 1.1: The closed-loop motor control circuit. Three main sub-components of the motor control circuit is considered, musculoskeletal system, low-level circuit (muscle-spinal cord), high-level circuits (spinal cord-high-level motor circuits).

This broad approximation indicates that an important part of the function of motor control circuits depends on sensory integration among several different modalities, hence the name of sensorimotor system (Riemann and Lephart, 2002a,b). At the high and middle level systems, sensory information about the environment and body positions are mainly dealt. Such as, integrating the mental image of the body with respect to surrounding environment, linking prevailing decisions to memory of past movement and maintaining the body posture are part of the high and middle level systems' function. Nevertheless, inclusion of sensory information about skeletal joints positions and muscle activations takes part in the low level of hierarchy such as translation of kinematic and dynamic properties of body parts based on sensory feedback from muscles and joints.

As a typical example of a motor behavior, like grasping an object, requires to first, move the

joints of the arm in such a way that a particular position in 3D space is reached and second, to move the joints of the hand in such a way that the object can be grasped. First, the position of the object in space needs to be determined from the sensory information (e.g. tactile, visual, auditory) which is processed in the high-level neocortical areas (Buneo and Andersen, 2006). The precise position of the body according to environment, global strategy to enable the desired action among several possibilities while considering the past experience are all processed in the high-level circuits. These global decisions and strategies are then used to plan a movement trajectory of the hand from its current position to the location of the object by the middle level cortical areas such as premotor and motor cortex (Nakayama et al., 2008). The main objective of these circuits is to proceed the movement by instructing the brainstem, cerebellum and spinal cord circuits. Besides the execution of the movement, the motor control circuits needs to monitor the changes in the kinematic and dynamic *properties* of the moving limbs, e.g. the joint positions, velocities and accelerations or the relationship between the force created by the muscles and the corresponding joint positions due to these forces (Adrian, 1943; Bastian et al., 2000). Nevertheless, for coordinated voluntary behaviors, high-level cortical areas are involved in the movement generation through multiple sensorimotor integration (Gritsenko et al., 2009; Sober and Sabes, 2005; Abbruzzese and Berardelli, 2003). The descending signals coordinate with the primitive reflex circuits and create either repetitive (e.g locomotion) or goal-directed behaviors (e.g. grasping and reaching) (Drew et al., 2008; Drew, 1988; Lajoie et al., 2010; Shik, 1966; Asanuma and Rosen, 1972; Ashe and Georgopoulos, 1994; Jackson et al., 2007).

Since the movement is generated by contracting multiple muscles and each muscle has a different role in the movement, the motor control circuits needs to decide which muscles need to be activated in which order to achieving the grasping movement. To obtain a desired movement out of a sequence of optimal specific joint angles while integrating the sensory modalities, the peripheral motor control is responsible to move these joints to the desired angles by appropriately contracting the respective muscle groups. There is typically more than one way of performing a specific movement. This phenomenon is called *motor equivalence* (Bernstein, 1966; Flash and Hochner, 2005). As of now, there is no agreement about which part of the motor control hierarchy is responsible to select the best out of the infinitely many movement trajectories. Apart from that, the movement, planned in egocentric coordinates, has to be translated into "muscle coordinates". This transformation is complicated by the fact that each joint is controlled antagonistically, that is, at least two muscles move one

joint by pulling in opposite directions. Moreover, a single muscle can be involved in the movement of the adjacent joints (Enoka, 2008; Kernell, 2006; Monti et al., 2001). Thus, to move a single joint against an external force (say gravity) to a desired position, the contraction of at least two antagonistic muscles needs to be coordinated. This low-level closed-loop control is implemented by neural circuits in the spinal cord. These peripheral loops provide the essential building blocks for automatic and stereotyped movements without the intervention of high-level cortical areas. Hence, the grasping movement is created by these low-level circuits while properly timed recruitment of the $alpha$-motoneurons when considering the sensory feedback. These feedback signals are created by the body itself and is called proprioception (Kandel et al., 2000; Bear et al., 2007) and are executed within the low-level circuits. And so, a real-time (closed-loop) control of movements requires both proprioceptive information about the state of the body (intrinsic) as well as sensory information from the environment (extrinsic). In summary, it can be said that the peripheral spinal motor control is continuously interacting with the high-level motor control through descending commands and ascending sensory feedback. This interaction enables two distinct features of the motor circuits (Miall, 2010; Cooke and Diggles, 1984; Sainburg et al., 1999; Lawrence, 2000; Wulf et al., 2010; Luft and Buitrago, 2005; Jenkins et al., 1994);

1. Error correction during movement
2. Motor learning to improve the performance of the movement

1.1.1 High-Level Motor Circuits

An intention to act

Motor control is the physical and material consequence of the desire of voluntary behavior accompanied with physical interaction of the external environment. It often involves processing several sensory modalities to integrate the conditions of the external world into decision making process. The evaluation of these external conditions interacts with individual's prevailing state of mind along with calls of memory in order to obtain the judgements of past experiences. All these distributed cognitive processes then yield the intention of interacting with the surrounding environment as a physical expression of voluntary behavior.

Throughout studies of arm movements that have attempted to study the individual information pathways related to motor planning and execution, it has been identified that none of

these cortical areas show homogenous activities. By instructing a delay between movement and execution cue in the experimental setup, a broad range of functionality has been observed within the same region of cortical areas. While some of the neurons were involved in the preparation phase of movement, others were activated for the execution of the task as well as some neurons observed for both phases of the voluntary movement (Crammond and Kalaska, 2000). It is indicated that there is no clear functional separation between different cortical areas that are involved in motor planning and execution. The intention and action are processed within this distributed network to obtain the voluntary behavior (Crammond and Kalaska, 2000).

A broad range of division of labor within cortical areas of motor circuit is an ongoing research area. It has been shown that motor control system is not only involved in executing voluntary motor behaviors but also in deciding on which goals to set and how to organize the remaining circuitry to obtain the desired movements (Andersen and Buneo, 2002; Cisek and Kalaska, 2010). Instead of being a passive circuit that only takes part in controlling the movement, it actively participates in several cognitive processes such as decision making and planning. Organizing and choosing the plan of a sequence of actions that leads to a desired behavior requires intimately the possession of control over interrelated cognitive processes (Rizzolatti and Luppino, 2001). One of the key properties of the motor system is the transformation of cognitive intention into voluntary movements which requires perceptual and cognitive processing (Colby and Goldberg, 1999; Rizzolatti and Sinigaglia, 2010). Based on the information about the external environment and past experiences, deciding which goal to set and how to organize the sequence of actions establish the perceptual and cognitive task of motor control circuit, mainly in parietal and premotor cortex (Cui and Andersen, 2007; Fogassi et al., 2005; Gallese et al., 1996).

Although there are neuronal activities while the execution of the act within parietal and premotor cortex, the role of particular neurons for the intention of the voluntary movement have been identified by the activity that has been observed well before the execution of the movement (Wang et al., 2019). One of the unresolved functional implications of the spatial closeness of the neurons that are discharged during the planning and execution is how the required time difference between the intention and act is preserved and why the movement immediately triggered after the decision given that there exists close proximity and interrelated activity within these regions (Libet, 2009; Kalaska and Crammond, 1995). Whathas been observed is that lesions within parietal and premotor areas cause deficiencies

that are related to the initiation or the release of the voluntary behaviors such as a loss of arm control that are self-initiated known as akinesia (Northoff et al., 2000). Other diseases that have been observed due to lesions within parietal and premotor cortex are forced grasping, groping movement, and alien hand syndromes (Goldberg, 1985).

Integration of external world and body

Grasping an object ubiquitously requires reaching out that object too. While executing the reaching movement, hand, wrist and fingers are organized continuously to grasp that object. Therefore it can be stated that, reaching and grasping are occurring simultaneously and in parallel based on evidence found by experiments of arm and hand kinematics (Rizzolatti et al., 1990; Cohen and Andersen, 2002). Performing such two tasks in parallel requires the integration of different sensory information within spatially proximal cortical areas. Consequently this interaction between different populations allows to perform these two movements together. To reach, the motor control system needs to translate the local position of the object in space to a relative position of the body and arm, particularly, it needs to translate and compare this localized information into continuously varying limb movements (Graziano et al., 2000). At the same time, it needs to integrate the information about the shape, size and orientation of that object and select the corresponding gripping position to be able to execute grasping movement smoothly (Murata et al., 2000). As the reaching movement is executed, depending on the size, shape and orientation of the object, the wrist is rotated, fingers are opened and then they start to close even before the reaching movement ends with an accurate timing (Castiello, 2005; Dijkerman and De Haan, 2007).

The conception of surrounding space and peripersonal spatial map along with the required plan to achieve grasping and reaching are processed and integrated with highly interconnected neurons mainly within the region of primary motor, parietal and premotor cortex (Castiello, 2005; Cohen and Andersen, 2002). To perceive the surrounding space and integrate it in motor planning, the visual stream is used as a primary source of information. This information processing is a part of cognitive processes which yields to understanding the details of not only the spatial position but also shape, size and orientation of the objects. Comprehending these details brings the knowledge of objects itself too, that's one of the reasons why this pathway is known as "where" pathway, one of the two visual streams (two-streams hypothesis) (Eysenck and Keane, 2015).

There exists two different explanations about the recognition of spatial position of objects; a single continuous expanse or multiple spatial maps (Eysenck and Keane, 2015). According to classical neurology, the spatial information about surrounding space and body is encoded in parietal cortex by integration of multiple sensory modalities. In this view, spatial map is constructed in all directions where each object has its own location relative to one another. Hence, this information is shared with all related regions of motor control system. However, hypothesis of multiple spatial map for the conception has been gaining more attention lately (Graziano et al., 1994). This view indicates that spatial maps are constructed with different neural populations based on sensory modalities and motor functions. Recent findings about the parietal cortex strengths this view such that it has been found several neural populations within this region works in parallel, the representations are constructed during the interaction with surrounding objects and the locomotion map is encoded in the hippocampus while the prevailing position of the animal itself is encoded in the entorhinal cortex (Ekstrom et al., 2003; Hafting et al., 2005; Goldman-Rakic, 1988). Accordingly, this hypothesis suggests that as the body interacts with the environment, the recognition of spatial representations are built in parallel.

Planning of an action

Voluntary behavior is constructed by the combination of several cognitive processes, such as integration and transformation of sensory modalities, planning and execution of actions and short-term or long-term goals. A motor behavior is not only governed by the information on the physical properties of the objects due to sensorimotor integration that we have discussed so far. In addition, the planning of the sequence of actions is required to be selected among all possible alternatives. As discussed previously, high-level motor control (prefrontal and parietal cortical) areas that are involved in sensorimotor integration, also contribute to the planning of a motor behavior (Goldberg, 1985; Cui and Andersen, 2007). These regions are thought to be involved in the process of abstraction, decision-making and also anticipation of the outcome of the planned actions.

Regarding the dynamics of planning an action, our knowledge is limited. What has been known so far is that planning entails association of symbolic or semantic cues to certain actions. It indicates that voluntary behavior is partially guided by static or dynamic association rules that leads to behavior based on cues. The stochasticity or involvement of consciousness in

planning is beyond the scope of this thesis. Recent studies with animal experiments reveal the role of prefrontal and dorsal premotor cortices' in action planning, particularly the aspect of setting and applying association rules (Buneo and Andersen, 2006; Wallis and Miller, 2003; Wang et al., 2019).

In experiments with monkeys, it has been shown that particular neural populations within premotor cortices become active when monkeys were given associative cues, subsequently monkeys learned to perform a motor behavior that is related to predefined rules (Wallis and Miller, 2003). Monkeys have learned to associate particular behaviors, such as releasing or holding a lever, to predefined selection rules, either a spatial or semantic rule. This association between neural responses and selection rule indicates the functional role of these neural populations within prefrontal and premotor cortices. It has been claimed that it is being a part of the interpretation of the sensory information along with related cues to device the motor planning. One of the outcomes of these experiments is that dorsal premotor cortex is highly involved in the setting and applying behavioral rules to the planning of the motor action. To the extent of how appropriate the behavioral decision is evaluated with respect to the desired movement.

1.1.2 Middle-Level Motor Circuits

Voluntary motor control

A significant part of the cerebral cortex is dedicated to voluntary motor control. Thus, understanding the structure and function behind the association of these regions for voluntary motor control would contribute to comprehension of the organization of the brain. The importance of the voluntary motor control emerges from its distinctive role in survival. It is responsible for the interaction between the body itself and the surrounding space. Be it foraging food, protecting family members or itself or building tools to ease life conditions, all of which are the outcome of a voluntary movement. The ability to control upper extremities, independent from locomotion, enhances the chance of survival of individuals. In phylogenetic order, the neuronal circuits that are dedicated to control the upper extremities emerged from the older circuits of locomotion control (Shaw et al., 2008).

The Primary Motor Cortex (PMC) is particularly involved in initiation and generation of motor commands while associating the sensory input with related context. The decision of which voluntary behavior will be evoked is determined in this region as well as eliminating

other possible voluntary behaviors. Linking the contextual relationship between physical properties of surrounding environment and behavioral salience is part of functionalities in PMC. Besides executing already embedded motor commands, learning new behavioral strategies and improvement of existing actions are also the primary feature of the middle-level motor circuits (Li et al., 2001). Acquiring new motor skills or improving behavioral responses to a familiar stimuli further improves the quality of motor outcomes that are necessary to maintain survival. The nature of the voluntary behavior then can be described as the enhancement of motor actions with continuous experience. In addition, it is not limited to the generation or learning of motor skills for a particular muscle activity, instead motor control circuit is a combined process of perception, sensory integration, cognition, behavior and learning (Rizzolatti and Luppino, 2001; Kalaska, 2009).

Population coding in motor commands

Significant studies of PMC aims at understanding how motor commands are generated given corresponding sensory stimulus integration and continuous learning mechanisms. One of the findings about functional properties of PMC is the organizational architecture of the motor command generation. It has been shown that a motor map exists to lead the spatial directionality of the voluntary movement (Georgopoulos et al., 1982, 1983).

The conventional approach of coding for the integration of movement direction is to consider the motor map as a look-up table within which related pools of neurons are recruited to generate the motor commands selectively. Here, each spatial direction of movement is encoded exclusively by different sites. The motor commands that yield the required muscle activity are the combined activity of these selected neuronal groups. However what has been found indicates much more complexity. Georgopoulos et al.showed that it is in fact a distributed functional map in which each direction is simultaneously encoded in different neuronal pools and each contributes the spatial direction gradually (Georgopoulos et al., 1986, 1988). In an experiment with conscient monkey, different neurons within PMC were recorded while the monkey was moving his arm for different directions in which all movements start from the same central initial position. In Figure 1.2a, raster plots of the same neuron in PMC for 8 different directions were given. As it can be followed, this neuron has tendency to fire at a maximum rate when the movement was centered around 180 degrees and a gradually lesser rate around the preferred direction. It reaches the lowest intensity when the direction is the opposite of

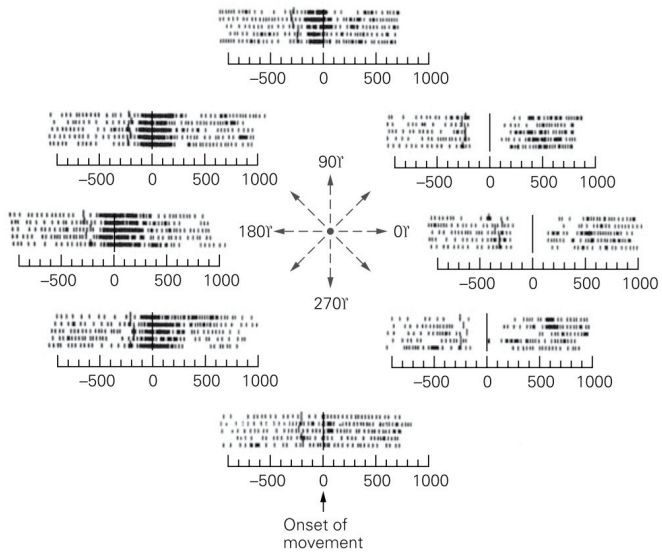
the preferred direction, which is 0 degree. Although the recordings of other neurons within the same region showed similar firing patterns, different neurons show distinctive preference for different directions. It is implied that there are individual neurons that respond correlated activity with their preferred directions (Evarts et al., 1983). Contrary to the expectation, nearby neurons show different preferred directions whereas neurons that share common preference have been found in different sites of the PMC for the arm map (Georgopoulos et al., 2007). As a result of this distributed spatial motor map within PMC, cells within this region show a broad range of activities depending on the direction of the arm during a voluntary reaching movement.

Despite this distributed motor map within PMC, vectorial addition of all single cells that fire given a directional movement show strong correlation to the movement direction itself, Figure 1.2b. In this figure, during each reaching movement for all eight different directions, the activity of all neurons are represented proportionally with their preferred direction with a thin black line. As shown, there is a clear alignment between the arm movement (dashed arrows) and the vectorial combination of all single cell activities (blue arrows). As a result, it has been shown that during a directional arm movement, a global unequivocal signal for the movement direction is encoded in the motor map of PMC with distributed neuronal activities which can be identified as a vectorial sum of related firing rate of neurons. Similar vectorial analysis of population coding reveals that the spatio-temporal pattern of PMC neurons shows similar continuous signal flow of the same population coding scheme for arm movements in which the direction of the arm is continuously changed (Moran and Schwartz, 1999; Paninski et al., 2004).

Kinematics and kinetics of movement

Population coding experiments show that PMC has direct implications over the computation of the arm movements trajectories. However, the execution of muscle activities that corresponds to specific arm movement requires not only the computation of movement direction but also evaluation of spatiotemporal form of the movement as well. Nevertheless, there are neurons within this region with direct connections to the spinal α-motoneurons and electrical stimulation of these neurons show that they can stimulate muscle contractions (Asanuma and Rosen, 1972; Moran and Schwartz, 1999). Given these connections, an important question is whether PMC sends motor commands that convey the information about the intended spa-

(a) Raster plots



(b) Vectorial directions

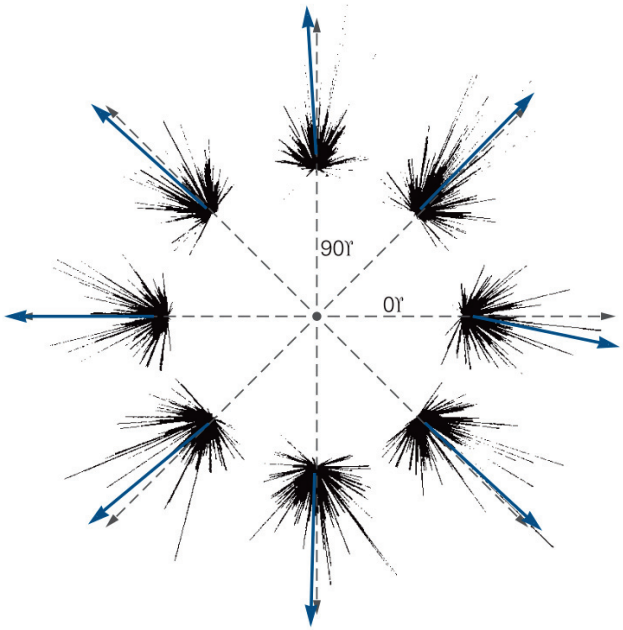


Figure 1.2: Population coding of neuronal populations in PMC. a. Raster plots of a single neuron during 8 different directional monkey arm movement. In each direction, several experiments conducted and the spiking activity of the neuron is given in each raster plot to show the trial-to-trial variability. b. The vectorial sum of each recorded neuron for all 8 directional arm movements. There is a strong positive correlation between vectorial summed of neuronal activities and an actual arm direction. Figure taken from (Georgopoulos et al., 1982)

tiotemporal form of the voluntary movement such that the kinematics (direction, amplitude, velocity and movement path), dynamic forces (external forces e.g. gravity) and the kinetics (causal forces and muscle activities) of the movement are encoded within these descending commands.

Single neuron recordings revealed that neurons within PMC are involved in the computation of the kinetics of the movement rather than the kinematics (Evarts et al., 1983). In this experiment, monkey needs to pull their wrist while a load is attached to it. The objective of the experiment is to exert compensation forces to the muscles in the opposite direction of the load which requires a constant kinematics of the wrist movement (direction, velocity and amplitude) whereas changing kinetics with respect to causal forces and muscle activities. Recordings in PMC neurons showed that related neuronal activities have increased with respect to their preferred direction in case of a load that is attached at the opposite direction, whereas a lesser activity is observed when the load was contributing to the preferred direction.

In another experimental study, the role of PMC in the kinematics or kinetics have been examined with isometric forces in which the subject is required to exert it against an immovable object instead of arm movement (Sergio et al., 2005). A linear relationship between isometric output force and the activity of recorded neurons in the PMC. The firing rate of these neurons have varied proportionately with respect to the necessary isometric force for all different range of directions while the directional population coding curves also remain similar to the ones with control cases. This experiment shows that neurons within PMC contributes to the computation of static and dynamic forces since this experiment doesn't require any movement during isometric force generation. In summary, PMC takes part not only in providing information about the kinetics of the motor outputs such as causal forces and corresponding muscle activities but also the intended direction of isometric force without the necessary magnitude.

Learning new motor skills

The knowledge of Basal ganglia being part of the motor control circuit stems from the relationship between diseases that are associated to movements and lesions in Basal ganglia (DeLong and Wichmann, 2007). For instance, Parkinson and Huntington diseases occur due to the abnormalities in dopaminergic neurons which are mainly located within the basal ganglia circuits. It is also known that the output of the basal ganglia reaches the motor cortex through thalamic projections (Alexander and Crutcher, 1990).

The Basal ganglia is composed of four main sub-networks; striatum, globus pallidum, substantia nigra and subthalamic nucleus. The inputs from different cortical regions, thalamus and brainstem arrives at the first principle structure of the basal ganglia, called the striatum. Input then is projected to the intrinsic components of basal ganglia through direct, indirect and hyperdirect pathways (Alexander and Crutcher, 1990). The major output structure of the basal ganglia is the globus pallidum, which projects the output mainly to the thalamus and brainstem. One of the nuclei of substantia nigra, which is called the Pars Compacta, contains the dopaminergic cells that mostly projects to the striatum which is known to be involved in sensorimotor activities.

The projections of different cerebral regions onto striatum show topographic organization, such that different functional regions pass through different pathways within basal ganglia and presume the topographic manner after projecting back to cortex through the thalamus, hence the name implies, basal ganglia-thalamocortical circuits. These different pathways are named based on their functional roles and originate in the frontal cortex, such as skelemotor, oculomotor, associative and limbic circuits (DeLong and Wichmann, 2007; Alexander and Crutcher, 1990).

Part of the knowledge about the functional role of the basal ganglia-thalamocortical circuits mainly arises from studies that are related to major motor diseases. Due to segregated anatomy of these circuits, malfunctioning or lesion causes different movement related disorders. In nonhuman primate experiments, the effects of a lesion at the output nuclei of Basal ganglia, globus pallidus, has been examined to assess the influence of this lesion on movement (Horak and Anderson, 1984). Based on these studies, it is now believed that Basal ganglia circuits takes the role in motor planning and sequential movement control (Desmurget and Turner, 2008). In addition to these studies, extracellular electrophysiological experiments indicate that the basal ganglia circuits are also involved in the action selection and assessment of these actions which is known as Reinforcement Learning (RL). The first findings of action selection can be traced back to patients with movement related diseases such as Parkinson and Huntington disease where the symptoms of these diseases are linked to movement initiation and control of movement itself. Furthermore, the relationship between the acquisition of reward or punishment and integration of new motor skills is observed in experimental studies of single cell recordings from basal ganglia circuits (Hollerman and Schultz, 1998; Holroyd and Coles, 2002).

1.1.3 Low-Level Motor Circuits

Periodic movements

The essence of movement is to provide animals a survival mechanism. Periodic movements to escape from predators and catch prey, or goal-directed behaviors to grasp food and hold tools for protection are the outcomes of the evolution of motor control circuit. Motor control for repetitive and periodic movements have evolved in many different forms such as swimming, flying, several forms of walking (biped, quadruped), and undulation. All these locomotion patterns have a common property, being periodic and repetitive which underlies the involvement of spinal circuits with limited control of high-level circuits. The engagement of high-level circuits takes place due to changing conditions in the surrounding environments in order to modify a stereotyped behavior. The nature of the neural control of locomotion triggers two main questions: What are the organizational principles of repetitive and sustained movements? How does the motor control circuit integrate sensory information to adapt these rhythmic patterns to changing conditions?

Research on the role of the spinal cord reaches back to the work of (Sherrington, 1952) in the early 20^{th} century. First insights about the division of labor between high-level and low-level circuits in motor control arise from experimental studies in which dogs without cerebral hemispheres generating steps and decerebrated cats walking on treadmill (Sherrington, 1910, 1952). He showed that a cat can express repetitive movements after its CNS is disconnected from its spinal cord circuitry. These experiments show that repetitive, stereotyped and reflexive behaviors are generated by circuits in the spinal cord, see Figure 1.3.

Other influential observations about locomotion were acquired by Thomas Graham Brown who conducted experiments with deafferented hind leg muscles of a cat. It has been shown that after the transected spinal cord operation, the rhythmic but alternating muscle contractions could be obtained. He concluded that there are reciprocal inhibitions between neural pools for flexor and extensor muscles which generates the sequential and rhythmic stepping patterns, which is now known as half-center oscillators, Figure 1.4a, (Brown, 1911, 1914). These early studies of locomotion can be summarized in four items which can also be regarded as the principal roles of low-level circuits in motor control

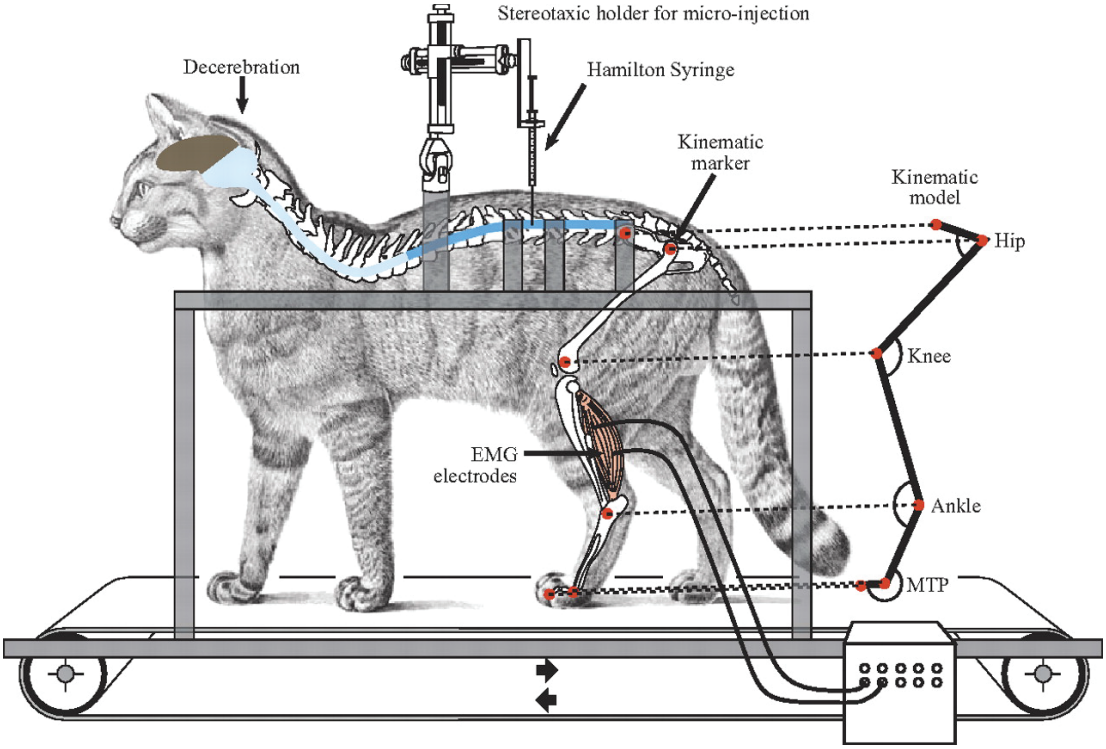


Figure 1.3: The experimental setup of a decerebrated cat on a treadmill. Due to sensory feedback triggered by the movement of the treadmill, a reflexive motor behavior emerges which leads the cat to be able to walk without the intervention of the descending modulatory signals. Figure taken in (Delivet-Mongrain et al., 2008)

1. Stepping and rhythmic movements don't require continuous modification of descending commands.

2. The rhythmic dynamic of stepping is entirely generated by neuronal pools within spinal cord.

3. Descending commands of the high-level circuits are involved in locomotion to adjust behavior to changing conditions in the environment.

4. The proprioceptive signals in rhythmic pattern generation is not a necessary condition, however it can regulate the rhythmic patterns.

A neuronal network that generates rhythmic motor pattern without sensory input is known as Central Pattern Generator (CPG). There have been several rhythmic behaviors observed

and identified due to CPG networks such as swimming, walking, flying and so on (Ijspeert, 2008; Duysens and Van de Crommert, 1998; Getting et al., 1980; Rybak et al., 2006). Studies on swimming behavior of the Lamprey by Sten Grillner provided important insights about the functional and organizational properties of the CPG, Figure 1.4b, (Grillner et al., 1995, 1987). Due to the complexity of the mammalian spinal cord circuits, most of the knowledge of the CPG have been obtained by invertebrates and lower vertebrates. Even though there is a lack of detailed information about spinal networks of humans, developmental studies indicate the existence of CPG in human infants who have the ability of generating rhythmic motor behaviors after birth (Forssberg, 1985).

Elementary unit of motor system: muscle

The building blocks of the motor control system not only involve neuronal networks that are responsible for the execution and planning of the movement, but also comprise muscles that transform the desire into physical movement. To achieve motor behavior, a broad variety of muscle combinations out of more than 600 muscles is required. There are two different types of muscles; striated and skeletal muscles. Skeletal muscles are the ones which exert force to move bones around skeletal joints, be it moving limbs, head or control respiration, or even facial expression. Muscles are connected to the bones with a connective tissue called tendons. Each muscle is composed of several muscle fibers, every one of them receives input from the α-motoneurons from spinal cord. Skeletal movement occurs only in two directions, either closing the joint angle or opening it, flexion and extension respectively. Muscles that act in the same direction is called synergist muscles, however, muscles that act on joints in different directions are called antagonists. The distinction between extensor and flexor muscles stems from the fact that muscles can only exert forces on one dimension, they can only pull. Due to these facts, CNS needs to regulate the extensor and flexor muscles in order to generate coordinated movement such that contraction of flexor muscle requires extensor muscles to relax. Depending on the strength of relaxation of the antagonistic muscles, CNS can modulate the speed of the movement (Bear et al., 2007).

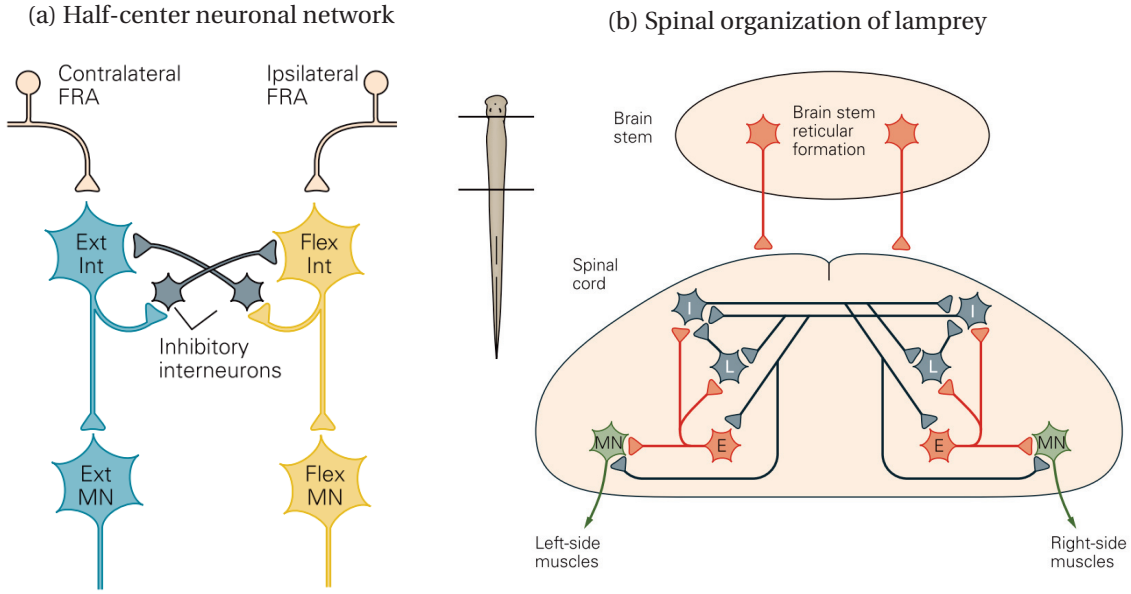


Figure 1.4: a. A mutual inhibition mechanism between two spinal interneurons to generate rhythmic stepping movement. b. Each segment of the lamprey body is controlled by a neuronal network that receives modulatory signals from the brainstem. Figure taken in (Kandel et al., 2000; Grillner et al., 1995)

α-motoneurons innervate the muscles through the ventral horn of spinal cord and sensory feedback arrives at spinal cord via dorsal root. Besides α-motoneurons, there are other motor neurons called gamma motor neurons. While α-motoneurons innervate the extrafusal muscle fibers, gamma motor neurons send their input to intrafusal muscle fibers. Force generation in muscles are directly controlled by α-motoneurons. An α-motoneuron can innervate more than one extrafusal muscle fibers, which collectively called motor unit. A movement in one joint is a combination of several motor unit recruitment by CNS. Nevertheless, a single muscle is coordinated by a pool of α-motoneurons which is called motor neuron pool.

There are three different sources of input that control the activity of α-motoneurons. First, the sensory input that comes from muscle spindles through dorsal root, then descending axons from the brainstem and motor cortex, and finally interneurons of the spinal cord. Sensory input that conveys the information about the state of the muscles and body, is carried by Ia axons. They wrap around the muscle fibers and detect the changes of length and velocity (proprioception). One of the important features of the Ia axons is that they are myelinated and they are among the fastest and largest axons in the body therefore they excite the α-motoneurons and interneurons very rapidly (Kandel et al., 2000).

There are two types of interneurons in the spinal cord from functional point of view, excitatory and inhibitory interneurons. The major inputs that activates the α-motoneurons comes from spinal interneurons. A critical role of inhibitory interneurons is to be responsible for myotatic reflex such that controlling the activity of antagonistic muscles coherently. It is also known as reciprocal inhibition. Myotatic reflex occurs due to the connectivity pattern of these inhibitory interneurons. An excitatory input coming from Ia axons triggers the activation of related inhibitory interneuron which in turn inhibits α-motoneuron that innervates the antagonistic muscle. Similarly, excitatory interneurons also take part in reflexive behaviors, such as flexor reflex (a.k.a withdrawal reflex). It is a reflex which is triggered by pain receptors and activated by several motor neurons therefore it can generate a very quick response given a possible damaging stimuli (Poppele and Bosco, 2003; Windhorst, 2007).

The last part of the input that arrives at α-motoneuron is the descending commands that modulate the α-motoneurons activity in order to initiate and control voluntary movement. Descending commands does not only innervate α-motoneurons but also spinal interneurons as well.

1.2 Degrees of freedom problem and related hypotheses

Due to the biomechanical properties of the motor control system, there are multiple possibilities of obtaining the same motor goal with different motor control states. The possibility of several movement trajectories in musculoskeletal system for the same goal is known as the degrees of freedom or motor equivalence problem and it is first defined by Nikolai Bernstein (Bernstein, 1966). The problem indicates that there isn't a one-to-one correspondence between a desired movement and a kinematic solution to that problem. CNS chooses a solution among abundant combination of muscle recruitments and joint coordinations. Notwithstanding, CNS continuously adapts its solutions to changing conditions given that the body is under constant development and aging.

The motor equivalence problem stems from the fact that there are redundancies in almost all parts of the motor control system, such as each joint is controlled by multiple antagonistic muscles. Besides the redundancy of the muscle control, there are kinematic redundancy as well, such that a movement can be obtained with different joint trajectories, velocities and accelerations. The redundancy becomes more complicated within the neural controller, for instance a muscle receives a stimulus from a bundle of α-motoneuron as well as a single

α-motoneuron is stimulating more than one muscle fibers (Flash and Hochner, 2005).

Selecting a movement among near infinite amount of solutions is the source of difficulties for understanding motor control system. A theoretical explanation about how CNS solves the motor equivalence problem requires a reduction of the biomechanical and neural control redundancies (Sporns and Edelman, 1993). Based on this approach, there have been several hypotheses suggested such as muscle synergies, equilibrium point and threshold control, force control and internal models, uncontrolled manifold and last but not least Optimal control theory (OCT) (Ting and McKay, 2007; Feldman, 1966; Asatryan, 1965; Ostry and Feldman, 2003; Scholz and Schöner, 1999; Scott, 2004).

1.2.1 Muscle synergy hypothesis

The first ideas regarding muscle synergies trace back to Nikolai Berstein (Bernstein, 1966). He claimed that functionality behind the muscle synergies is to create a strategy to tackle with redundancy problem of muscle control. Recent findings support his proposal that CNS recruites a fixed group of muscles together each time it generates the same movement (Torres-Oviedo et al., 2006; d'Avella and Bizzi, 2005; Ting and Macpherson, 2005; Krishnamoorthy et al., 2003). Muscles are constrained within the same groups in which they are recruited by the same neural command. The mechanism behind muscle synergies is that, a complex motor behavior (e.g. locomotion, postural control, grasping with fingers) is a combination of summed activities of muscle groups. The advantage that muscle synergy brings into the motor control is that the neural control signal is reduced to the same command which activates several muscles at the same time coherently. It has also been shown that muscle synergies are robust and have the ability to be generalized (d'Avella and Bizzi, 2005).

Electromyography (EMG) data obtained from frogs and humans demonstrated a dimensionality reduction due to muscle synergy mechanisms (Ting and McKay (2007); d'Avella et al. (2003)). As it is shown with several dimensionality reduction techniques (e.g. Principal component analysis, Non-negative matrix factorization) that are applied to these EMG data, muscle synergies provide low-dimensional representation of the corresponding movement (Tresch et al., 2006). It can be claimed that muscle synergies are recruited by CNS as eigenvectors of the related movement. Research on stroke patients also strengthens the idea behind muscle synergies. It was observed that subjects with stroke history uses lesser muscle synergy mechanisms compared to healthy subjects for similar motor performance (Roh et al., 2012). The

idea of muscle synergies aim at reducing the dimensionality of the low-level motor control to a combination of summed activity partially addresses the Bernstein's problem of motor equivalence.

1.2.2 Equilibrium-point hypothesis and threshold control

The closed-loop between muscle sensory organs and the muscle activation indicates that the motor control is primarily maintained at the very low level of motor hierarchy. The importance of low-level closed-loop motor control is first indicated by Sherrington who claimed that spinal reflexes are more than stereotypical responses to a given unequivocal stimulus, instead they form the basis of motor coordination (Sherrington, 1910). Further studies on Sherrington's idea of modular reflex circuits paved the way to the Equilibrium-point hypothesis (EPH) (a.k.a Threshold control theory) (Feldman, 1966, 1986).

According to the EPH, CNS generates movement while maintaining a transition from one equilibrium state to another, mainly in muscle states. This transition occurs either voluntarily or a change in the surrounding environment triggers this equilibrium shift. The idea of movement is the consequence of interactions between the body and its environment also aligned with a well known developmental theory, known as Thelen's Dynamic System Theory (Thelen and Smith, 1996).

In EPH, a movement is obtained while CNS maintains the transition of equilibrium points in each consequent state of the movement in order to guarantee a zero force field that acts at the joints. he balance condition among opposite muscles along a desired behavior can also be regarded as satisfying. To achieve the next state of the movement, CNS controls the opposing state of muscles with respect to change in muscles that create the force in movement direction. With this regard, the stimulus of α-motoneurons changes the force-length relationship of the corresponding muscles, therefore it creates a shift in the equilibrium point of the entire body which has to be compensated by the next action to maintain equilibrium stability. This hypothesis indicates that CNS estimates the sequence of actions for a desired trajectory while taking sensory information of muscle dynamics into account, not necessarily the limb dynamics directly (Feldman, 1966). However, there have been studies indicating contradictions between EPH and cerebellar internal models (Hinder and Milner, 2003). Despite the evidence obtained for each theory, it is unknown how internal models recruit spinal modular reflex circuits.

1.2.3 The uncontrolled manifold theorem

One of the theories that opposes the elimination of redundancy in motor control is called the Uncontrolled manifold theorem (UMT) (Scholz and Schöner, 1999). It has been claimed that the exploitation of available motor abundance leads to the improvement of the performance of the motor task, known as the principle of motor abundance (Gelfand and Latash, 1998). It is claimed that the phenomenon of near-infinite amount of possibilities available to motor control brings an advantage rather than a caveat (e.g. curse of dimensionality). It can be claimed that the UMT is inlined with the principle of motor abundance.

The UMT hypothesis states that all the components of the motor system is necessary to ensure flexibility and stabilization, however not all of them are under the direct control of CNS. Thus abundance of redundancy is steered instead of restricted despite the opposite claims from earlier theories mentioned above. It emphasises the concept of separation between variables that are controlled and uncontrolled by CNS. The controlled variables are defined as a set of variables in Jacobian space which are perpendicular to ones that are not relevant to the task. In order to achieve a desired movement, there is a gradual control over variables such that highly relevant ones are tightly controlled whereas others are left free to vary.

In (Scholz and Schöner, 1999), the control over muscles is examined on a sit-and-stand test to evaluate the hypothesis. It has shown that there is a gradual transition of control over muscles such as sagittal center of mass was mainly controlled movement, whereas the horizontal head position was controlled to a lesser extent. To highlight this transition, it showed that joints that are irrelevant to movement appears to be merely controlled, e.g. hand motions and vertical position of head.

Experiments with patients with Down syndrome highlights the principles behind UMT (Scholz et al., 2003). It was shown that the task performance of Down syndrome patients is improved with an experimental setup that considers the hypothesis of UMT. It has been claimed that the task performance has increased due to an increase in the variance of variables that are orthogonal to uncontrolled manifold which is accompanied by the decrease of variance that are perpendicular to the uncontrolled manifold. Since an increase in the task performance was obtained due to reduced errors, mainly in the task-specific dimensions, it is claimed that there is an alignment between UMT and the hypothesis of optimal feedback control (Todorov and Jordan, 2002).

1.2.4 Optimal control theory

Lately, OCT has been gaining attention to tackle Bernstein's problem of motor equivalence (Scott, 2004; Todorov and Jordan, 2002; Todorov, 2004; Wolpert and Ghahramani, 2000; Guigon et al., 2007; Uno et al., 1989; Hoff and Arbib, 1993; Dingwell et al., 2004). The hypothesis of OCT relates the solution of motor equivalence to the objective of minimizing a certain cost in a principled way.

For instance, the optimal control hypothesis aims at giving scrupulous explanations to the details of motor control system such that how CNS deals with redundancy, uncertainty and link between invariance and task performance (Guigon et al., 2007; Körding and Wolpert, 2004; Harris and Wolpert, 1998). The distinctive feature of OCT from other hypotheses is that the explanation of motor behavior is connected not only to evolution but also motor learning by the definition of objective functions (a.k.a task performance, cost, cost-to-go functions).

There were several different cost functions proposed depending on the task of the movement, for instance minimum energy consumption has been associated with the locomotion tasks or obtaining a precise movement trajectory has been linked to the task of reaching an object (Todorov, 2004). There were also other attempts to design more complex cost functions mainly to explain how OCT could decipher the relationships among each level of motor control hierarchy.

To that extent, the hypothesis of OCT on two main parts of the motor control system are: an internal model which estimates the current state of the body while integrating the afferent sensory feedback and the commands of the motor system (McNamee and Wolpert, 2019) and error correction during motor learning with adjusting only necessary components instead of all related parts of the motor system (Todorov, 2004). The integration of sensory feedback explains how CNS creates the association between sensory signals and motor commands and the selective error correction relates the OCT to the UMT. There has been several studies supported the idea of OCT, for instance spinal reflexes of a hindlimb of cat (He et al., 1991), posture control (Kuo, 1995), and also volitional motor control (Meyer et al., 1988).

Internal model estimation:

One of the hypotheses to explain the nature of behavior is based on the idea of small-scale models, first suggested by Kenneth Craik (Craik, 1952). Craik suggested that organisms create

a mental representation of their surrounding environment and themselves to be able to comprehend the outcome of all their possible actions. Within this representation, they can integrate the past experiences and project them to future plans. This hypothesis also has been widely discussed in neuroscience, known as constructive predictive models or internal models (Miall and Wolpert, 1996; Kawato, 1999). Several experiments suggest that humans use prior statistics to build a representation of the external world and their musculoskeletal dynamics (Körding and Wolpert, 2004; Scott et al., 2015). This internal representation of the body and surroundings constructs the basis of explanation for OCT to tackle the Bernstein's motor equivalence.

The internal models provide the framework of how task objectives and external reality are combined as a result of sensory integration and movement control within brain. With this regard, probabilistic inference (or Bayesian inference) and optimal control have been suggested to address the construction of internal models (Todorov, 2004). In (Wolpert and Kawato, 1998), it has been suggested that Bayesian inference has the ability of explaining how CNS estimates the probability of consequent states, thus creating an internal model.

Based on Bayesian rule, they claim that the probability of current sensory feedback is represented as likelihood, and CNS estimates this likelihood with a sensory forward model which in turn can be used to predict the upcoming sensory feedback. In this study, multiple pairs of inverse and forward models have been used to generate the movement control in inverse models based on the discrepancy between the predicted and actual sensory feedback in forward models. This modular and parallel neural architecture is one of the first attempts to rigorously model the relationship between sensory feedback and movement control with the idea of internal models.

Sensorimotor control:

Once an internal model has been estimated, this model then can be used to control the motor commands in order to achieve the desired movement. As probabilistic inference is a widely studied concept to address state estimation under uncertainty and noisy sensory feedback, optimal control has been suggested to explain the sensorimotor control to obtain the motor commands or to broadly construct the control laws. The objective of OCT is to explain the motor behavior from an optimization point of view and it relies on Bellman's principle of optimality (Bellman, 1966) therefore it is an accepted and rigorous mathematical framework

to study sensorimotor control.

The combination of estimated internal model and sensorimotor control laws give rise to the emergence of voluntary behavior. Studies have been conducted regarding how this interaction could be performed in motor control system from OCT perspective (Todorov, 2005; Todorov and Li, 2003). In these studies, OCT models have been suggested to explain how a sequence of motor commands could be generated to obtain a desired trajectory in arm movements with different cost functions (e.g smoothness, time derivative of acceleration, joint torques) (Uno et al., 1989; Flash and Hogan, 1985).

One of the important findings of these models, supported by experiments with humans, is to provide an explanation for the role of sensory feedback signals in movement control. For instance, it has been shown that sensory feedback in the context of OCT is to bring the hand back to a desired trajectory under disturbance (Scott, 2012). In addition, there are OCT models that posits the human hand characteristics, such as bell-shaped speed profile (Todorov and Jordan, 2002).

In the context of motor equivalence problem, OCT points to several relationships between different structures of the motor control system, such as the role of sensory feedback, muscle recruitments and abundance redundancy. For instance, the ability of the CNS to adapt its behavior for an uncertain perturbation is explained by OCT with the role of sensory feedback on control law. Aligned with the UMT, OCT models explain how a correction of hand movements occur as movement deviates due to the disturbance, see Figure 1.5 (Scott, 2012). In this experiment, subjects were asked to reach a target which can be either narrow or wide. In each case, subjects had the tendency to create hand movements that were always close to a straight trajectory regardless of the length of the target.

In case a disturbance occurred during the movement generation, it was observed that only subjects that were asked to reach a narrow target produced the corrected movement, whereas there were no significant corrections for a wide target. The behavioral explanation about this experiment is that since a perturbation doesn't change the performance of the task in the wide target, subjects omitted the correction. These studies reveal the underlying structure of goal-directed behaviors which requires physical knowledge of the limb, and also the changes in external reality. As it has been shown in this experiment, sensory feedback and optimizing a cost function forms the basis of voluntary sensorimotor control.

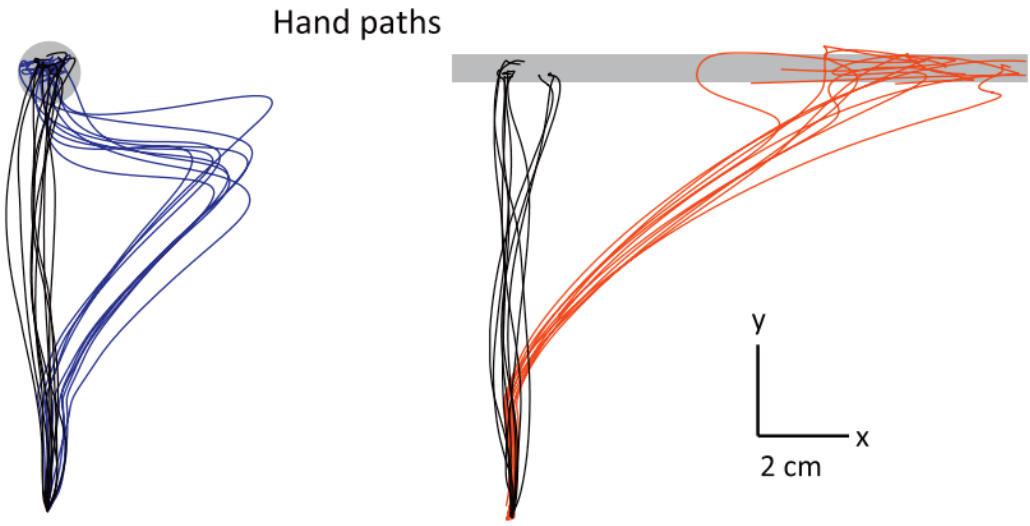


Figure 1.5: Behavioral experimental study to reveal the underlying mechanisms behind sensorimotor control. Human subjects were asked to move their hands to a target which is either wide or narrow. During the movement, subject movements were disturbed and end position of hand is compared. Figure taken from (Scott, 2012)

1.3 Current state of research

There is a wealth of research and approaches to understand the different aspects of the mammalian motor system in general and because of its high biomedical relevance on the human motor system in particular. With the advent of highly realistic animations in movies and computer games, interesting new approaches have also been developed.

Our approach is based on two main assumptions: First, that the different neuronal circuits and pathways involved in the execution of a voluntary movement need to be able to support periodic as well as non-periodic movements. We will therefore not consider neural models, based on coupled oscillators, that are common in the study of multi-legged locomotion or undulatory swimming movements (Grillner, 1985, 1975, 2011; Ijspeert et al., 2007; Ijspeert, 2008). Second, we assume that the neural control circuits must learn the kinematic as well as dynamic properties of the musculo-skeletal system, as these are highly variable throughout the life of an animal.

1.3.1 Musculoskeletal simulations

Musculoskeletal simulations usually focus on one region of interest such as lower or upper extremities. Models of movements of the lower extremities typically focus on periodic movements such as the emergence and transition between different gaits as well as jumping, hopping and running (Pandy et al., 1990; Geyer et al., 2003; Lim et al., 2003; Ackermann and Schiehlen, 2006; Geyer and Herr, 2010). Models for the upper extremity of the human body are more versatile. Some focus on kinematic studies of the human arm and its control of motion coordination (Seth et al., 2003; Garner and Pandy, 1999), some consider the contribution of the rather complex shoulder anatomy on arm movements (Van der Helm, 1994). Other upper extremity research is dedicated to hands, arms and finger movements (Flash and Hogan, 1985; Uno et al., 1989; Harding et al., 1993; Soechting et al., 1995; Rosenbaum et al., 1995, 2001; Santos and Valero-Cuevas, 2006; Valero-Cuevas et al., 2007; Biess et al., 2007; Friedman and Flash, 2009). Apart from dynamics and kinematics of the movements of human extremities, the control scheme of the neural motor control system is also a central interest of musculoskeletal studies (Van der Helm et al., 2002). Based on these studies, one of the lately emerging fields is the transformative studies where next generation prosthesis and orthopedic solutions are provided (Cavallaro et al., 2005).

The musculoskeletal/biomechanical simulation studies are broadly divided into two different approaches: First approach relying on data collected through human/animal motion capture experiments (Multon et al., 1999) or synthetically generated data via computer animation methods. The objective of these data oriented simulations is to integrate the observations of human movements in biomechanical simulations to obtain the muscle forces via calculating the inverse dynamics (Happee and Van der Helm, 1995). The second approach uses the mathematical modelling of forward multibody dynamics (Xiang et al., 2010) in the case of simulations of human walking behavior. Essential detail of the musculoskeletal simulations is to consider the dynamical properties of multibody systems. To obtain physiologically plausible simulated movements in these multibody dynamics, it is required to use an accurate muscle model. Depending on the question of interest, the muscle model that is used in these multibody simulations can vary; the coarse movements are accurately obtained with Hill-type muscle models (van den Bogert et al., 1998; Winters et al., 2012) on the other hand the physiological details can only be studied with models that cover the physical and three dimensional structure of the muscles such as used in (Röhrle et al., 2013).

A big challenge of biomechanical movement control is its redundancy: The number of joints to be controlled in the musculoskeletal system are significantly less than the number of muscles that exerts force into those joints. Therefore accurate models of the musculoskeletal control must solve the ill-posed problem of recruiting the right set of muscles for a particular joint movement. Several optimization methods have been used to address this problem, such as (Silva and Ambrósio, 2003; Quental et al., 2012) where each of these studies rely on some objective criterion based on physiological evidences. In (Umberger et al., 2003; Ackermann and Schiehlen, 2006), the metabolic energy consumption is chosen as the objective to be minimized, whereas the endurance system is the criteria to be maximized in (Crowninshield and Brand, 1981). The second approach, which is the most common way of studying biomechanical musculoskeletal simulations is forward dynamical simulations, where the goal is to actuate the musculoskeletal system with prior muscle stimulation, as in (Anderson and Pandy, 2001). Although this methodology provides more accurate motions since it avoids integrating the accumulated errors in motion capture procedures, it requires higher computational cost due to the fact that it is necessary to calculate the precise muscle stimulation. For instance, in order to study human body locomotion, the number of muscles to be stimulated by a range of predetermined time can be approximately a couple of dozens which can turn into a very expensive optimization problem (Garner and Pandy, 2001).

1.3.2 Reinforcement Learning and Neural Networks

The use of RL for musculoskeletal control has been gaining attention among several studies (Jaśkowski et al., 2018) along with metaheuristic optimization methods (Lee et al., 2014). A common strategy in RL studies is to obtain a solution that maximizes the cumulative reward regardless of the movement. Most training schemes for musculoskeletal models use reference motion learning, similar to the motor learning by imitation, where the reference motions are created by humans and recorded with motion capture techniques, others use hand crafted motion captured data such as in (Geijtenbeek et al., 2013; Coros et al., 2011). By contrast, we will present an approach that uses RL to learn movement strategies that are optimal according to OCT.

Recently, with the integration of RL into Neural networks with more than two hidden layers (Mnih et al., 2015) (a.k.a Deep RL), there has been significant improvements and successful solutions for the control of high-degree articulated robots and musculoskeletal systems (Peng

et al., 2018, 2017; Lee et al., 2019). It has been shown that these models have the capability of reproducing highly complex human behaviors, such as walking, flipping, reaching and grasping. The performance of these models highly depends on the construction of the control architecture, for instance (Peng et al., 2017) showed that different actuators significantly alter the success of these simulations. It has been reported that muscle and Proportional-Derivative controller succeeds over torque controller. However, due to excessive number of parameters in muscle controller, the policy learning of these musculoskeletal systems with Deep RL methods is more challenging and it requires longer training procedures than torque control architecture. It also has been reported that a promising policy learning for a musculoskeletal running is achieved with a similar learning architecture (Kidziński et al., 2018), however in 2D dynamics.

The problem of scalability of meta-heuristic optimization methods, such as stochastic optimization methods, limits the possibility of studying complex dynamic models. However, following the development of the Deep RL methods, revolutionary advances have occurred in musculoskeletal learning, where higher degrees of freedom has been extensively started to be considered successfully. Unlike the previous problem definitions (Lee and Terzopoulos, 2006; Sok et al., 2007; Yin et al., 2007). Deep RL methods are not only applied to musculoskeletal control problems but they have also demonstrated great success in torque controlled dexterous robots, including walking and running problems (Peng et al., 2017), as well as control of different morphologies (Won et al., 2017). Apart from plain implementation of Deep RL methods, there also has been an attention to develop hierarchical control architectures to study sparse reward problems (Levy et al., 2018).

Despite the great success of Deep RL methods in musculoskeletal simulations with high degree-of-freedom and high number of muscles, Deep RL methods still don't scale up to the level of human motor control with excessive degree-of-freedom and abundance muscle. The concern about these methods rise due to the fact that Deep RL methods are not sample efficient, that is, they are too slow to match up with exponential growth of the complexity of the human musculoskeletal simulations. However, recent improvements in Deep RL, namely Episodic Deep RL (Botvinick et al., 2019; Gershman and Daw, 2017; Pritzel et al., 2017) and Meta-Rl (Andrychowicz et al., 2016; Finn et al., 2017), encourages higher complexity of the musculoskeletal simulations.

1.3.3 Optimal Control Models

The formulation of boundary value problems with OCT has been applied to a variety of different biomechanical musculoskeletal simulations where the idea is to obtain the timely sequenced actuator values, either it can be torques to be applied to the joints or stimulus of muscles to generate force (Ackermann and Schiehlen, 2006; Pandy et al., 1990). The emphasis on these optimal control formulations is given mainly to the definition of objective function as it is required to match to the physiological findings and biologically motivated. Such an implementation can be found in (Uno et al., 1989) where the grasping motion of an arm is simulated while considering an objective function to minimize torque generation in the joint level. In addition, there are hypothesis where the objective function of a human movement optimization is to obtain minimum jerk of the desired movement as studied in (Flash and Hogan, 1985). Apart from plain torque or jerk minimization, there has been several other objective functions that are proposed such as energy consumption. The objective functions are developed with motivation of biological relevance behind the movement generations (Soechting et al., 1995; Biess et al., 2007) or comfort of limb movements. These biological phenomena are considered as minimization criteria in the optimal control formulations (Rosenbaum et al., 1995, 2001). The promising outcome of the consideration of comfortable movement search is that the movements that diverge from the comfortable solutions found by the numerical optimization can be studied as the identification of possible movement disorders.

The overall goal of these optimal control approaches is to be able to obtain a solution for the musculoskeletal system and compare these movement trajectories with collected human motion data. One of the outcomes of this approach is to identify the biologically plausible constraints in the movement generation, such that several objective functions are evaluated and control hypothesis are built on the basis of these findings. Since the constraints and biomechanical conditions can be directly linked to the equations of motion and therefore to the problem formulation, it is clear this approach has significant advantage of movement disorder studies. For instance recently with the availability of larger computing resources one of the optimal control methods, trajectory optimization, has been used to model the muscles in volumetric details that will allow us to understand some of the details of movement disorders (Lee et al., 2018) as well as investigation of human locomotion patterns (Umberger and Miller, 2018).

1.3.4 Motor Control Models

Recent studies of motor control have been focusing on different parts of the problem, either in a behavioral perspective or understanding the neural phenomena. Recent studies of behavioral analysis of motor control also investigate the motor control system from separate point of views, such as there are studies dedicated to comprehend the stereotypical trajectories of arm movement (Scott, 2004), the velocity profile of human arm (Todorov and Jordan, 2002) as well as the ability for the motor system to learn new motor skills (Miall and Wolpert, 1996; Sanner and Kosha, 1999). On the other hand, from a modeling perspective there are also distinctive studies to attempt modelling the motor control system, for instance modeling the neural activities in the level of spiking neural network modeling (Churchland et al., 2012), or activity profile of motor system at the level of population coding (Shenoy et al., 2013; Churchland et al., 2010). There has also been attempts to model the motor control system at the large-scale with different levels of complexity (DeWolf et al., 2016; Eliasmith et al., 2012).

Apart from modeling the motor system at the network level, there also are studies to integrate the available data on the part of the motor control system. Experimental studies point out that descending commands of motor system is represented by highly complex beta regime oscillations (Murthy and Fetz, 1992; Sanes and Donoghue, 1993). In (Heitmann et al., 2015, 2013), these beta band oscillations have been modelled to study the execution of movements in the muscle level. They aimed at answering how beta band oscillations are integrated in the spinal cord to construct the muscle control for voluntary behavior.

Another research direction in motor control is to integrate a part of the functional model of motor system into robotics. For instance, there are studies focusing on modeling cerebellum at the spiking level and simulating these networks in robotic applications (Carrillo et al., 2008; Casellato et al., 2014). In addition, there has been attempts to model modular spinal reflex circuits with different levels of complexity (Schumacher and Seyfarth, 2017; Marques et al., 2014).

1.4 Problem statement

From the computational point of view, musculoskeletal control is the problem of controlling a dynamic multi-body system with many degrees freedom. Moreover, the units are redundant and highly nonlinear in the sense that each skeletal joint is controlled by several antagonistic

muscle pairs along with highly nonlinear dynamics of each muscles. In addition, the underlying structure of the neural motor control system comprises several neural circuits, each of which has its own role on the overall control architecture of the musculoskeletal system. The interactions between related motor circuits and the definitive role of each of them are yet to be revealed, and it makes dissection of the neural motor control system an arduous scientific question.

Given that a single joint of a skeletal system is controlled by several antagonistic muscle units, the problem becomes almost intractable in case of a multi-body joints control of a musculoskeletal system, for instance a coordinated movement of an arm, with 9 degrees of freedom, requires to be controlled by more than a dozen muscles. Thus, the goal of reaching a single point in space is the result of finding a specific trajectory in an high dimensional muscle space. Nevertheless, the movement generation problem can be considered as an ill-posed problem given that a simple movement itself is a movement in one-dimensional space while the neural circuit and musculoskeletal system acts in an high-dimensional space.

An example of steering a wheel can be given to highlight the underlying structure of the control problem. While the goal of steering a wheel is to achieve control in a two dimensional space, meaning that the end effector moves in restricted coordinates, nine degrees of freedom exist for the joints of the arm, although the controller of those joints, muscles, are acting in a multiplied dimensional space of joints by the number of muscles. Thus, examining the system from end effector to the controller behind it yields the increasing complexity of the problem.

Given these challenges of motor control, such that it is a high-dimensional system with redundant muscle actuation as well as it is controlled by a complex neural network structure, one of the strategies of understanding the design principles of motor control would be to dissect the functionality of different part of the system. At the same time, understanding how CNS generates a smooth movement requires to combine different knowledge from diverse disciplines such as robotics, machine learning, computational and experimental neuroscience, control theory and last but not least modelling studies. In this thesis, we aimed at bridging these two different aspects of the motor control problem, while on the one hand we focused on how to separate individual components of the motor control system from functional point of view, on the other hand we prepared an optimization and learning framework with bringing different methods together from different disciplines. Therefore we claimed to seek a contribution to different aspects of research, such that we developed an optimization and

learning framework which can be useful for robotic simulations and musculoskeletal control problems, at the same time we developed a computational model of motor control system to test the hypothesis of optimal control. The questions to be answered in the thesis can be summarized as follows:

1. Can we solve the redundancy in musculoskeletal control problems, in such a way that the solution is optimum?
2. What are the principle roles of reflex circuits in spinal cord, how can we obtain these circuits with learning rules that have biological relevance?
3. How can we integrate proprioception signals intio motor control circuit to create a closed-loop system?
4. Can we use the idea of reverse engineering to model the neural motor control system for movement generations?

To answer these questions, we will start at the dynamics of the movement and then we try to *reverse engineer* the underlying neural structures.

1.5 Overview of the thesis

After the introduction, in Chapter 2, we will present the theoretical foundations of our optimization framework for skeletal movement trajectories, based on a combination of RL and OCT. We briefly introduce the basic concepts of RL and how it can be used to learn movement trajectories, based on the difference between a generated movement and a reference trajectory. We then introduce the main concepts of OCT which we will use to generate the reference trajectories for our model. After that, we discuss how an optimal control problem can be solved numerically by non-linear programming methods. Finally, the chapter presents an actor-critique RL algorithm, that trains a network to generate optimal movement trajectories for a given task.

In Chapter 3, we present a number of applications of our learning framework. The first example is a 2D model of a human arm. We derive the dynamic equations and boundary conditions for the musculoskeletal system of the arm and present numerical results for periodic (arm swing) as well as non-periodic reach-to-point movements and compare the results of the optimal

controler to the results of RL. The next model is a more complex model of humal locomotion, with 7 links and 18 muscles. Again, we derive the dynamic equations and also examine the integration of biomechanical properties of the musculoskeletal system, mainly the nonlinear dynamics of the system e.g antagonistic muscle pairs, and overactuation due to the muscle control while leaving the muscle recruitment, fatigue and synergies as future research for future work.

Chapter 4, presents the integration of musculoskeletal control into an computational neural motor controlframework, that comprises spinal reflex cuircuits and high-level motor control areas. We will investigate how spinal reflex circuits can control α-motoneurons to stimulate the muscles that need to be involved in a particular movement. We use the idea of *muscle twitching* or *motor babbling* to let the network explore its dynamic properties and to learn the connectivity required to initiate muscle movements, using a Hebbian learning rule. After this learning step, the network can appropriately stimulate the extensor and flexor muscles for a particular joint. To assess the performance of the model, we will use the musculoskeletal human arm model. We demonstrate that the model will generate movements for given goal positions without training the model for that specific goal position. Finally we will study the effect of modulatory descending commands to the spinal reflex circuit. First, we use randomly generated descending signals to train the descending connections. Then, we use the reference trajectories to generate more realistic descending signals to the spinal reflex circuits which then trigger movement of the musculoskeletal system. We show that the proposed neural motor control model can generate the movement to an arbitrary goal position.

Chapter 5 will conclude the thesis with a discussion of our achievements and possible future work.

2 Learning and Optimization Framework

In this chapter, we will develop a learning framework where we can study the neural control of movement. One of the goals is to build a framework where desired movements can be used as supervision signals for a computational motor control model. In this framework, either kinematic information of a human/animal or synthetic movement data can be used as a source signal. The kinematic trajectory found by motion capture techniques or optimum trajectories found by OCT can give us the information about joint movements at the torque level, whereas the first goal of the reverse engineering the motor circuit is to find out what are the force generations with redundant muscles. This in fact can be regarded as an ill-posed problem, where the objective is to identify the multiple driving forces coming through redundant extensor and flexor muscles which yields a single dimensional behavior. In order to examine the optimality of the movement trajectories, we focus on the OCT to find out the movement trajectories in the level of joints. We use these signals as reward function for RL to find out the level of α-motoneuron activity which is in charge of controlling muscles. Solution of this ill-posed problem yields the joint control with multiple extensor and flexor muscles. The objective of utilization of the RL in this thesis is to find time dependent stimulus for muscle contractions given a desired trajectory of the musculoskeletal system. Therefore, the principle idea behind the proposed learning and optimization framework can be summarised as to utilize optimum joint trajectories as supervised signal and use them to obtain a policy function for muscle control, which is sketched in Figure 2.1.

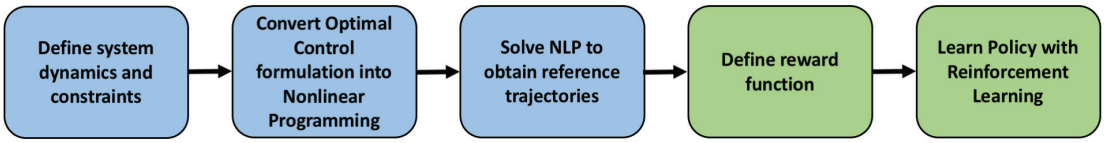


Figure 2.1: An overview of the proposed learning and optimization framework. The objective is to obtain a desired movement in musculoskeletal system with reference trajectories. To obtain these reference trajectories, a simplified model is used in numerical solution of optimal control formulation. The framework requires to write the simplified system dynamics with related constraints. To solve an optimal control problem, it needs to be translated into a Nonlinear Programming (NLP) where a numerical solver is used to obtain desired trajectories. In the second part of the framework, reference trajectories are used in reward formulation of RL which learns a policy of musculoskeletal control

These reference motions can be either human and animal motion data or synthetically generated motions such as a solution of an optimal control formulation. We show that integration of state-of-the art RL methods with OCT that are linked through reward functions is capable of addressing this inverse imitation learning problems. Lately, the imitation RL has been extensively used in animation studies where user defined keyframe motions are used to generate the synthetic data one by one (Geijtenbeek et al., 2013; Coros et al., 2011; Peng and van de Panne, 2017). Instead, our approach not only allows us to generate necessary reference trajectories with optimal control but also solve redundancy of the musculoskeletal systems. The overall schema of the learning framework is given in Figure 2.2. The first part of the framework is about simplifying the control problem with OCT. The equation of movement dynamics are written and the corresponding model is simulated (Figure 2.2A) and with numerical optimal control methods the optimum joint trajectories and optimum torque sequence are found. A variety of different movement trajectories are used as reference trajectories. To train a neural network, we rewrite the reward function as a minimization between these reference trajectories and current trajectories in musculoskeletal system (Figure 2.2B). The outcome of the RL is to identify the corresponding alpha motor neuron activities that are responsible for simulating the extensor/flexor muscles. Once we obtained the alpha motor neuron activities, this is in turn used as a supervised signal to adjust the computational motor circuit synaptic distributions. The output of the motor circuit is then used as a validation and adjustment of the simplified model in order to improve the quality of the outcome of the learning framework.

Another objective behind the development of this learning framework is to be able to integrate

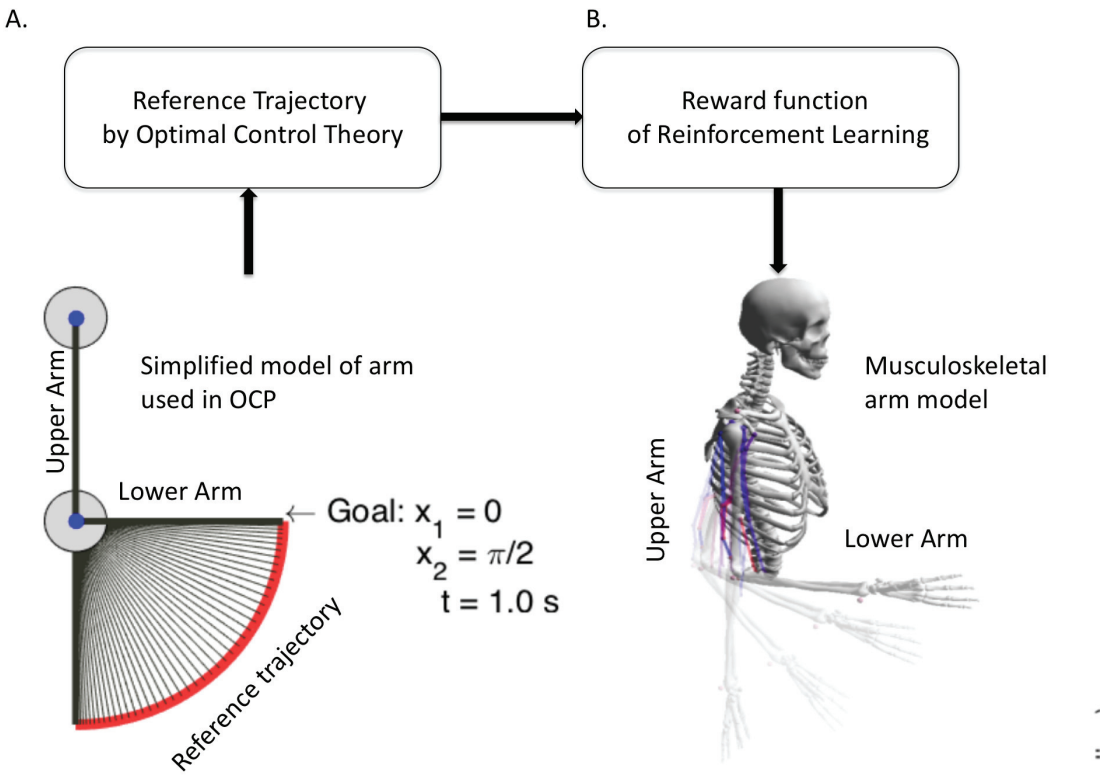


Figure 2.2: The schema of the proposed learning framework. It composes of 2 different procedures. A. Simplification of the control problem and obtaining the desired optimum trajectories with OCT. B. Training a neural network in order to simulate a musculoskeletal system and find out learning the activity of alpha motor neurons with reinforcement learning.

possible human and animal kinematic and dynamic data. The classical yet promising meta-heuristic optimization methods (e.g. particle swarm optimization, simulated annealing, genetic algorithms) and dynamic programming (e.g. RL) provide promising solutions yet they provide one of the possible multiple solutions. The solution that is found by these methods doesn't necessarily yield the precise trajectories that we are interested in. Instead combination of OCT and RL manage to generate the trajectory that is optimum with respect to cost function. It is optimum due to the fact the formulation of the OCT takes into account the system dynamics and all corresponding physical constraints while obtaining this solution. Therefore the proposed learning framework is not only capable of integrating the human/animal motion capture data but also has the ability to generate a set of synthetic data to be integrated to the computational motor circuit model. We tested our approach through a variety of different

control problems, ranging from a human arm control with different complexities to the human locomotion and standalone leg controls.

Before explaining the details of the algorithm that we used in our learning and optimization framework, we provide essentials of RL and OCT. We first explain OCT and the methodology behind how to write a nonlinear programming to numerically solve OCP then explain the reward function reformulation. After that we discuss an actor-critic formulation of RL.

2.1 Optimal Control

The fields of RL and OCT are usually considered separate optimization frameworks yet they relate to each other since both of these methods have been developed to solve the Bellman Equation (Bellman, 1957) which defines the necessary condition of optimality;

$$Q(S,A) = \sum_{S'} P^{A}_{S\to S'} \left[R^{A}_{S\to S'} + \gamma \sum_{A'} \pi\left(S^{'}, A^{'}\right) Q\left(S^{'}, A^{'}\right) \right] \tag{2.1}$$

where $Q(S,A)$ is the q value for state S and action A, $P^{A}_{S\to S'}$ is the probability of taking action at state S to next state $S^{'}$, $R^{A}_{S\to S'}$ is the reward and $\pi(S^{'}, A^{'})$ is the current policy. While RL methods have been developed to solve the Bellman's principle of optimality without explicit knowledge of the dynamic models of the environment and system to be controlled, Optimal Control methods consider these optimization problems as the solution of the differential or difference equations to be found with underlying dynamic models. Both methods are concerned with an optimization problem of minimization/maximization of expected cumulative cost/reward with sequential decisions. A solution of optimal control corresponds to a desired trajectory of the dynamic system for the sequential decision problems and each problem needs to be solved separately whereas an RL agent finds a strategy to obtain the control signals of given problem and policy function makes it possible to generalize the solution of the sequential decision problems. One difficulty arises in the context of the Optimal Control formulation is the prerequisite of precise determination of the underlying dynamic system instead it is implicitly estimated by the RL agent through exploration of the system. However, the solutions of the optimal control methods are more precise and accurate then RL methods due to the knowledge of the system dynamics. The solution of Bellman equation is also known as

Dynamic programming which is usually associated to the discrete-time optimization problems. However, for the continuous-time optimization problems, the analogous formulation of Bellman equation, which is called Hamilton-Jacobi-Bellman Equation (HJB), is utilized. The proof of continuous-time RL implementation for the actor-critic network which satisfies the HJB equation can be found in (Doya, 2000).

On the other hand, the OCT has started to be developed before the studies of Richard Bellman. Starting from 15^{th} century with the main interest on brachistochrone curve, the OCT has started to provide the essentials and the principles of the continuous time dynamical system controls. It has been evolved as an extension of the calculus of variations (Teo et al., 1991; Bryson, 2018; Kirk, 2012; Liberzon, 2011; Kamien and Schwartz, 2012). Theoretical foundations of Optimal Control have been developed by several generations of mathematicians including Johann Bernoulli, Isaac Newton, Leonhard Euler and finally last century with the breakthrough of Lev Pontryagin and Richard Bellman. Most of the state-of-the-art solutions in the control engineering (aerospace, robotics, financial instruments e.g.) attempt to solve the Pontryagin's Maximum Principle (PMP) which is the necessary condition of optimality (Pontryagin, 2018). The details of the derivations about the maximum principles for optimal control problems can be found in the textbook of Pontryagin (Pontryagin, 2018). The proof of the PMP is given in (Macki and Strauss, 2012). In (Hartl et al., 1995), a survey about the problems of having mixed and pure state constraints applied to maximum principle can be found.

2.1.1 Problem statement with Optimal Control

The basic problem formulation of the optimal control is to find a continuously differentiable funtion $u(t), t_1 < t < t_2$ to;

$$\begin{aligned} &\min_{u(.)} \int_{t_1}^{t_2} l\left(t, x\left(t\right), u\left(t\right)\right) dt \\ &\text{subject to } \dot{x} = f\left(t, x\left(t\right), u\left(t\right)\right) \\ &\qquad\qquad x(t_0) = x_0 \end{aligned} \tag{2.2}$$

minimize a cost functional $l(t, x(t), u(t))$. In optimal control problems (OCP), the variables can be given as state and control. The control variables correspond to input of the state variables.

One of the powerful part of the optimal control is its ability to handle the equality/inequality, boundary and path constraints within a solid mathematical framework. For instance, it is possible to introduce the dynamics of the system as an equality constraint into the optimal control formalization. Afterwards, the objective of the Optimal control is to find the appropriate time-dependent control signals in order to obtain the desired state trajectories that satisfy all the introduced physical constraints while minimizing the performance criterion.

The solution of the optimal control problem requires to convert the OCP formulation into a NLP while obtaining a solution which has to satisfy the PMP (Bazaraa et al., 2013; Bertsekas, 1995; Luenberger et al., 1984).

2.1.2 Pontryagin's Maximum Principle

The formulation of the optimal control problem with a terminal cost is given as;

$$\begin{aligned} \min_{(u(.),t_f)} \int_{t_0}^{t_f} l\left(t, x\left(t\right), u\left(t\right)\right) dt + \phi\left(t_f, x\left(t_f\right)\right) \\ \text{subject to } \dot{x} = f\left(t, x\left(t\right), u\left(t\right)\right) \\ x(t_0) = x_0 \\ x(t_f) = x_f \\ P_k(u, t_f) = \psi\left(t_f, x\left(t_f\right)\right) = 0 \end{aligned} \tag{2.3}$$

In PMP formulation, while inital time is fixed, terminal time is not given. The cost functional of the PMP then can be written as $\phi\left(t_f, x\left(t_f\right)\right)$ subject to the system dynamics $\dot{x} = f\left(t, x\left(t\right), u\left(t\right)\right)$. In this formulation, only equality constraints are considered such that system dynamics, initial and terminal position as well as constraints on input.

The PMP is the necessary conditions for a dynamical system to be controlled from an initial state to a terminal state under physical constraints. In order to derive PMP, one has to write down the Hamiltionian function associated with the OCP. Then Hamiltonian function is writen as the sum of system dynamics and cost with Lagrangian multiplier;

$$\begin{aligned} H(t,x,u,\lambda) &= l(t,x,u) + \lambda^T f(t,x,u) \\ &= l(t,x,u) + \left\langle \lambda, f(t,x,u) \right\rangle \end{aligned} \tag{2.4}$$

for a continuous control variable $u \in C[t_0, t_f]^{n_u}$, where the Lagrangian and system dynamics have to be continuous functions, l, f, and satisfy that they both have continuous partial derivatives with respect to system and control variables x and u for all $(t,x,u) \in [t_o, t_f]\ x\ \Re^{n_x}\ x\ \Re^{n_u}$. If a control variable $u^* \in C[t_0, t_f]^{n_u}$ provides the cost functional to have a local minima of the Optimal Control Problem and then $x^* \in C^1[t_0, t_f]^{n_x}$ is the related state trajectory, suct that a Lagrangian vector $\lambda^* \in C^1[t_0, t_f]^{n_\lambda}$ with u^*, x^*, λ^* exists. PMP states that optimal state trajectory x^*, control variable u^* and Lagrangian vector λ^* is the solution of the maximum of the Hamiltonian function given in Eq. 2.17. The Hamiltonian function states that all other admissible control sequences, $u(t)$ have to be bigger than the optimum control sequence, u^*,

$$H\left(x^*(t), u^*(t), \lambda^*(t), t\right) \leq H\left(x^*(t), u(t), \lambda^*(t), t\right) \tag{2.5}$$

According to the PMP statement, partial derivatives of the system dynamics and control input is required to exists x and u for all $(t,x,u) \in [t_o, t_f]\ x\ \Re^{n_x}\ x\ \Re^{n_u}$, which are also called costate equations;

$$\begin{aligned} \dot{x} &= H_\lambda(t, x(t), u(t), \lambda(t)) \\ x(t_0) &= x_0 \\ \dot{\lambda} &= -H_x(t, x(t), u(t), \lambda(t)) \\ \lambda(t_f) &= \Phi\left(t_f, x\left(t_f\right)\right) \\ 0 &= H_u(t, x(t), u(t), \lambda(t)) \\ t_0 &\leq t \leq t_f \end{aligned} \tag{2.6}$$

It also must satisfy that transversal condition determines the optimal terminal time t_f such that

$$\psi\left(t_f, x\left(t_f\right)\right) + H\left(t_f\right) = 0 \tag{2.7}$$

and finally if there exists an optimum terminal time;

$$\lambda^T(t_f) = \psi_x\left(t_f, x\left(t_f\right)\right) \tag{2.8}$$

where Eq 2.20 and 2.21 provide the boundary conditions.

2.1.3 Numerical Optimal Control and Nonlinear Programming

Unless the problem of interest can be solved analytically while considering PMPs (for instance a linear quadratic regulator where the system dynamics is a set of linear differential equations with quadratic cost), one has to deploy a numerical solution for the optimal control problems. Among many solutions, there are two main aproaches, direct and indirect methods. The brief explanation about the difference between these methods is that direct methods find the sequence of points in collocation points in the feasible set in order to converge the minimum of the objective function. To do that, problem is required to be written as a system equation then it can be solved numerically. Whereas indirect methods aims for providing a solution that satisfies the necessary conditions of optimality. Indirect methods would be more reliable since the minimum is found by solving the necessary conditions however they are computationally expensive methods. Extensive survey of the numerical methods for optimal control problems can be found in (Diehl and Gros, 2016). The direct methods can be described with two main steps; first system dynamics and objective functional have to be discretized, only after that the nonlinear programming techniques can be applied. Thus, problem becomes a finite-dimensional optimization problem. With the discretization of the control problem, we obtain an optimization problem in the form of nonlinear programming. In this thesis, we used direct sequential method where the continuous control problem first discretized, e.g. control variables $u(.)$, state trajectories $x(.)$ and cost function $l(.)$ are parameterized and the nonlinear optimization is done in this parameter space.

The formulation of the Nonlinear Programming can be given as;

$$\begin{aligned} \min_{x} \; & f(x) \\ \text{subject to } & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{2.9}$$

where inequality and equality constraints are given by $g(.)$ and $h(.)$. In order to obtain an extreme point in NLP formulation either a maxima or minima depending on the sign of the objective function, the first order optimality conditions, known as Karush-Kuhn-Tucker (KKT) conditions (KKT) must be satisfied. The extreme point then is called a regular point or KKT point which represents the optimum solution of the given problem.

Given a continuous cost function $f : \mathbb{R}^n \to \mathbb{R}$, inequality constraint $g_i : \mathbb{R}^n \to \mathbb{R}$ and equality constraint $h_i : \mathbb{R}^n \to \mathbb{R}$ at a point x^* such that the optimum solution of the cost function is at x^* then there exist KKT coefficients $\mu_i (i = 1, ..., m)$ and $\lambda_j (j = 1, ..., l)$, such that

$$\begin{aligned} \nabla f(x^*) = \sum_{i=1}^{m} \mu_i \nabla g_i(x^*) \sum_{j=1}^{l} \lambda_i \nabla h_j(x^*) \\ g_i(x^*) \leq 0 \;\; for \;\; i = 1, .., m \\ h_i(x^*) = 0 \;\; for \;\; j = 1, .., l \\ \mu_i \geq 0 \;\; for \;\; i = 1, .., m \\ \mu_i g_i(x^*) = 0 \;\; for \;\; i = 1, .., m \end{aligned} \tag{2.10}$$

Then the goal of the direct/indirect methods is to first turn the OCP formulation into NLP formulation,which requires discretization of the continuous time problem, and deploy a numerical solver to obtain the parameters of a given NLP problem. Among several methods, in this thesis we used direct collocation method to solve the OCP problems in the context of NLP formulation.

Direct Collocation Method

The direct collocation method first requires to convert the continuous time formulation into a discrete time formulation which then allows us to turn the OCP formulation into an NLP to

be solved with nonlinear solvers. Discretization of the OCP problem also convert the infinite amount of decision variables into a finite set that is suitable for nonlinear solvers. The discrete points of the continuous trajectories are called the collocation points which refer to specific points in time;

$$\begin{aligned} t &\rightarrow t_0, ..., t_k, ..., t_N \\ x &\rightarrow x_0, ..., x_k, ..., x_N \\ u &\rightarrow u_0, ..., u_k, ..., u_N \end{aligned} \tag{2.11}$$

The conversion of the continuous time systems into discrete set of constraints is called Trapezoid collocation or Trapezoid rule where the integral of the system dynamics is written as a set of state difference between two collocation points. Then the approximation of an integral is used to identify the set of constraints to be solved by a nonlinear solver, a compact form of the derivations for the system dynamics collocation are shown below;

$$\begin{aligned} \dot{x} &= f(x(t), u(t)) \\ \int_{t_k}^{t_{k+1}} \dot{x}\, dt &= \int_{t_k}^{t_{k+1}} f(x(t), u(t))\, dt \\ x_{k+1} - x_k &= \frac{1}{2} h_k \left(f_{k+1}(x(t), u(t)) + f_k(x(t), u(t)) \right) \end{aligned} \tag{2.12}$$

where $h_k = t_{k+1} - t_k$. Given the variables $w = x_0, u_0, x_1, u_1, ..., x_N, u_N$ and the Lagrangian multipliers $\lambda = \lambda_0, \lambda_1, \lambda_2, ..., \lambda_{N-1}, \lambda_N$, with the Trapezoid collocation, the OCP in the NLP form is the following;

$$\begin{aligned} \min_w \; & F(w) \\ \text{subject to } & G(w) = 0 \\ \text{with } & L(w, \lambda) = F(w) - \lambda^T G(w) \end{aligned} \tag{2.13}$$

where

$$G(w) = \begin{bmatrix} x_1 - x_0 - \frac{1}{2}h_k(f_1 + f_0) \\ x_2 - x_1 - \frac{1}{2}h_k(f_2 + f_1) \\ . \\ . \\ . \\ x_N - x_{N-1} - \frac{1}{2}h_k(f_N + f_{N-1}) \\ r(x_0, x_N) \end{bmatrix} \tag{2.14}$$

When we embed the objective function and constraints into the Lagrangian function;

$$L(w,\lambda) = F(w) - \lambda^T G(w)$$

$$= \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) - \sum_{k=0}^{N-1} \lambda_{k+1}^T \left(x_{k+1} - f(x_k, u_k)\right) - \lambda_N^T r(x_0, x_N) \tag{2.15}$$

Similarly, the collocation method needed to be applied for the cost integral therefore the NLP formulation can be given as;

$$\int_{t_0}^{t_f} l(t, x(t), u(t))\, dt + \phi\left(t_f, x\left(t_f\right)\right) \tag{2.16}$$

$$= \sum_{k=0}^{N} \frac{1}{2} h_k \left(l_{k+1}(x(k), u(k))\right) + l_k(x(k), u(k)) + \phi(x(k_N)) \tag{2.17}$$

With this formulation of Trapezoid collocation, a nonlinear solver can be used to solve the system equations to obtain the optimization parameteres, which is known as Direct Method. Alternatively, based on the NLP formulation above, one can also obtain the KKT conditions to be given to the nonlinear solvers if indirect methods are considered to be used or a nonlinear solver can be deployed to solve this NLP formulation to obtain the optimum solution of the OCP formulation. In case of the indirect methods, the KKT conditions are needed to be derived based on the NLP formulation and it is given below;

$$
\begin{aligned}
\nabla_w L(w,\lambda) &= 0 \\
G(w) &= 0 \\
\nabla_{x_0} L(w,\lambda) = \nabla_{x_0} L(x_0,u_0) + \frac{\partial f}{\partial x_0}(x_0,u_0)^T \lambda_1 - \frac{\partial r}{\partial x_0}(x_0,u_0)^T \lambda_r &= 0 \\
\nabla_{x_k} L(w,\lambda) = \nabla_{x_k} L(x_k,u_k) - \lambda_k - \frac{\partial f}{\partial x_k}(x_k,u_k)^T \lambda_{k+1} &= 0 \\
\nabla_{x_N} L(w,\lambda) = \nabla_{x_N} E(x_N) - \lambda_N - \frac{\partial r}{\partial x_N}(x_0,x_N)^T \lambda_r &= 0 \\
\nabla_{u_k} L(w,\lambda) = \nabla_{u_k} L(x_k,u_k) + \frac{\partial f}{\partial u_k}(x_k,u_k)^T \lambda_{k+1} &= 0 \\
x_{k+1} - f(x_k,u_k) = 0 \quad for \quad k = 0,1...,N-1 \\
r(x_0,x_N) &= 0
\end{aligned}
\tag{2.18}
$$

With this rewriting, we parameterize the optimal control problem and turn it into a finite NLP while considering the system dynamics and the additional constraints in the problem. We can now use a numerical integration algorithm to evaluate the values of the objective and constraint functions.

Primal-Dual Interior Point Algorithm

Following the KKT conditions, a solution of the nonlinear system equations given above is needed. The Primal-Dual Interior Point applies a variant of Newton methods to find the primal-dual solution of the problem while adapt the search direction and step lengths. The nonlinear system equations are linearized around the current point and the search direction is obtained by;

$$
J(x,\lambda,u)\begin{bmatrix}\Delta x\\ \Delta\lambda\\ \Delta u\end{bmatrix} = -F(x,\lambda,u) \tag{2.19}
$$

where the Jacobian of the F is given by J. Satisfying the condition that the current point must be in feasible set, then the update equations of the Newton method is;

$$\begin{bmatrix} 0 & G^T & I \\ G & 0 & 0 \\ U^k & 0 & \lambda^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta u^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -x^k u^k e + \sigma_k \mu_k e \end{bmatrix} \tag{2.20}$$

The details of the algorithm is given in Appendix 1. In this thesis, we used MATLAB's FMINCON's interior-point (Wright, 1997) solver as a numerical integration algorithm.

2.2 Optimal solution as a reward function

There exists several approaches to write a reward function for RL problems. It can be defined as a high-level goal such as obtaining a forward motion or reaching a specific position in space. It can also be engineered and defined as a combination of several objectives, such that a minimization of energy while obtaining a goal position. In our approach, which is similar to imitation learning problems, we defined the reward as a measure of how close the state of musculoskeletal system tracks a given joint trajectory. Therefore policy search is not defined as a high-level goal, e.g. position specified reward function, instead a desired trajectory is given as a reward function for RL formulation to be imitated. Consequently, reward function is the sequence of joint positions and velocities of the musculoskeletal system defined as a minimization of the difference between the given and actual trajectories. Defining the reward function as a minimization of motion trajectories and the actual trajectories corresponds to an instance of an inverse RL formulation. Finding out complete reward function that would yield a solution of the inverse RL problem however is beyond the objective of this thesis.

It is also known that the characteristic of human arm movement is creating a smooth trajectory with minimal trembles (Flash and Hogan, 1985; Shadmehr et al., 2005). To generate this smoothness in movement, it is claimed that CNS minimizes the derivative of acceleration, knowns as jerk cost funtion. Therefore, we also integrated this jerk term into our reward function formulation as a regularization term to minimize the trembling movement and create a smooth movements.

After this intuitive description, reward function can be defined as a sum of weighted differences between desired and actual trajectories not only for the differences between positions but also the velocities;

$$r_t = \left(\sum_{i=0}^{n} r_i \exp\left(- \left\| q_{d,i} - q_{a,i} \right\|^2 \right) + \sum_{i=0}^{n} r_i \exp\left(- \left\| \dot{q}_{d,i} - \dot{q}_{a,i} \right\|^2 \right) \right) + \alpha \sum_{i=0}^{n} (\ddot{q}_{d,a}) \tag{2.21}$$

where $q, \dot{q}, \ddot{q}$ denote the positions, velocities and accelerations of the joints respectively, r_i is the weight of each minimization objective and α is a constant. The formulation of the reward functions as a sum of weighted exponential functions provides the instantaneous reward to be "1" when it obtains its maximum level. L2-norm is used in order to always obtain a scalar value for the error between desired and actual trajectories.

As mentioned before, the desired trajectories to be learned can be obtained by motion capture or by simulation. With the possibility of directly using motion capture data, which could be collected through animal or human experiments, one can also find out those trajectories with using OCT. Then the solution of the OCT establishes the link between the joint space solutions to the muscle space solutions within the reward function of the RL formulation.

2.3 Reinforcement Learning

The main objective of RL is to map sensory information to actions through interactions between an agent and an unknown environment so as to obtain maximum cumulative rewards (Sutton et al., 1998). Unlike supervised learning where a teaching signal is required, an agent discovers all the necessary information by trial-and-error without explicit information about the desired sequential decisions. As a consequence of an action taken by agent, an information of reward or penalty and the new state are observed by the agent. These interactions between the agent and the environment build up the knowledge of the agent which then yields the quality of the actions given current states. Only those collected information are used to increase the performance of the agent whereas an unsupervised or supervised learning either statistical properties of the data or teaching signal are necessary to form the learning process. Applications of the general RL framework shown to be successful in a variety of artificial intelligence problems including playing computer games (Tesauro, 1994; Mnih et al., 2015) and continuous control problems (Deisenroth et al., 2013; Levine et al., 2016; Abbeel et al., 2007; Kolter et al., 2010).

One of the challenges that arises during this interaction with the unknown environment is the fact that agent needs to find out a balance between the exploration of the environment

and the exploitation of the collected knowledge. RL agents need to explore the environment with trials and errors due to the fact that there isn't any a priori knowledge at the beginning of the experiments at the agent's disposal. Additionally, the knowledge that agent gains during each trial does also not necessarily be the optimum solution of the problem. Therefore RL agents evaluate the information that it gains until that moment in order to either exploit that knowledge since it might be an optimal solution or explore the the environment more in order to avoid sub-optimum solutions. During the exploration, a sub-optimal sequence of decisions are taken while trying actions that have not taken before whereas the exploitation phase involves taking the sequence of actions which returned high rewards known from the experience gathered during the exploration. Finding the optimum balance between stochastic and determined action selection is called exploration-exploitation dilemma (Sutton et al., 1998). Though this dilemma is considered to be studied broadly, the optimum solution of it still remains an open question in the RL literature. To deal with this dilemma, agent needs to have a strategy that would yield higher rewards and avoid sub-optimal solutions. This strategy of determining which action to take given a particular state on each time step is called *policy function*.

Apart from a policy function, there are three entities in the framework of RL; a reward or punishment signal, a value function and a transition function of state-action pairs. The information about how good or bad the action taken is given by the reward signal which is to be maximized. Contrary to the reward signal which is received by the agent on each time step and specify the immediate consequence of that action, a value function represents the total amount of expected reward over the future for each action taken in a given state. The values assigned to each state-action pairs correspond to the possible long-term cumulative rewards that agent needs to maximize. Last but not least, a transition function of state-action pairs describes the model of the world which is unknown to agent *a prior*. It is also described as a transition probability between a state-action pair and a possible successor state. The transition function is not capable of representing all the details of the model of the world yet it reveals the necessary knowledge to be used during the learning process, mainly for planning. The successor states are implicitly indicated before an agent experience those corresponding states. Based on the difference whether methods that use models and planning, there are two classes of RL algorithms, ones which utilize them are called *model-based* methods in contrast with *model-free* methods that doesn't use the transition probability.

2.3.1 Problem Formulation

In RL, the selection of actions in any unknown environment is formulated as a Markov Decision Process (MDP) with state space S and action space A. The states and actions are denoted at each time step by s_t and a_t, respectively. In order for an agent to obtain a reward $r_t = r(s_t, a_t)$ at state s_t, it takes an action a_t considering a policy $\pi(a|s_t)$. Based on state action pairs (s_t, a_t), the successor state is given by the transition function $p(s^{'}|s_t, a_t)$. The weighted sum of the consecutive state-action pairs defines the cumulative reward given by

$$\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \tag{2.22}$$

where γ is called discount factor to assign weights to reward sequence, $0 < \gamma < 1$. The objective of RL is to maximize this cumulative reward that an agent receives (Sutton et al., 1998), to do that for each state-action pair, an estimation of the cumulative reward is needed to be calculated based on the current experience that agent gathered so far with

$$Q(s,a)^{\pi} = E_{\pi(a_t|s_t),p(((s_{t+1}|s_t,a_t))}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t)_{|s_1=s,a_1=a}\right] \tag{2.23}$$

It is also called value function which is the principle formulation of value-based learning methods (Sutton et al., 1998). Then the expected return of a sequence of state-action trajectory is given by

$$J(\pi) = E_{p(s),\pi(a|s)}[Q^{\pi}(s,a)] \tag{2.24}$$

To tackle the problem of maximizing cumulative rewards, there has been three different types of RL algorithms developed: value-based, policy gradients and actor-critic algorithms (Sutton et al., 1998).

2.3.2 Temporal Difference Learning

Value-based algorithms are based on *temporal-difference* learning which makes the agent to learn through the interactions with environment without explicitly knowing the dynamic's of it. The objective of the value-based algorithms is to estimate the value function where the future expected rewards are represented then to infer the optimal policy solely based upon the value function (Sutton et al., 1998). The update schema of the temporal difference learning is

$$Q(S_t) \leftarrow Q(S_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}) - Q(S_t)] \tag{2.25}$$

from the transition to successor state S_{t+1} while receiving the immediate reward R_{t+1}. Here Q denotes the value function to be learned, α is the learning parameter and γ is the discount factor. The estimation of the value function here depends on the difference between the immediate reward received by the agent and the current estimate plus the discounted future expectation. The temporal difference algorithm requires a discretization of the state and action values which makes it not suitable for continuous problems that have high-dimansional space. This problem is known as curse of dimensionality in machine learning (Bellman, 1957). Although there exists extended versions of the temporal difference algorithm for continuous problems, those algorithms require tuned constraints on the value functions (Gu et al., 2016; Amos et al., 2017).

2.3.3 Policy Gradient

To avoid the classical problems of value-based algorithms such as nonexistence of optimal value function, intractability of the state information and being computationally expensive for continuous time problems, policy gradient based algorithms have been suggested and shown to be effective to deal with those aforementioned problems (Deisenroth et al., 2013). While value based algorithms an estimation of a value function, policy gradient based algorithms depend only upon a parameterized policy function, π, where the idea is to maximize the expected future return through sample approximations. The expected future return to be maximized by the optimization of the policy;

$$\theta^* = \underset{\theta}{argmax}\left(E_{p(s),\pi_\theta(a|s)}\left[Q^{\pi_\theta}(s,a)\right]\right) \tag{2.26}$$

represents the goal of the policy gradient based algorithms where θ is the policy parameters to be optimized in order to obtain maximized cumulative reward. The update rule of the policy is

$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_\theta J|_{\theta=\theta_t} \tag{2.27}$$

based on the gradient descent. There has been a variety of policy gradient based algorithms developed such as REINFORCE (Williams, 1992) where the update of policy parameters θ

$$\theta \leftarrow \theta + \alpha E_{p(s),\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta Q^{\pi_\theta}(s,a)\right] \tag{2.28}$$

are updated with gradient ascent. It formulates the objective function as an expectation which can be approximated by sample trajectories, or intuitively measuring the log likelihood of the sample trajectories. One of the disadvantages of REINFORCE formulation is the fact that the convergence of parameters to optimum value can be slow given the possibility of high variance.The other example of a policy gradient algorithm is the Natural Policy Gradients (Amari (1998)) where the Fisher information matrix, F,

$$\widehat{F}_{\theta_\pi} = \frac{1}{T}\sum_{t=0}^{T} \nabla_\theta \log_{\theta_\pi}(a_t,s_t)\, \nabla_\theta \log_{\theta_\pi}(a_t,s_t)^T \tag{2.29}$$

is used to estimate the curvature of the policy function in order to avoid big changes in policy update which would result in divergence. Then the update rule of a natural policy gradients can be given as

$$\theta_{t+1} \leftarrow \theta_t + \alpha \widehat{F}_{\theta_\pi}^{-1} E_{p(s),\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta Q^{\pi_\theta}(s,a)\right] \tag{2.30}$$

Although Natural Policy Gradient converges faster than REINFORCE, it is computationally more challenging, i.e. possible difficulty of the inversion of the Fisher information matrix.

The main difference between a value based algorithm and a policy gradient based algorithm is the applicability of policy gradient based algorithms on continuous time problems due to the natural consequence of parameterized continuous policy function. Furthermore, it is proven that policy gradient based algorithms successfully converge to a local optimum policy given regularity conditions (Sutton et al., 1998).

2.3.4 Actor-Critic Networks

Beyond the value based and policy gradient based algorithms, actor-critic methods have been studied in order to merge the effective parts of the value based and the policy gradient based algorithms together. An actor-critic method deploys a critic function to estimate a parameterized value function while using another function, namely actor, to assess a policy function based on the expected return calculated by the very same critic function. Instead of using an unknown true value function, an estimation of a value function by critic function is used, $Q^w(s,a) \sim Q^\pi(s,a)$, then the update rule of critic function is similar to update rule of temporal difference learning with the error function;

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, a_{t+1}) - Q^w(s_t, a_t) \tag{2.31}$$

is used to update the parameters of the critic function

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t) \tag{2.32}$$

then the actor function similarly uses a gradient ascent update schema of weighted policy function in REINFORCE with respect to the estimation of critic function.

$$\theta_{t+1} \leftarrow \theta_t + \alpha_\theta E_{p(s_t),\pi_w(a_t|s_t)} \left[\nabla_\theta \log \pi_\theta Q^w(s_t, a_t) \right] \tag{2.33}$$

where the parameterized value estimation is a linear approximation of the stochastic policy function;

$$Q^w(s_t, a_t) = \nabla_\theta \log \pi_\theta^T w \tag{2.34}$$

Actor critic methods provide several improvements for RL formulations, for example; the high variance problem, seen in policy gradient methods, is less observable, they are computationally favorable for action selections and it is shown to be efficient in learning stochastic policies (Sutton et al., 1998; Peters and Schaal, 2008). Computational efficiency of the actor critic methods allow these methods to be utilized in RL formulations where promising results for high-dimensional problems have been successfully achieved. Based on the universal approximation theorem, Weierstrass approximation theorem, which states that given any uniform convergent series, a continuous function with a closed interval $[a, b]$ can be approximated and it is proven that neural networks are universal function approximators (Hornik et al., 1989). Therefore, utilization of neural networks in the context of actor-critic formulation leads the possibility of tackling the high-dimensional continuous action and state space problems in a more computationally efficient manner, which is nowadays known as Deep Neural Networks (Mnih et al., 2015) and integration of RL into Deep Neural Networks is now called Deep RL (Lillicrap et al., 2015). When a multi-layer neural network is used for actor and critic to learn the value and policy function in a straightforward architecture and with few iterations, an actor-critic network is becoming more capable of approximating the high-degree continuous problems. With the increased capability of computational resources, a RL algorithm also has the ability of scaling to more challenging problems. The RL algorithm deploys the same learning rules with only the exception of using multi-layer neural networks for the actor and critic parameter assignments.

2.4 Proposed Method - Trajectory Mimicking

To obtain a realistic physics-based musculoskeletal control, we followed a data-driven approach to obtain a trajecory mimicking in musculoskeletal systems. A similar approach has been gaining attention in animation studies, (Lee et al., 2014; Peng et al., 2018). In contrast to those studies where motion capture frames are provided, we generate necessary kinematic

trajectory data by using an optimal control solution. The objective of trajectory mimicking is to learn a control policy in an actor-critic neural network with RL. The learned policy controls muscles in order to generate a motion that resembles given reference trajectories. The proposed learning architecture with actor-critic network is given in Figure 2.3.

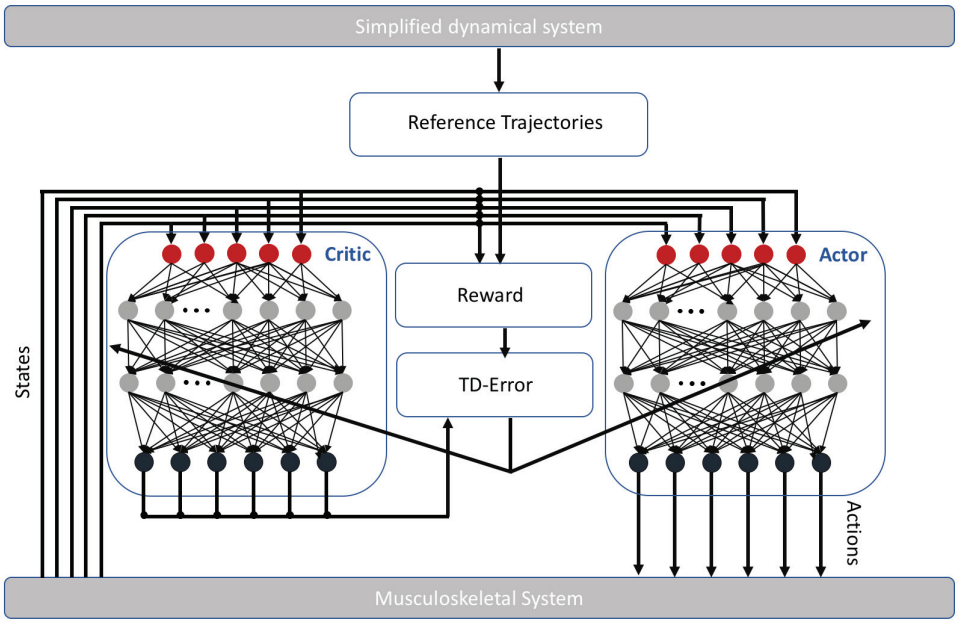


Figure 2.3: Trajectory mimicking with actor-critic network. Reward function is defined as a minimization between reference trajectories and current state of the musculoskeletal system. TD-error is calculated with values provided by Critic network and reward which in turn used in the update of both critic and actor network.

Algorithm1 illustrates each step of the optimization and learning framework. As it can be followed, the first part involves finding a numerical solution for optimal control formulation. The objective is to minimize the cost function given lagrangian multipliers that corresponds to constraints.To obtain the reference trajectories, we formulate the control problem with OCT. As mentioned earlier, OCT allows us to write a constrained optimization problem while strictly constraining the solution with the dynamical system. Although one can find an analytical solution of a control problem, it is very limited to simple control problems, therefore a numerical solution is required to solve this constrained optimization problem. We used Direct collocation method to solve OCP numerically. It requires us to convert the continuous

time Optimal control problem into a discrete time optimization problem. The necessary part is to convert the differential equations into a system equation where collocation points represents the necessary constraint of the solution. This conversion is given in Eq. 2.14.

These collocation points have to be satisfied while obtaining a candidate solutions in a finite-dimensional space which has to satisfy the PMP. Therefore, we need to write a Lagrangian function while considering this collocation points, which is given in Eq. 2.4 (line 2 in Alg. 1). We start a random initial values for a tuple (x^0, λ^0, u^0), state, lagrangian value and input respectively (line 3 in Alg. 1). Then we utilize the MATLAB's FMINCON function with primal-dual interior-point algorithm to obtain the reference trajectories with corresponding torque values (line 3-13 in Alg. 1) to solve this NLP. The primal-dual interior-point algorithm is called within FMINCON funtion to solve the nonlinear programming (Mathworks, 2018). Since the optimal control formulation requires to write the system dynamics and constraints, we managed to obtain the analytical gradients of the problem from the functions of the system dynamics and constraints, then these gradients are given to the FMINCON to accelerate the optimization procedure. The tolerance rate (TolFun) is set to high at the beginning of the numerical simulations with 5 collocation points (FMINCON,'TolFun' = 1e-3), if the optimum solution cannot be achieved with this setup then we decreased the tolerance rate and used 20 collocation points (FMINCON,'TolFun' = 1e-6). Our aim was to avoid unnecessarily long optimization. The solutions of the arm movement approximately took 2.34 seconds and 3630 iterations to compute in average in FMINCON's dual-point interior-point method. The termination condition of the algorithm is either user-defined maximum iteration (4000) or TolFun that guarentees that cost function is lower than user defined minimum value (FMINCON,'TolFun' = 1e-6). Algorithm ends when a candidate solution has less than user-defined minimum cost value (TolFun) while satisfying all the constraints. The candidate solution represents the reference trajectories to be used in reward functions of RL (line 16 in Alg. 1). After obtaining the reference trajectories, we then define the reward function which takes into account the difference between the reference and actual trajectories along with cost of jerk to obtain a smooth movement. We used this reward function to be minimized in the learning of actor-critic neural network.

After obtaining reference trajectories, we used an actor-critic network trained with RL, where the policy is represented by actor network ($\pi_\theta(a|s)$), and the value is represented by the critic network ($Q_w(s,a)$) in which the parameters are updated in tandem. The role of the

Algorithm 1 Computation of muscle actuations with OCT and RL

procedure OBTAIN THE REFERENCE TRAJECTORY
Problem Statement: Minimize $F(w)$ subject to $G(w) = 0, L(w,\lambda) = F(w) - \lambda^T G(w)$
Input: Determine the tuple (x^0, λ^0, u^0) ,strictly feasible
For $(k = 0, 1, ..., 4000)$ or (TolFun$< 1e-6$):
    Set $\alpha_k \in [0,1]$
    Solve the system equation to obtain the next tuple $(\Delta x^k, \Delta\lambda^k, \Delta u^k)$
$$\begin{bmatrix} 0 & G^T & I \\ G & 0 & 0 \\ U^k & 0 & \lambda^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta u^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -x^k u^k e + \sigma_k \mu_k e \end{bmatrix}$$
    Obtain the tuple for the next iteration
    $(x^{k+1}, \lambda^{k+1}, u^{k+1}) \leftarrow (x^k, \lambda^k, u^k) + \alpha_k(\Delta x^k, \Delta\lambda^k, \Delta u^k)$
    choose α_k so that $(x^{k+1}, u^{k+1}) > 0$
Until Convergence:
    where $\sigma_k \in [0,1]$, $\mu_k = ((x^k)^T s^k)/n$ and $e = (1, 1, ..., 1)^T$
Output: Reference trajectory $w = (x^0, u^0, x^1, u^1, ..., x^K, u^K)$ where $q_d = x[0,:]$, $\dot{q_d} = x[1,:]$

function ASSIGN THE REFERENCE TO REWARD
    $r_t = \left(\sum_{i=0}^{n} r_i \exp\left(-\left\| q_{d,i} - q_{a,i} \right\|^2\right) + \sum_{i=0}^{n} r_i \exp\left(-\left\| \dot{q}_{d,i} - \dot{q}_{a,i} \right\|^2\right)\right) + \alpha \sum_{i=0}^{n} (\ddot{q}_{d,a})$

procedure OBTAIN THE MUSCLE ACTUATIONS
Input: Initial Actor network $\pi_\theta(a|s) = \mathcal{N}(a|\phi_\theta(s))$ Initial Critic network $Q_w(s,a)$ Learning rate α_{rl}
Initialization: $\pi_0 = (s_0, g_0, a_0, r_0, s_0^{'})$
For $(t = 0, 1, ..., T_{max} = 4000000)$:

    procedure ROLLOUT OF TRAJECTORY SAMPLES
        Observe state s_t and sample action $a_t \sim \mathcal{N}(a|\phi_\theta(s_t), \Sigma)$
        Execute sample action, a_t observe the reward, r_t observe successive state s_{t+1}

    procedure CRITIC NETWORK UPDATE
        Take the actions $a_t \sim \mathcal{N}(a|\phi_\theta(s_t), \Sigma)$
        Calculate the temporal difference error δ_t
        $\delta_t = r_t + \gamma Q_w(s_{t+1}, a_{t+1}) - Q_w(s_t, a_t)$
        Update the critic network parameters w_{t+1}
        $w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$

    procedure ACTOR NETWORK UPDATE
        Update the actor network parameters θ^a_{t+1}
        $\theta^a_{t+1} \leftarrow \theta^a_t + \alpha \nabla_{\theta^a} J|_{\theta^a = \theta^a_t}$

EndFor:
    Obtain the optimum actor and critic networks
    Obtain the desired stimulus to be given to muscle dynamics

critic network $Q_w(s,a)$ is to assign values for actions taken by actor network $\mathcal{N}(a|\phi_\theta(s_t),\Sigma)$. We used 256 neurons for actor network and 512 neurons for critic network. The activation function of each neuron is sigmoid function except the output layer of critic network which is a linear function. Each network consists of 3 hidden layers, the output of critic network is only one neuron which indicates the corresponding value for a given state and action, however the output of the actor network is equal to the number of muscles in the musculoskeletal system. The training of the actor-critic network starts with initialization of the variables by a tuple ($\pi_i = (s_i, g_i, a_i, r_i, s_i^{'})$) that corresponds to state, goal, action, reward and next state with exploration noise from Ornstein-Uhlenbeck process with ($\theta = 0.1, \mu = 0.0, \sigma = 0.2, \sigma_{min} = 0.05$, annealing = $1e-6$) (line 20 in Alg. 1). Training of an actor-critic network first requires a rollout of trajectory samples and observe instantaneous reward r_t (line 24-25 in Alg. 1). Based on this reward, we first calculate the temporal difference error while considering difference between Q-values of successive states and current states (line 30 in Alg. 1). Then critic network is updated with temporal difference error (line 32 in Alg. 1) whereas the actor network is updated with Policy Gradient given in Eq. 2.26 (line 36 in Alg. 1). Training of an actor-critic network ends with a user-defined maximum number of iterations (4000000). At the end of each iteration, tuples are stored to be used to update actor and critic network (line 38 in Alg. 1). The goal of training the critic network (value function) is to predict the cumulative expected reward given a policy and consider the consecutive states to goal, whereas updating the actor network (policy) while considering the value function provides the improvement of the actions. Both parameters of actor and critic network are updated with stochastic gradient descent. At the end of the training, we obtained optimum actor and critic networks where actor network provides the stimulus to the muscle dynamics to mimic the reference trajectories.

2.5 Conclusion

One of the aims of the proposed learning and optimization framework is to obtain reference trajectories and solve the ill-defined musculoskeletal control problem. The reason why we used OCT is to create reference trajectories that are mathematically optimal and also to test different hypotheses of arm movement. In addition, OCT allows us to generate the reference trajectories that are aligned with the hypothesis of Optimal control. It also allows us to integrate biomechanical constraints into optimization formulation in a rigorous manner.

Although the problem of finding a solution for muscle control can be regarded as a regression

problem, in which a standart supervised learning or linear and nonlinear regression methods could be utilized, it is an ill-defined problem. There is a dimensionality of state mismatch between the supervised signal of joint trajectories (in joint space) and number of muscle actuators (in muscle space). Nevertheless, this problem requires finding the activation of higher dimensional muscle control from low dimensional reference trajectories. Therefore, one needs to solve this ill-posed problem either with a meta-heuristic optimization methods (e.g. particle swarm optimization, CMA-ES) (Geijtenbeek et al., 2013; Lee et al., 2014) or probabilistic inference (Antonova et al., 2016) or RL (Lee et al., 2019). It has been widely discussed that RL has caveats from computational point of view, such that it is not sample efficient and converges slowly. However, recent development has improved the scalability and computational effiency of the RL algorithms (Botvinick et al., 2019). Due this scalability of Deep rl methods with neural networks, we focused on RL in this learning and optimization framework, however the comparison of different methods has been left for future work.

3 Results

3.1 Introduction

How is multi-joint movement control obtained with redundant muscles? What is the best strategy to solve this ill-posed problem? These questions have become an interest of severel machine learning and motor control research (Jaśkowski et al., 2018; Lee et al., 2014; Geijtenbeek et al., 2013; Lee et al., 2019). Obtaining a control strategy to solve not only simple reflex movements but also multi-joint muscle control problems can reveal the basic principles of the neural motor control system once the details of this control problem are understood. This bottom-up approach can be considered as identifying the physical limitations of the motor control system as well as putting constraints to the underlying neural circuits.

One of the objectives of musculoskeletal learning research is to identify these possible constraints of the neural motor control. It can be stated that once we obtain the solutions of multi-joint redundant muscle control problems, then we can ask for the details of the underlying neural circuits that generate these solutions. Since the two-joint motion is considered as vastly complicated, for instance the dynamics of double pendulum (simple two joint system) can generate chaotic behavior, a rigorous approach is necessary to solve these high degrees-of-freedom control problem. In this chapter we demonstrate the capability of our proposed learning and optimization framework to tackle the redundant muscle control problems in musculoskeletal systems. To address the question of what is the basis of CNS to decide which specific trajectory from a variety of possibilities, we investigated one of the movement hypothesis, OCT.

3.2 2-Joints 6-Muscles Arm Control

The challenge of the musculoskeletal system arises due to the redundancy of the architecture of the muscle control, for instance a joint is controlled by several extensor and flexor muscles simultaneously. Besides the redundancy, the system itself and the muscle dynamics are both nonlinear systems. Obtaining a control signal to articulate a musculoskeletal system requires dealing with this nonlinearity and redundancy. Here,as a proof-of-concept a two-joints six-muscles human arm model has been used to test the ability of the learning algorithm that has been introduced in the previous chapter. We performed several experiments to determine the robustness and quality of the learning algorithm. The experiments include trajectory finding for a given goal position, transition from one goal position to another, creating repetitive trajectories, obtaining a goal position withing a given time and switching between desired goal position to repetitive movements.

3.2.1 Models and Methods

The dynamic model of the 2D arm is identical to the double pendulum, (it is called pendubot if it is underactuated) which is a two-link robot and it is the simplest form of a robotic arm. In case of a single control input, the torque is exerted to the first joint, but second joint is freely move without receiving any control signal. In an arm control, both joints are controlled by two different actuators. The sketch of the kinematics of the arm model is shown in Figure 3.1.

The links are representing the forearm and upper arm respectively and it is simulated in horizontal plane with 2 degree of freedom. The muscles that have been used in this model are; long head Biceps, short hand Biceps, Brachialis as flexors, long head Triceps, lateral head Triceps and medial head Triceps for extensor muscles. The difficulty of this problem is that the 4-dimensional nonlinear system is controlled by two actuators and the dynamical behavior of the two-link arm is an example of a chaotic system. The equation of motion for the arm model derived by using Euler-Lagrange methods. Θ_1 and Θ_2 are the angles of the elbow and shoulder whereas u_1 and u_2 are torques of respective joints. The mass of the forearm and upper arm are m_1 and m_2. the lengths are l_1 and, l_2 respectively. The inertia of the forearm and upper arm are I_1 and I_2 with gravity g. The dynamics of the arm system ($\ddot{\Theta}_1$, $\ddot{\Theta}_2$) with respect to control inputs (u_1, u_2) is shown below.

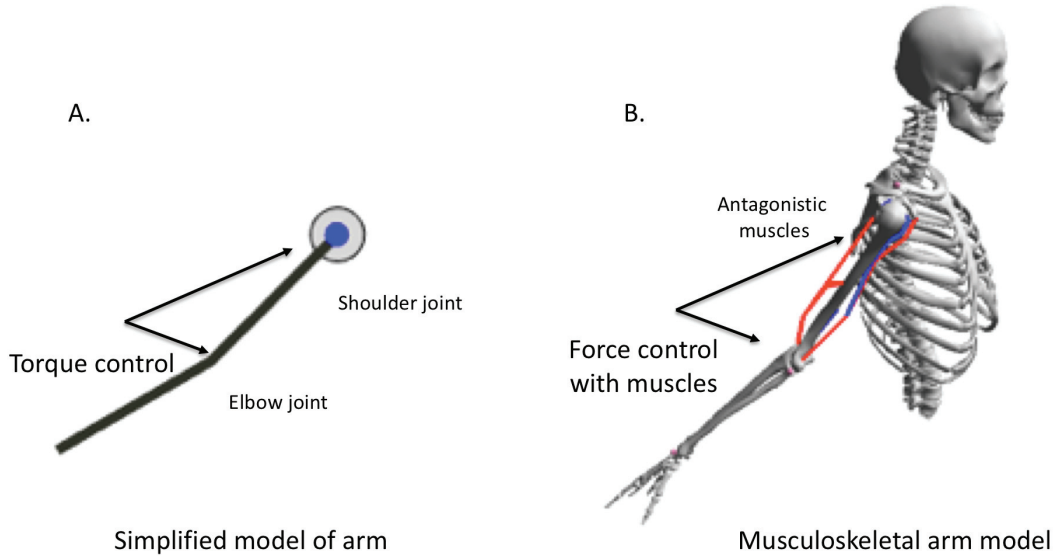


Figure 3.1: A: The simplified model of the musculoskelatal arm used in Optimal Cotrol simulations. B: A musculoskeletal human arm model. Reference trajectories are obtained with optimal control formulation with simplified arm model then these solutions ara mapped onto muscle control in human arm model.

$$\begin{aligned} u_1 &= \ddot{\theta}_1\left(l_1^2(0.25m_1+m2)+I_1\right)+\ddot{\theta}_2 0.25m_2 l_1\cos(\theta_1-\theta_2) \\ &+l_1\left(0.5m_2 l_2\dot{\theta_2}^2\sin(\theta_1-\theta_2)-g\sin(\theta_1)(0.5m_1+m_2)\right) \\ u_2 &= \ddot{\theta}_1 0.5l_1 l_2 m_2\cos(\theta_1-\theta_2)+\ddot{\theta}_2\left(0.25m_2 l_2^2+I_2\right) \\ &-0.5m_2 l_2\left(l_1\dot{\theta_1}^2\sin(\theta_1-\theta_2)-g\sin(\theta_2)\right) \end{aligned} \tag{3.1}$$

In order to solve the dynamics of the arm system, one has to transform the second-order dynamical system into a system of first-order form. This transform requires rewriting the states which should include the angles and their derivatives. The details of this transformation is also given in Appendix and the physical properties of musculoskeletal system is given in Table 3.1:

Table 3.1: Bone Physical Properties

	Humerus	**Ulna**
Mass	1.864572 kg	1.534315 kg
Mass center	(0, -0.180496, 0)	(0, -0.181479, 0)
Length	34.0cm	28.9cm
Inertia	(0.01481, 0.004551, 0.013193)	(0.019281, 0.001571, 0.020062)

Objective Function

The optimal control problem formulation of the arm system is considered as the integral of actuators efforts to obtain the minimum energy consumption. This formulation can be adapted according to the goal of the optimization problem. For instance, we studied the optimal control solution for a given time T, and we also include it as a free optimization parameter.

$$\min_{u(.)} \int_0^T 0.5u_1^2(t) + 0.5u_2^2(t)dt \tag{3.2}$$

subject to $\forall\, t \in [0, T]$;

$$\begin{aligned} \dot{x} &= f(x(t), u(t)) \\ u(t) &\in [-5, 5] \\ [x_1(0), x_2(0), x_3(0), x_4(0)] &= [0, 0, 0, 0] \\ [x_1(T), x_2(T), x_3(T), x_4(T)] &= [x_1(T), x_2(T), x_3(T), x_4(T)] \end{aligned} \tag{3.3}$$

Using an objective function which considers the minimization of the actuator effects tends to provide smooth trajectories of torque and joint states which has two different advantages;

1. Most of the numerical integration methods require to have a solution of the optimal control problems which is well-approximated by polynomial splines. Therefore, it avoids to encounter numerical integration problems.

2. It also provides stabilization of the controllers in case it is implemented in real time hardware systems.

Boundary constraints

In order to obtain a trajectory between two joint positions, initial and final desired positions can be defined as boundary constraints to force numerical optimization to obtain a solution withing these boundary constraints. These constraints restrict the solution of the optimal control within these boundaries therefore one can achieve possible optimum solutions of movement trajectories for desired initial and goal positions. In addition, besides defining the positions as boundaries, one can also integrate the desired velocities withing the boundary constraints. An example of a reference trajectory with boundary constraints are given in Figure 3.2 where initial and goal positions are;

$$\begin{aligned} x_1(t_0 = 0) = 0, x_1(t_f = 2)) = \pi/8 \\ x_2(t_0 = 0) = 0, x_2(t_f = 2)) = 0 \\ x_3(t_0 = 0) = 0, x_3(t_f = 2)) = \pi/8 \\ x_4(t_0 = 0) = 0, x_4(t_f = 2)) = 0 \end{aligned} \tag{3.4}$$

State and Control constraints

In addition to the initial and final joint positions, we can also include path constraints, for instance the ranges of the state and control variables as constraints of the optimal control formulation. In the context of arm control, the state of the joints can not go beyond certain positions due to the anatomical constraints. Furthermore, the applied torque is also another constraint of the problem due to the fact that there is a limitation of the muscle force generation in the joints. Moreover, it is also possible to encounter different constraints in different context of the optimization problems, for instance an obstacle avoidance. In this case, the positions of the obstacles can also be integrated into the optimal control formulation. Here we integrate this anatomical constraints in the optimal control formulation.

$$\begin{aligned} x_{\min} \leq x \leq x_{\max} \\ u_{\min} \leq u \leq u_{\max} \end{aligned} \tag{3.5}$$

In the context of numerical solutions of a differential equation which is a part of the numerical optimal control implementations, not only the initial conditions but also the complexity of the derivatives of the equations are needed to be taken care of. The numerical issues can rise due

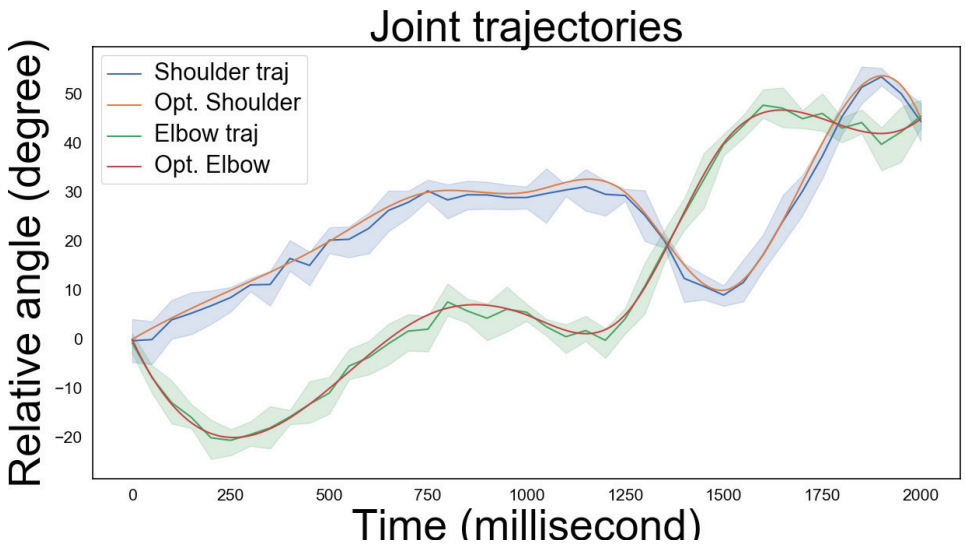


Figure 3.2: Movement trajectories of elbow and shoulder joints. The optimum trajectories are found by Optimal control which is used as desired trajectories for RL in order to find the precise muscle activities for each joint.

to fact that underlying numerical integration methods can diverge, therefore divergence of the time dependent solutions and their derivatives can be observed. One of the possibilities to be able to determine the accuracy of the numerical integration methods is to use error analysis of the implementation. Therefore, we plot the difference between the analytic solution of the arm dynamics and the solution of the optimal control to be able to understand the accuracy of the trajectories found by the numerical optimal control. The examination of the error dynamics can be useful to evaluate the limitations of the controller and also provide us to define the limitations of the RL implementations. In Figure 3.3, the error in the system dynamics and the cumulative error of the state estimations are given. Here, it shows that given this particular controller, sharp directional changes create higher collocation errors even though the range of errors is relatively small and it doesn't create divergence in the numerical solutions. In case of higher error observations, it is required to increase the complexity of the collocation points in order to avoid the numerical problems. On the other hand, more complex collocation points can generate computationally expensive solutions in this case a coarse collation points would be enough to model the solution of the differential equations. Therefore, the resolution of the collocation points depends on the complexity of the movement trajectories.

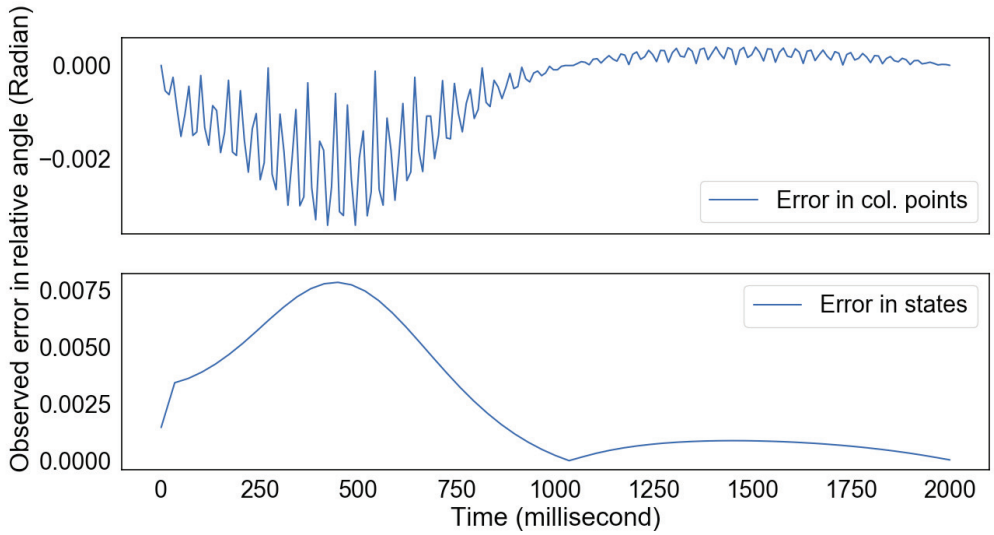


Figure 3.3: Upper: In each collocation point, we observe a numerical error which is related to solution of the differential equation. Bottom: The error we observe in collocation points is translated to the error in the solution of the state variables.

Numerical OCP formulation of Arm Control

In this section, we will explain the musculoskeletal arm control with the solutions of the numerical optimum control. The objective is to minimize the error between reference trajectories and current trajectories of the musculoskeletal arm.

Numerically solving the objective function subject to the system dynamics, boundary, state and control constraints forms the nonlinear programming formulation of the arm control. The objective of the numerical optimal control is to minimize discretized sum of the torque-squared of all the actuators while considering the discretized time system dynamics. The solution of the numerical optimal control must satisfy the collocation constraints and provide the solution that lies within the boundary of the constraints. Here, we use direct collocation with polynomial splines to achieve the numerical solutions of the problem. The summary of the algorithm (Alg. 2) is given below;

Optimizing the precise reference trajectories start with the definition of the objective function (line 2 in Alg. 2) where we used squared torque minimization to obtain energy efficient solutions. The decision variables of the optimization procedure are the state and input variables, given in (line 3 in Alg. 2). The optimization procedure is bounded with equality

Algorithm 2 Arm Control

1: **procedure** MINIMIZE
2: $J = \sum_{k=0}^{N-1} \frac{h_k}{2} \left(u_{1,k}^2 + u_{1,k+1}^2 + u_{2,k}^2 + u_{2,k+1}^2 \right)$ ▷ objective function
3: $x_1, x_2, x_3, x_4, u_1, u_2$ ▷ decision variables
4: subject to
5: $\frac{h_k}{2}(f_k + f_{k+1}) = x_{k+1} - x_k$ ▷ collocation constraints
6: $x(t_0) = x_0, x(t_f) = x_{t_f}$ ▷ boundary constraints
7: $x_{\min} \leq x \leq x_{\max}$ ▷ path constraints
8: $u_{\min} \leq u \leq u_{\max}$ ▷ path constraints

and inequality constraints that allows us to consider the physical limitations of the system, such that collocation constraints that defines the system dynamics and written in discrete time (line 5 in Alg. 2), intiial and final time constraints to define the initial and goal positions (line 6 in Alg. 2), as well as path constraints to define the boundaries of the state and input respectively (line 7-8 in Alg. 2).

3.2.2 Arm Control and Experiments

Neural control of motor system is capable of generating a variety of different smooth and accurate behaviors. Those are the behaviors that require precised control of highly redundant musculoskeletal system. Yet, human motor control system can perform all the necessary movements within the limit of the human body, for instance it can control all the variables of the system such as position, velocity, acceleration and strength of the movements. Current state-of-the-art control algorithms lack of performing similar behaviors with similar accuracy. There has been ongoing research on revealing the mechanisms of the neural control system and integration of this information to the bio-inspired robotics yet it is still an open question.

In this thesis, based on an influential idea, the optimal control hypothesis, those aforementioned broad range of movement generations with musculoskeletal system are examined with a proposed learning framework. We performed several experiments to address the capability of the framework on the generation of different type of control problems. In order to compare the ability of our learning and optimization framework, we investigated the characteristic properties of human arm movement (Morasso, 1981; Viviani and McCollum, 1983; Rosenbaum, 2009; Flash and Hogan, 1985; Sartori et al., 2013), known as invariant of movements of human arm. In Figure 3.4, we compared the experimental results and the ability of our learning and optimization framework in replicating these findings.

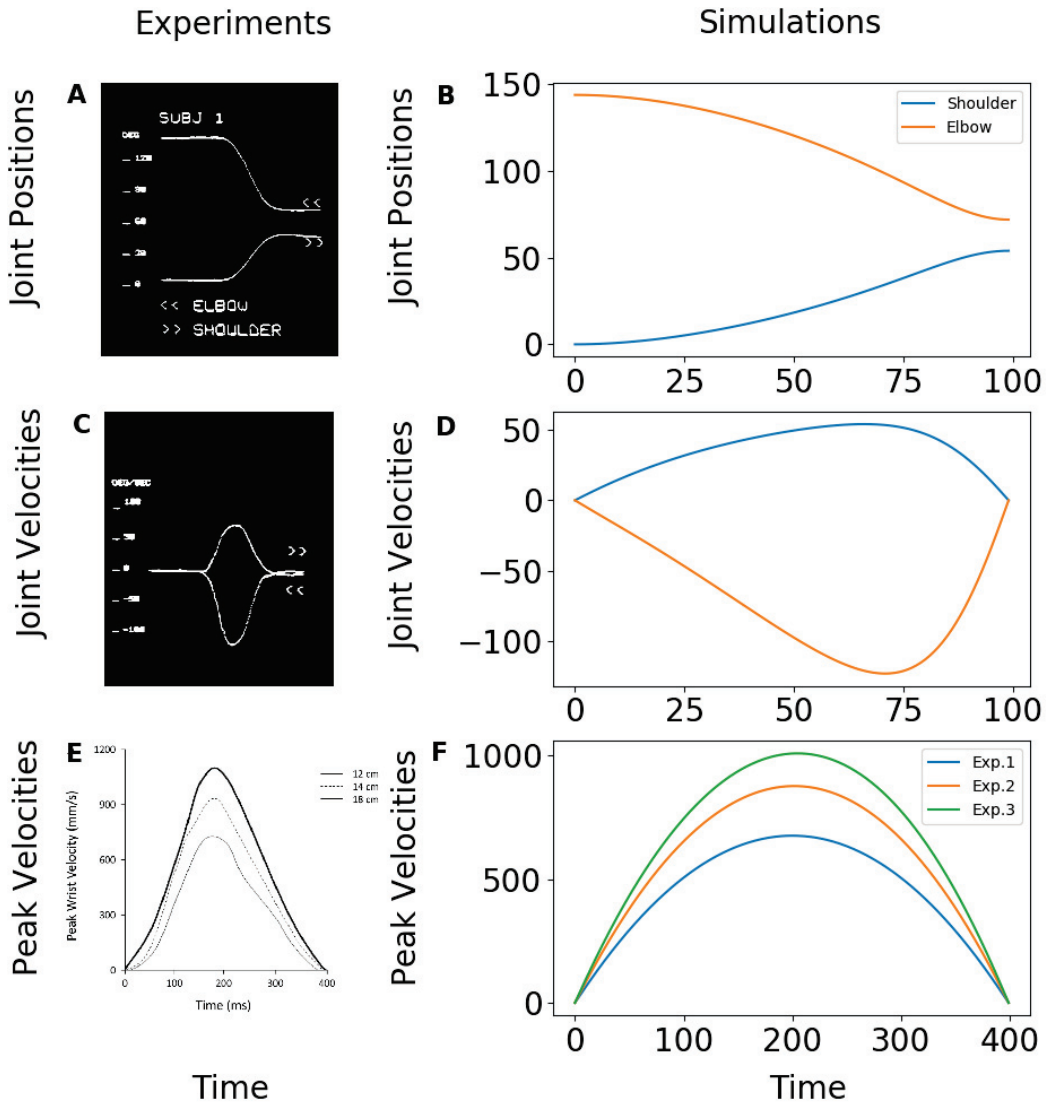


Figure 3.4: Invariants of movement of human arm. A,C. Joint trajectories of a human subject and its corresponding velocity profile B,D. A trajectory and the velocity profile that is obtained with the learning and optimization framework. Velocity profile of the simulation is shifted compared the experimental results. E. Three peak velocity profiles of a monkey experiment with an increasing target position. F. Peak velocity profile of our corresponding simulations. The peak velocities of simulations are matching the peak velocity of the monkey experiments. Figure A and C are taken fom (Morasso, 1981), and Figure E is taken from (Sartori et al., 2013)

We conducted two different experiments to evaluate the performance of the learning and optimization framework. In the first experiment (Morasso, 1981), it has been shown that

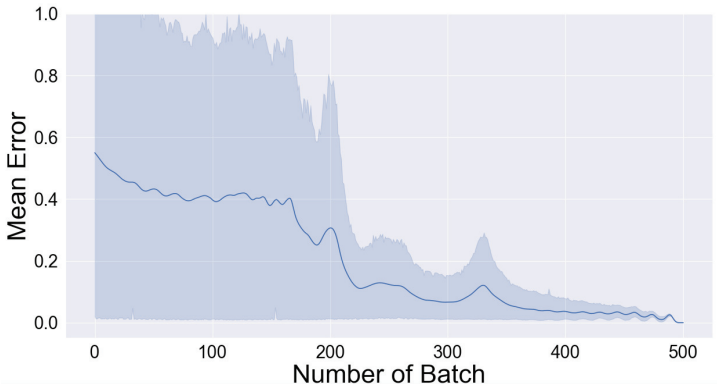
the movement of human arm has a bell-shaped velocity profile, Figure 3.4C and it has been reported this bell-shape velocity profile consistently appears in distinctive joint position profiles. To compare our results, we assigned the initial and final positions of the experimental human arm as boundary constraints with terminal time of the experimental setup. Although the joint positions of the simulation closely follow the experimental findings, the velocity profile of the simulation is a shifted bell-shaped curve. We claim that this shift has appeared due to fact that the mass distribution of the human arm subjects were not reported therefore it can be claimed the mass assignment in our simulations is not identical to the human subject. In the next experiment, (Sartori et al., 2013) Figure 3.4E, the Isochrony principle of human arm has been reported. It has been shown that human subjects have tendency to keep the execution time of the movement constant given a linear increase of the distant of the goal positions. To achieve the constant executive time, it has been reported that humans adjust the velocity of the hand movements in a linear relationship to the distance of the goal. To replicate the movement, we linearly increased the goal position in our simulations and results of the hand velocity is given in Figure 3.4F. In this simulation, we showed that this linear relationship between the distance of the goal and velocity is captured in our learning and optimization framework.

The following experiment that we designed was inspired by a classical optimal control problem which requires an end point control within a given time. The goal is defined as to reach the same target point in space albeit with different timing. In Figure 3.5-3.6, we performed three different experiments with identical goal position with a different reaching time. In all experiments, the tip of the hand of arm is required to reach $\pi/2$. In this experimental setup, the upper arm is needed to stay stable while only the forearm was allowed to move. The complexity of this problem arises due to the momentum compansation at the upper arm created by the lower arm. This constraint is integrated into the optimal control formulation as a path constraint for the upper arm and the solutions that we obtained with numerical optimal control is given in Figure 3.5c,3.6a,3.6c. In Figure 3.5a, the schematic representation of the musculoskeletal arm is given where only four snapshots of the movement are provided for the sake of clear visualization. Although the optimal solution has the ability of stabilization of the upper arm perfectly, the musculoskelatal movement has slight disturbance. In Figure 3.5b, the evolution of the error function is given. Since the definition of the reward function in our formulation is the exponential difference between the optimum trajectory and the current trajectory, reward function is interpreted as error function and the objective is to minimize

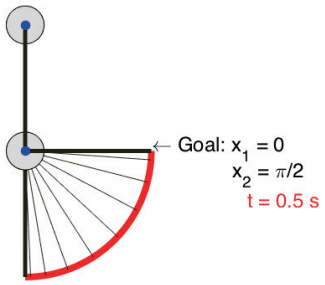
this difference. As it can be followed, although there exist a high variance and error at the beginning of the training, we obtained a low variance and error convergence at the end of the trials.

(a) Representation of the Movement

(b) Error convergence



(c) Time control task, 0.5s



(d) Musculoskeletal movement

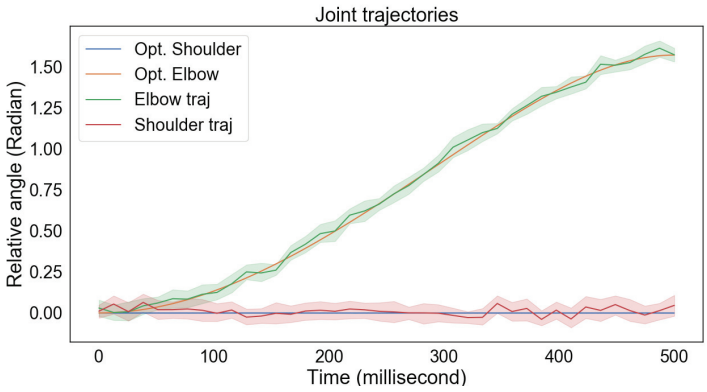
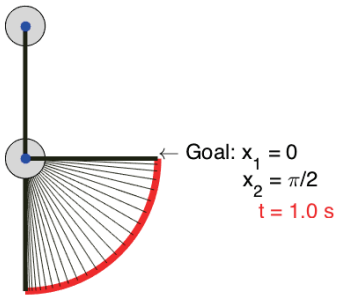
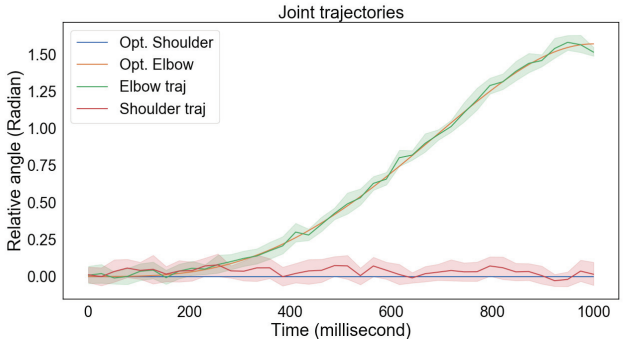


Figure 3.5: The simulation results of optimal control and comparison with musculoskeletal trajectories, time control task. A. Four snapshots of the trajectory are given above. The movement starts from resting position (00, 00) and the goal is to achieve a certain position (π/2,0) in the space within given time, 0.5 seconds. B. Error convergence at the end of training procedure C. Snapshots of the solutions of optimal control with a simplified arm model. D Trajectory comparison between simplified and musculoskeletal arm model

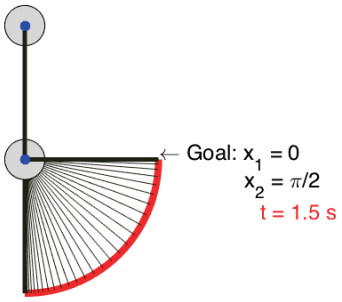
(a) Time control task, 1.0s



(b) Musculoskeletal movement



(c) Time control task, 1.5s



(d) Musculoskeletal movement

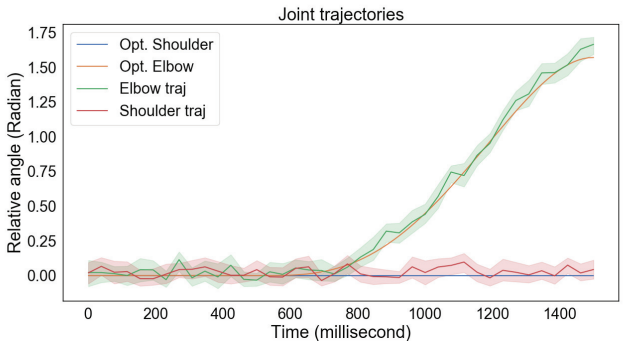


Figure 3.6: Time control task with final time at 1 and 1.5 seconds. The movement starts from resting position (00, 00) and the goal is defined as (π/2,0). A, C. Snapshots of the solutions of optimal control with a simplified arm model with 1 and 1.5 seconds respectively as final time, B,D. Corresponding trajectory comparisons between simplified and musculoskeletal arm model.

In Figure 3.5c,3.6a,3.6c, the time dependent evolution of the joint trajectories can be found where we provide all the outcome of the three different experimental setups. The main difference between each of these different joint trajectories is the torque profile found by the numerical optimal control. Since there exists a path constraint that defines the initial condition and the final condition on state values, the position and velocity profiles are obtained accordingly. As it can be followed, the solution of the numerical optimal control forces the system to start the torque generation with a time lag in order to satisfy the path constraints. While in the first experiment, the movement starts at the very beginning of the simulation, at the second and third experiment the execution of the movement starts in couple of milliseconds. In all experiments, the movement has been finalized with reaching the goal state with zero velocity, therefore we also obtain stabilizing the arm at the range of desired end point.

In Figure 3.5c,3.6a,3.6c„ the comparison between the position and the velocity profile of the musculoskeletal arm and solution of the optimal control have given. As it can be followed, both position and velocity profile of the musculoskeletal movement matches with the optimum solution, while having a strong correlation with respect to mean and variance. In all experiments, the trajectory of the musculoskeletal arm have been provided for comparison. To assess the robustness of the solutions, we performed 48 simulations of the same experiment, given in Figure 3.5. The range of errors for each of these simulations are given in Figure 3.7. We performed 20 test cases with the actor-critic network after learning has finished and collected the sum of the error during each trial. The simulations that has error below 0.1 have considered successful experiments. 36 simulations out of 48 have labelled as successfull experiments in which the RL agent learnt to perform the desired movement in the musculoskeletal arm model.

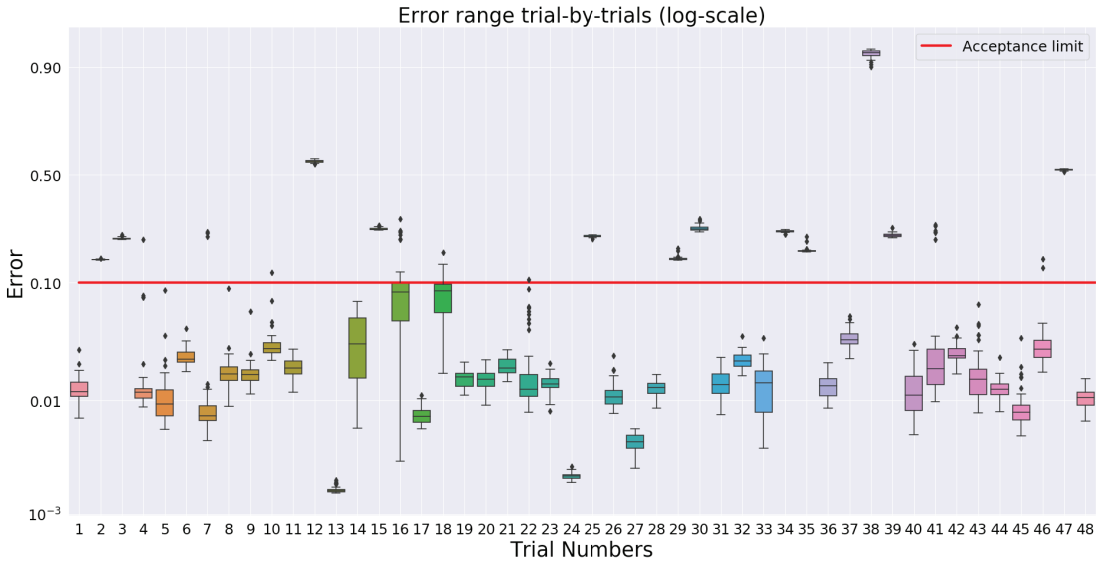
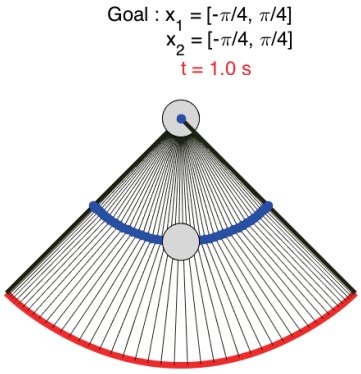


Figure 3.7: Log-scaled error range of the time control task. The same task is evaluated 48 times and there were 36 successful simulations. Mean and standard deviation is given in each bar with dots represent outliers. The acceptance limit of the error is 0.1 and given with red horizontal line.

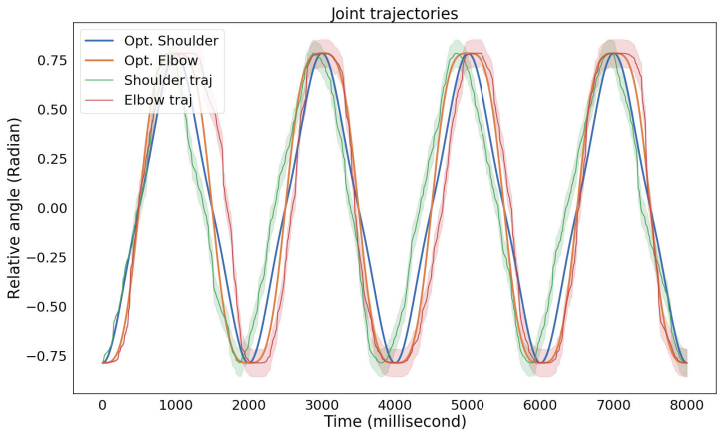
In addition to reaching task shown in Figure 3.6, we tested our solutions on two different repetitive swinging movement control experiments in order to show that the learning framework can incorporate not only goal-directed movements but also repetitive movements. Figure 3.8 shows the snapshots of the optimal control implementation (Figure 3.8a, 3.8c) and correspond-

ing musculoskeletal simulations (Figure 3.8b,3.8d). The trajectories of the arm movement found by the numerical optimal control for each of the tasks along with corresponding joint trajectories of the musculoskeletal arm show the convergence of musculoskeletal learning.

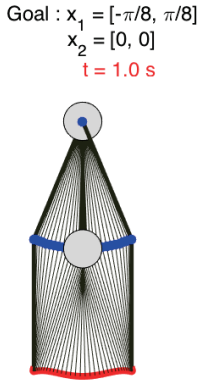
(a) Repetitive movement 1



(b) Musculoskeletal movement



(c) Repetitive movement 2



(d) Musculoskeletal movement

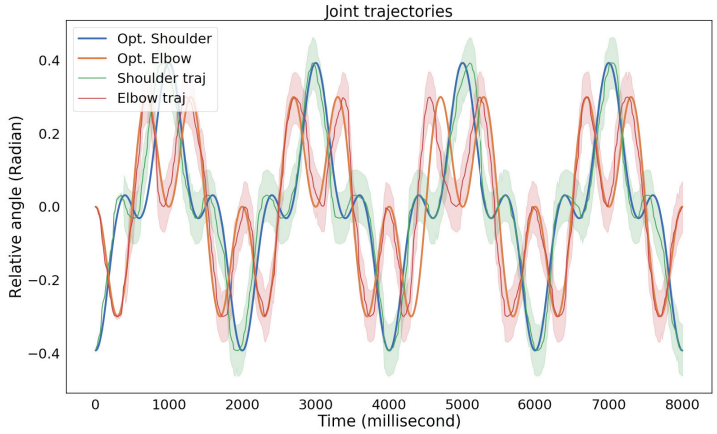


Figure 3.8: Repetitive movement task. The goal is to obtain a repetitive movement with different ranges in joint angles. A. Reference trajectories are following the same range of angle, such that both elbow and shoulder joints are rotating between $[-\pi/4, \pi/4]$. B. The comparison between reference trajectory and musculoskeletal trajectories are given. C. The reference trajectory for shoulder and elbow joints are required to be in between $[-\pi/8, \pi/8], [0, 0]$. D. Corresponding musculoskeletal trajectories with optimum trajectories. In this experiment, while shoulder joint is moving in between $[-\pi/8, \pi/8]$, elbow joint is required to stay in vertical position.

In this experimental setup, the objective is to create a periodic movement between two given positions in space within given time. In these experiments, repetition within two states in space and also the frequency of those periodic movements are the target of the control

problem. As it is discussed for the first experiment (see Figure 3.6), formulation of optimal control can incorporate timing of the movement, so the frequency of the periodic movement can also be controlled due to the boundary constraints. In addition to the initial and final time constraints, we also assigned the duration of the one cycle of the movement.

In both tasks, the periodic and repetitive movements were achieved within the range of $[-\pi/4, \pi/4]$ for shoulder position and $[-\pi/4, \pi/4]$ for the elbow position in the first experiment and also $[-\pi/8, \pi/8], [0, 0]$ were targeted for the second experiment. After the training phase, the control of musculoskeletal system is able to follow the desired trajectories as it can be seen in Figure 3.8. In these experiments, we observe higher error around the turning point of the movement directions nonetheless the musculoskeletal arm still manages to closely follow the desired trajectories that is given as targeted movements. One of the possible explanations of the high variance at the directional turn is that the human arm model composes only six muscles attached between upper arm and the upper part of the forearm. Therfore, the model lacks of muscles between forearm and wrist which increases the possibility of performing necessary strength at the joints. Considering the muscles between forearm and wrist will increase the accuracy of the performance since they take part in stabilization of the arm at the tip point.

The repeated the error analysis for the first repetitive movement task with same setup where we conducted the experiment 48 times and obtained 42 successful simulations, see Figure 3.9. The performance of the actor-critic network was better compare to time control task. It can be claimed that repetitive task is less complex than time control task since movement requires repetition of same trajectory.

Up until this point, we focused on tasks that require an arm movement to a single target in state space and discussed the performance and shortcomings of the neural controller. Though, one of the most striking abilities of optimal control formulation is that the objective of the task can be enlarged to sequential decision problems such that a consequential target positions can be tackled. Here, we present two additional experiments to study the ability of the learning framework on sequential target achieving problems, first moving from one stable goal position to another one and last but not least a transition from a dynamic and periodic movement to a stable goal position. During the transition of those movements no such waiting period was assigned although it can be incorporated to the formulation with additional constraint.

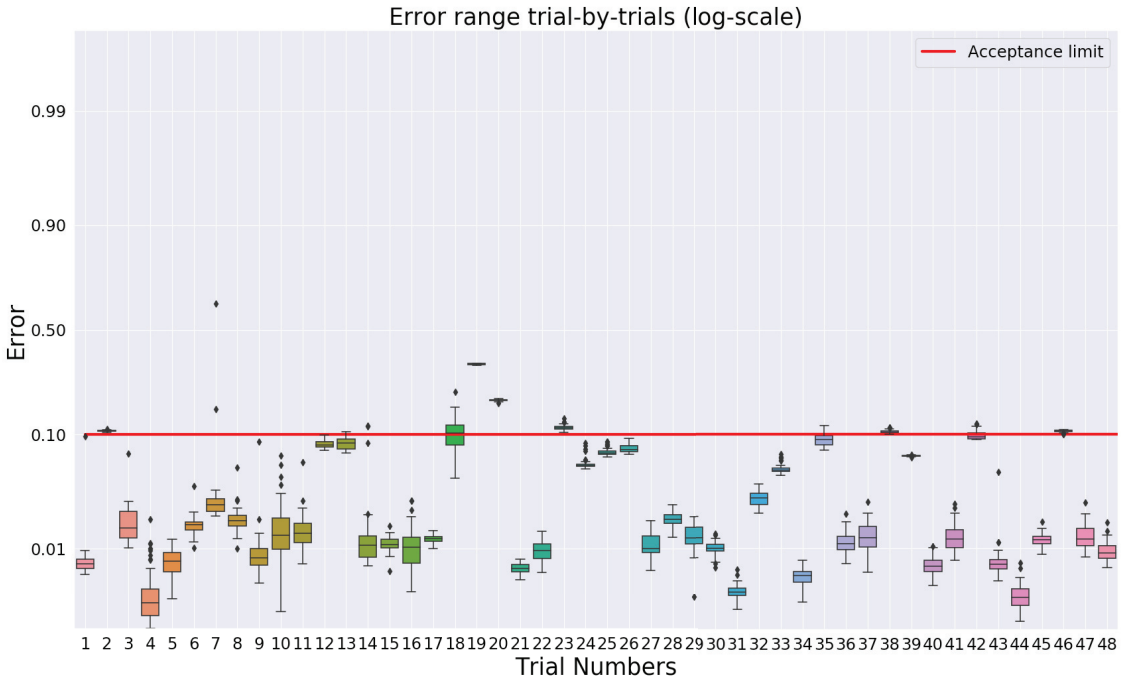


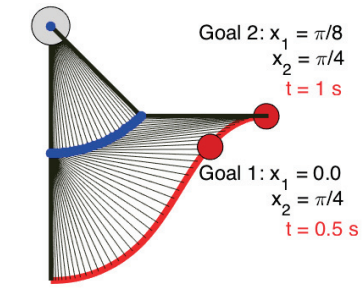
Figure 3.9: Log-scaled error range of the repetitive movement task 1. The same task is evaluated 48 times and there were 42 successful simulations. Mean and standard deviation is given in each bar with dots represent outliers. The acceptance limit of the error is 0.1 and given with red horizontal line.

In the first experiment where two static goal positions were targeted sequentially with a precise reaching time. The final target of the first task is to reach $[\pi/8, \pi/4]$ in 1 second while achieving the first goal position of $[0, \pi/4]$ in 0.5 second. The corresponding joint trajectories can be followed in Figure 3.10. We concluded this experiment with another setup while the target was assigned to be same $[\pi/8, \pi/4]$ in 1 second, but the intermediate goal position was $[0, -\pi/4]$ in 0.5 second.

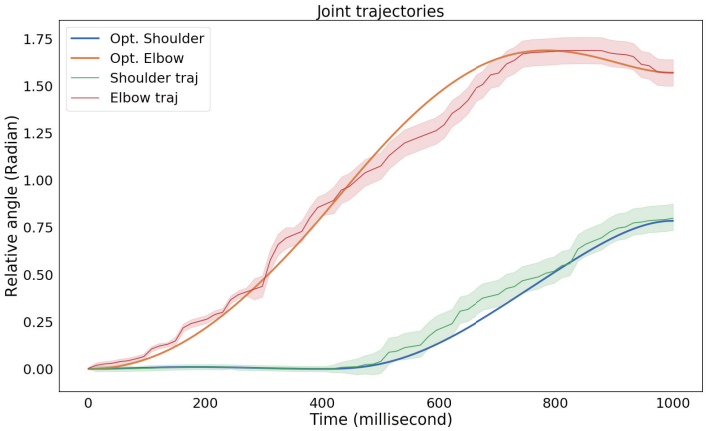
Finally, we analyze the capability of the learning framework on another sequential task where a transition from a periodic movement to static goal was performed. In this experiment, we also conduct two tasks in order to test the robustness of the approach. The first task involves following a periodic movement within the range of $[-\pi/8, -\pi/8]$, $[-\pi/8, -\pi/8]$ for the elbow and shoulder joints respectively during 0.5 seconds of a period and the task was ended when musculoskeletal arm model reaches the static goal position in state space, $[\pi/16, \pi/3]$ with an

error range of $[-0.1, 0.1]$ radian. The timing of the switch between two decisions is the time when a periodic movement ends its period then human arm model is targeted to the second goal position immediately.

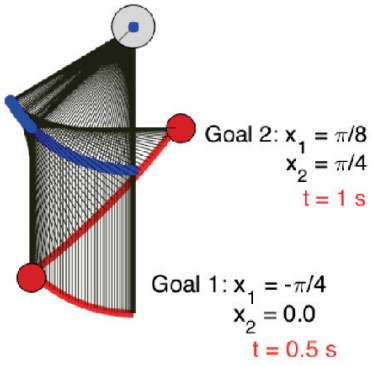
(a) Sequential reaching 1



(b) Musculoskeletal movement



(c) Sequential reaching 2



(d) Musculoskeletal movement

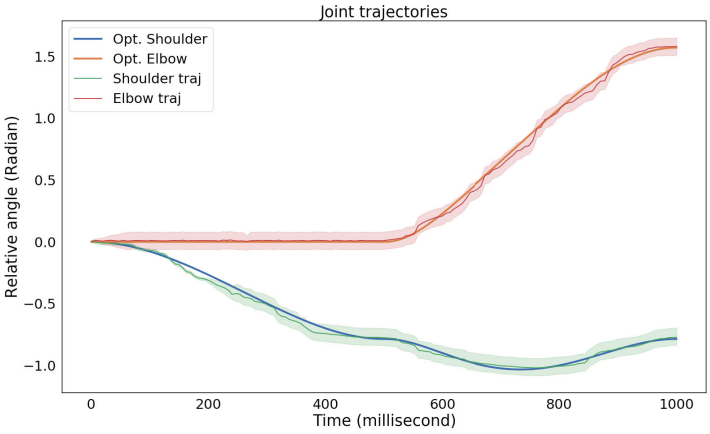


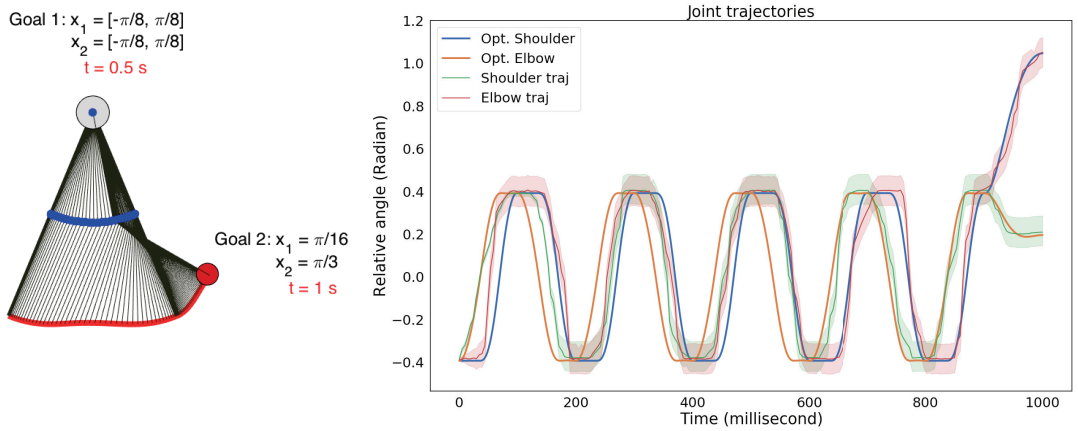
Figure 3.10: Sequential reaching task. The goal is to obtain sequential goals with given order. A. Both goal positions requires forward movement of the arm model. The first goal is $[0, \pi/4]$, and then the second goal is $[\pi/8, \pi/4]$ for shoulder and elbow joints respectively. B. Optimal trajectories as reference trajectory and musculoskeletal arm trajectories are given. C. The first goal position ($[-\pi/4, 0]$) requires to obtain a backward movement of the arm, after the arm is moving forward in order to reach the second goal position $[\pi/8, \pi/4]$. D. Corresponding optimum and musculoskeletal trajectories

The second task similarly involves a transition from a periodic movement to a static goal position albeit with different range for the periodic movement and the static goal. The range of the periodic movement is $[-\pi/4, \pi/3]$ and $[\pi/4, -\pi/3]$ for the elbow and shoulder joints

respectively, and the end goal of the human arm was $[\pi/3, \pi/3]$. The joint trajectories of each task for solution of the numerical optimal control and the musculoskeletal human arm can be seen in Figure 3.11.

(a) Reaching after repetitive 1

(b) Musculoskeletal movement



(c) Reaching after repetitive 2

(d) Musculoskeletal movement

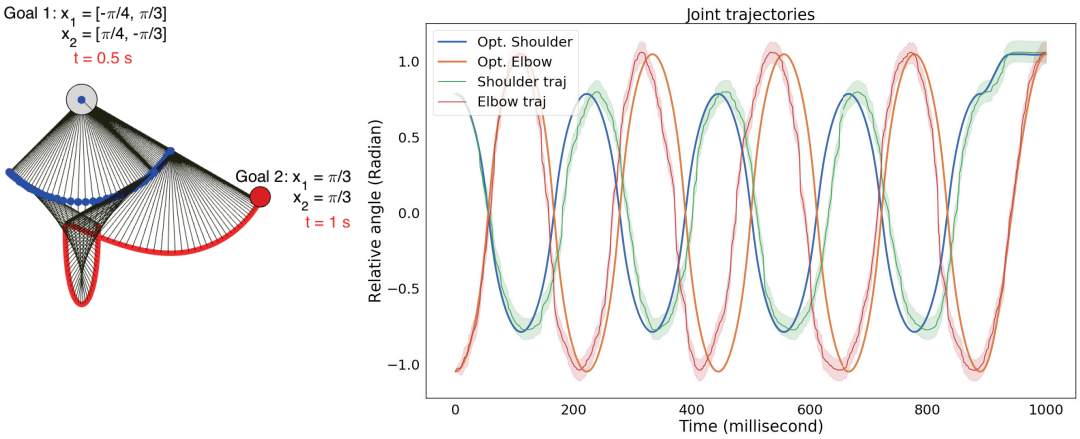


Figure 3.11: First repetitive then reaching task. A. Required repetitive task is defined as moving the shoulder and elbow joints within the same range of $[-\pi/8, \pi/8]$. After 4 full cycle, the final goal is required to be obtained which is $[\pi/16, \pi/3]$. B. Corresponding optimum and musculoskeletal trajectories C. Similar experimental design except the repetitive task is moving the shoulder and elbow joints within the range of $[-\pi/4, \pi/3]$ and $[\pi/4, -\pi/3]$, then the final goal is to reach at $[\pi/3, \pi/3]$. D. Corresponding optimum and musculoskeletal trajectories

The results of these experiments will provide the basis of the reverse engineering of the computational motor control circuit, to be discussed in the following Chapter. Based on these experiments, we obtained a several optimum trajectories to be used in the training of the spinal cord model. Not only these presented results but also several movement trajectories

that have randomly assigned goal positions were generated and used in the creation of training set of the motor control circuit building. The representation of each of these joint positions, velocities and acceleration to the spinal cord building is considered as supervised signals for adjusting the weight distributions of the motor circuit model to generate not only these optimum movements but also we obtained trajectories that a goal position is arbitrarily chosen. Error analysis of the first repetitive than reaching task is given in Figure 3.12 and the result is at the success range of time control task which was 36 successful simulations out of 48.

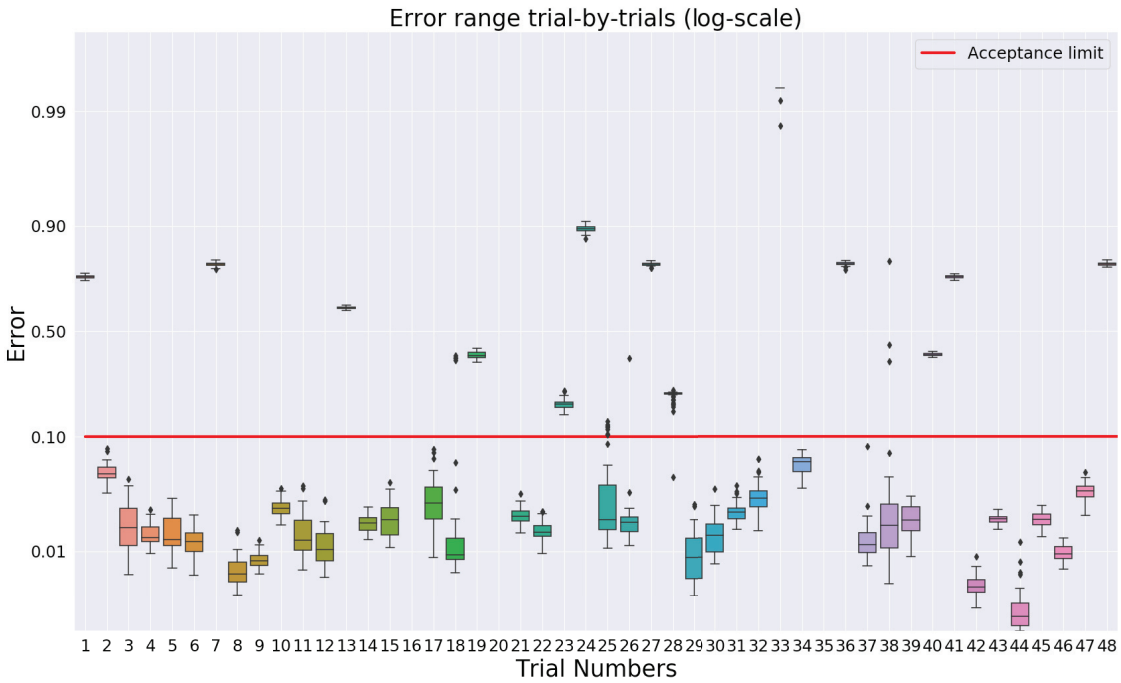


Figure 3.12: Log-scaled error range of the repetitive than reaching task 1. The same task is evaluated 48 times and there were 35 successful simulations. Mean and standard deviation is given in each bar with dots represent outliers. The acceptance limit of the error is 0.1 and given with red horizontal line.

3.2.3 Network properties

Neural network training is known to be susceptible to hyper-parameters, such as learning rate, batch size, momentum coefficient, number of neurons and number of hidden layers. These parameters have significant effect on the performance of the training. Finding the optimal

hyper-parameters improve the quality of the training and it can allow us to avoid overfitting or underfitting. Most of these hyper-parameters are tuned manually and it requires to have expertise on the problem itself and it also depends on excessive trial-and-errors. For instance, learning rate is usually selected from a small number between [0,1], the more it is closed to 0 means more stable but slow learning. However, number of neurons in each actor and critic network for RL training should be determined according to problem. Although there exists theoretical methods to determine most of these hyper-parameters, it is mainly in the case of classification problems (Haykin, 1994). Problems such as a regression needs to be studied by trial-and-error.

In order to show the significant effect of the network size on actor-critic neural network training, we run the same experiment with different actor-critic network properties. In order to obtain statistically plausible results, we run the same experiments five times and we collected the error from 20 test simulations from the same actor-critic network with identical problem setup e.g. number of hidden layers, number of neurons in each layer and also activation function for neurons. We chose one of the sequential reaching task that is given in Figure 3.10a. The results can be seen in Figure 3.13. In these experiments, the actor and critic networks are composed of one, two and three hidden layers each with ranging neuron numbers from [16,32], [32,64], [64,128], [128,256], [256,512] and [512,1024] for actor and critic network respectively. Besides the number of hidden layers and number of neurons in each hidden layer, we also investigated the role of different activation functions for neuron models. We designed the network with following activation functions; all linear, all relu, all tangent hyperbolic, all sigmoid, all tangent hyperbolic except linear activation at the output of critic, all tangent hyperbolic except relu activation at the output of critic, all sigmoid except linear activation at the output of critic and all sigmoid except relu activation at the output of critic. As it can be followed, small network size such as 16 neurons for critic and 32 neurons for actor regardless of number of hidden layers yields very high error which results a very poor control. Similar to (16,32) network, a network with (32,64) neurons for critic and actor respectively has similar results. Increasing the number of hidden layers linearly improve the quality of the simulation, although with two hidden layers error dropped to acceptance level (0.1) for tangent hyperbolic with linear and relu as well as sigmoid with linear and relu. The best results have been obtained with three hidden layers, [128,256], [256,512] neuron numbers for actor and critic respectively with again tangent hyperbolic with relu and linear and sigmoid with relu and linear activation functions. The increase of number of neurons, such as [512,1024] has caused an increase at

the error range. These experiments had been simulated at the Blue Brain supercomputing facilities, named Blue Brain Five (BB5) (HPE SGI 8600 cluster) with 72 cores in each node, and 8 nodes were dedicated to these simulations in total. In each core there is an Intel Xeon 6140 cpu, V100 SXM2 gpu with 384GB memory space and each simulation took approximately 32 hours to complete.

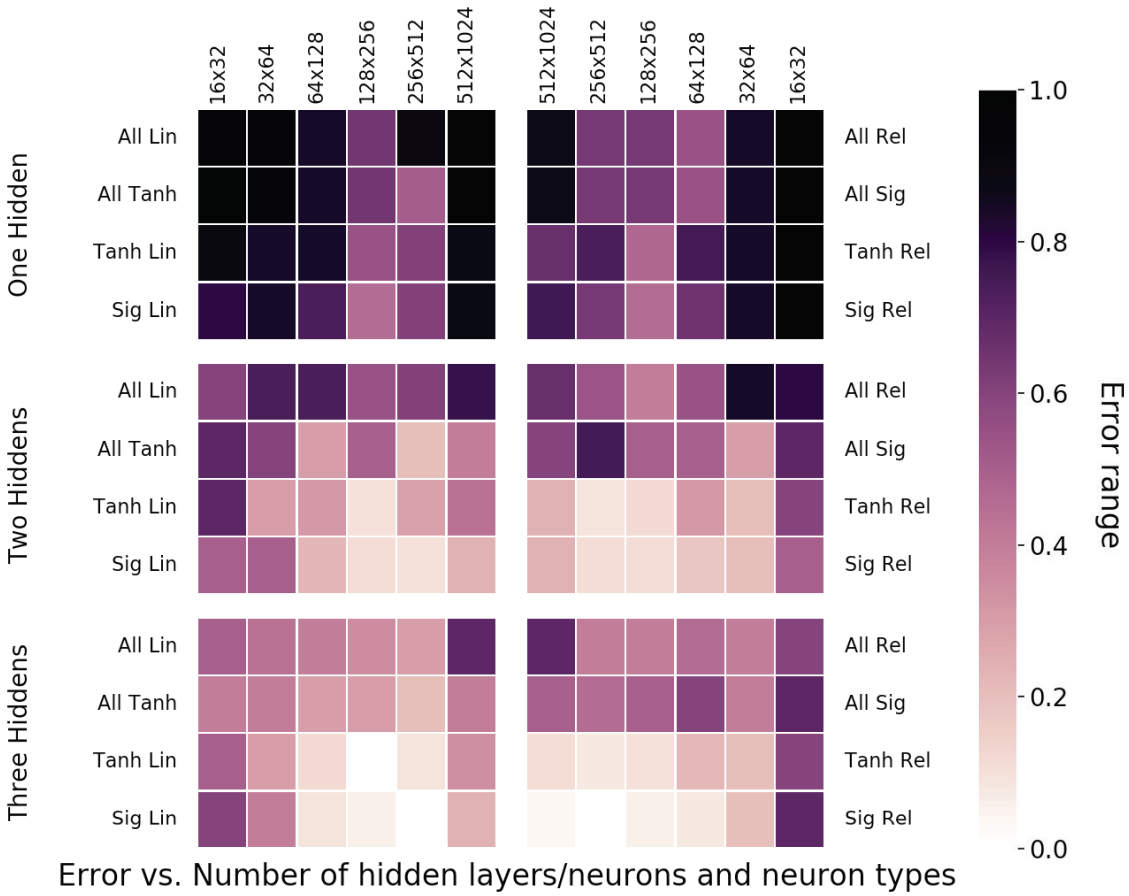


Figure 3.13: Error analysis of the actor-critic network with different network setup. There is three parameter settings, number of hidden layers (1,2 or 3), number of neurons in actor and critic networks respectively ([16,32], [32,64], [64,128], [128,256], [256,512], [512,1024]) and neuron types with different activation functions (All linear (All Lin), all linear (All Rel), all tangent hyperbolic (All Tanh), all sigmoid (All Sig), all tangent hyperbolic except linear activation at the critic output (Tanh Lin), all tangent hyperbolic except relu activation at the critic output (Tanh Rel), all sigmoid except linear activation at the critic output (Sig Lin), all sigmoid except relu activation at the critic output (Sig Rel)). The best result has been achieved with three hidden layers, [128,256], [256,512] neurons for actor and critic respectively, with Tanh Lin, Tanh Rel, Sig Lin and Sig Rel activation functions.

The evolution of the error shows the effect of number of hidden layers and different activation functions, see Figure 3.14 for only one hidden layer with [256, 512] neurons for actor and critic network with all different neuron types and Figure 3.15 for three hidden layers. It can be seen that regardless of the neuron types, error stays very high from the beginning of the simulation to the end. However, with an increase of hidden layer number to three, the effect of the different neuron types on the performance of the network becomes clear. While the error converges to below of the acceptance ratio of 0.1 with Tanh Lin, Tanh Rel, Sig Lin and Sig Rel where the best performance was achieved with Sig Lin activation function setup (mean error is 0.0012).

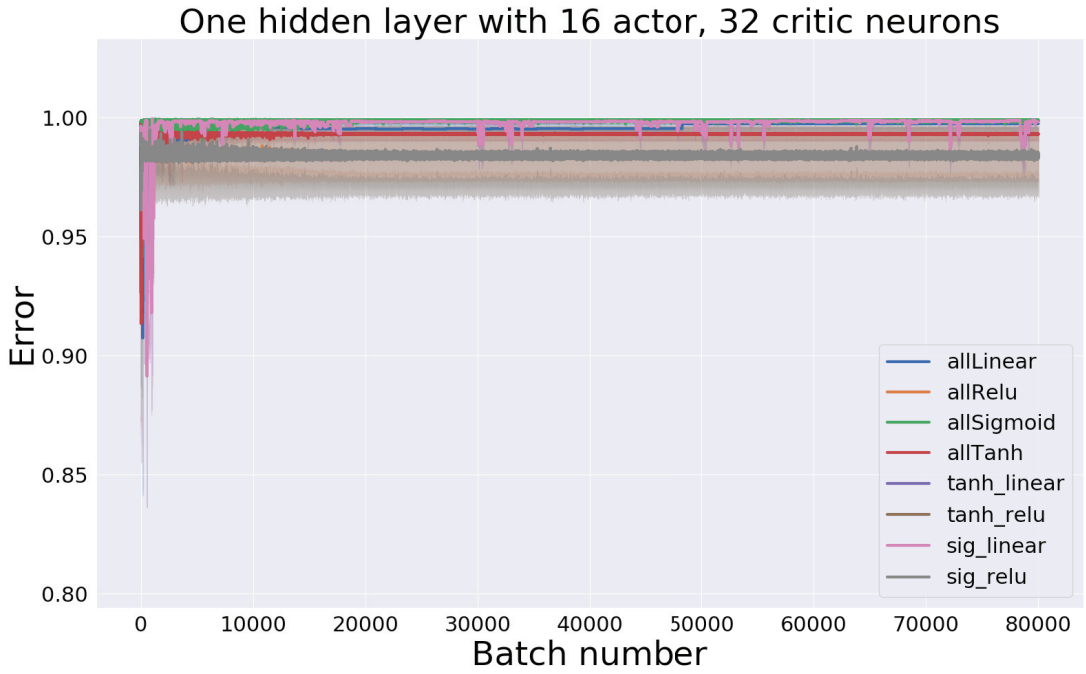


Figure 3.14: Error evolution with only one hidden layer with [256, 512] neurons for actor and critic network and different neuron types. All networks performed relatively poor.

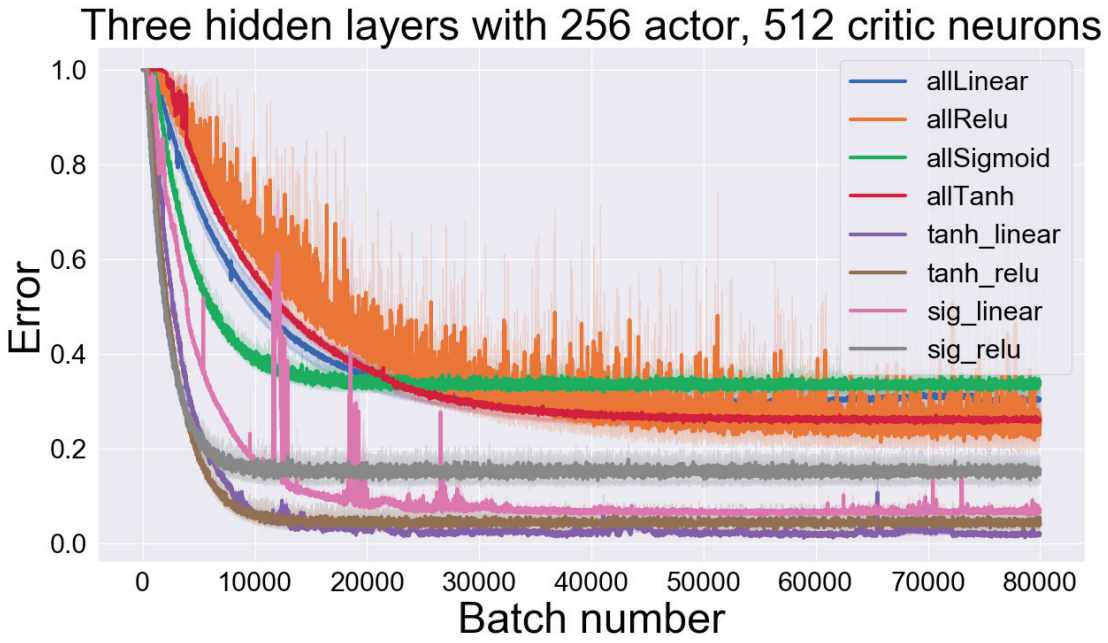


Figure 3.15: Error evolution with three hidden layers with [256, 512] neurons for actor and critic network and different neuron types. A linear or relu activation function at the outout of the critic network improved the quality of the learning. In all these networks with linear or relu output in the critic network, error converged to acceptance level of 0.1 and below.

3.2.4 Performance of trajectory mimicking

To compare the performance of our proposed learning and optimization method with a conventional RL paradigm, we run an experiment with a global reward function for the time control experiment that is given in Figure 3.5. Experimental setup is chosen identical to make the comparison reliable. As it was explained above, the problem was to obtain a goal position, $(\pi/2, 0)$, withing a given time, $0.5s$. In a classical experimental setup that result is given in Figure 3.16, the reward in RL is given to the agent when this goal position is achieved otherwise agent receives zero as a reward. The goal positions are indicated with thick lines: green for shoulder joint and red for elbow joint. As it can be followed, timing of the movement is shifted and the movement shows oscillations around the goal position. This shows that global reward function formulation requires more iterations to achieve the convergence of the solution.

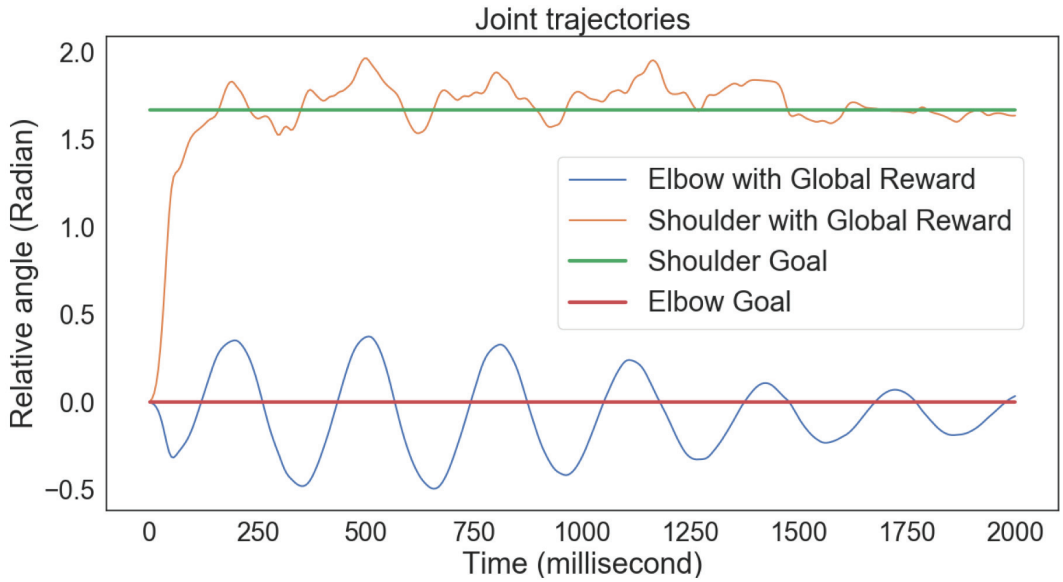


Figure 3.16: Time control experiment with a global reward function formulation. We compared a RL with a global reward values that only identify the target joint angles at the end of the simulation, which is indicated with green and red for shoulder and elbow joints respectively. Although the target has achieved the trajectories are unstable.

3.2.5 Region of learning

To assess the limitation of the proposed learning and optimization framework, we investigated the possible region of joint space that we can manage to achieve. Since there exists infinite amount of trajectories from any initial joint angles to final joint angles, we aimed at solving the problem of different initial joint angles for same final joint angles. For each initial joint angles, we run the same experimental setup of the actor-critic neural network sixteen times (3 hidden layers with 256, 512 neurons for actor and critic networks). Then we chose the first five most successful networks to obtain the test data. The success rate of the learning has been discussed in Figure 3.7, Figure 3.9, Figure 3.12. In Figure 3.17, it can be seen that for all initial joint angles, each network managed to achieve the target joint angles, [1.17, 0.87] within an error margin 0.08 mean and 0.1 standart deviation. The trajectories that we obtained show smooth movements and the arm is stabilized around the goal position for all initial joint angles.

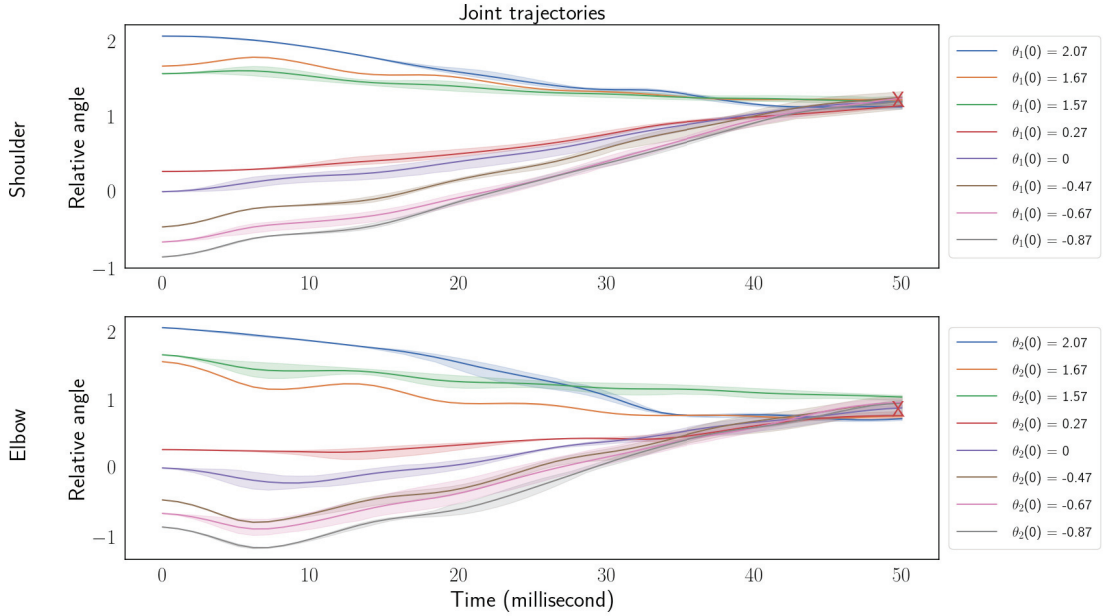


Figure 3.17: Reaching the same goal positions with different initial conditions in 0.5 seconds. There are eight different initial conditions for the shoulder and elbow, [-0.87, -0.87], [-0.67, -0.67], [-0.47, -0.47], [0.0, 0.0], [0.27, 0.27], [1.57, 1.57], [1.67, 1.67], [2.07, 2.07] respetively. The goal position is indicated with a red X, [1.17, 0.87]

3.3 7 Joints 18-Muscle Locomotion Control of Human Model

Understanding human-like locomotion and comprehending the details of the motor circuit are currently the central interest of the neuroscience community. At the same time, the revealed findings of the neuroscientific research on human-like locomotion has been one of the most common approaches in robotics research. The state-of-the art locomotion controllers are based on the findings of this ongoing interaction between biological motor control and modern control theory. The proposed inverse learning algorithm relies on this interaction between biological findings and the OCT, and here it is applied to a human musculoskeletal system to study more complex behavior generations. We demonstrate the ability of the proposed algorithm on several motor control scenarios applied on a musculoskelatal human model which has eight joints to be controlled with eighteen muscles to exert forces on those joints. These experiments include simulation of a stable joint controls, hopping behavior and locomotion control. Here we used a 7-link planar biped model of a human to find the optimal trajectories, to be used for musculoskeletal model of human, projected with RL on muscle

actuations.

3.3.1 Models and Methods

The 7-link planar biped model is a simplified model of a human locomotion, it is based on the assumption that the biped model is symmetric with respect to left and right legs therefore solutions obtained with this model are periodic. As a consequence of this symmetry, the joint trajectories for right/left leg are identical for the successive step of the left/right leg. The symmetric feature of the model allows us to formulate the objective of the optimization as a finding of a periodic solution which then projected for the upcoming steps of the biped.

This planar model of a human locomotion comprises three joints for each legs and one at the torso. These links are modeled with a mass located at the center of the link to include the inertia of each link. Locomotion under these assumptions is modeled as a hybrid dynamical system: Depending on the position of the leg, whether there is contact with the ground, the system switches between two dynamical systems, a ballistic dynamic system during the swing phase and an inverted pendulum dynamics during the stance phase. The equations of motions have been obtained based on existing studies (Hurmuzlu, 1993a,b; Kelly, 2017).

Figure 3.18 shows the simplified 7-link biped model and its corresponding musculoskeletal human model. During a step, the body is balanced during the stance phase while the motion of the body is created during the swing phase. As the movement is periodic, the stance and swing phases are switched with identical trajectories for the successive step. Therefore the optimization problem is reduced to finding a stable trajectory of a single step movement. The trajectories that have been found by the optimal control based on 7-link simplified biped model then are projected to the muscle control of the musculoskeletal system with RL to generate the identical movements claiming that the projected solutions are the optimum solution of a musculoskeletal control to be studied for neural controller behind these trajectories.

The derivations of the equations of motion of 7-link biped model are generated automatically with Matlab scripting. However, here is the summary of the dynamic equations of the 7-link biped, we used Matlab symbolic math toolbox to obtain the differentiation of the equations of motion. The model has seven degrees of freedom, the angles of ankle (q_1, q_7), knee (q_2, q_6), and hip (q_3, q_4) for each legs and a single control of the angle of torso (q_5) and these angles represent the state vector given by q and its derivatives for velocity $\dot{q}$ and acceleration $\ddot{q}$. Then given the state variables, $x = [q, \dot{q}]$ the system dynamics of a 7-linked biped robot can be

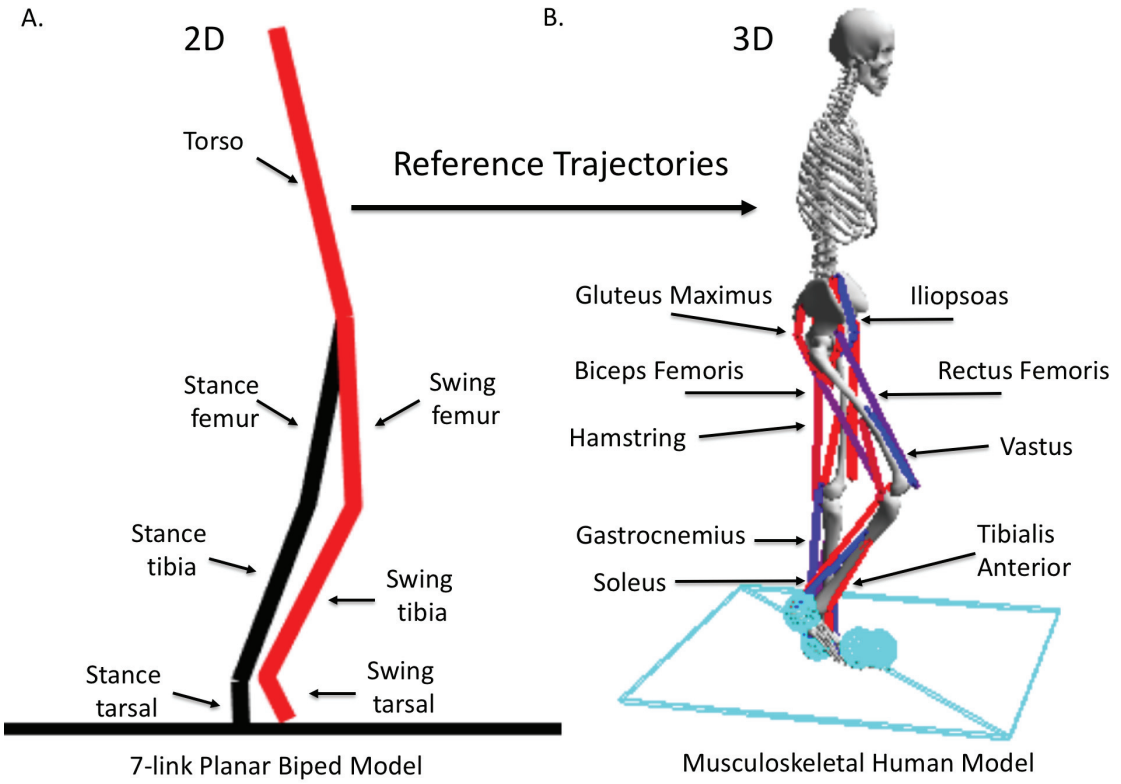


Figure 3.18: A. 7-link planar biped robot model. It comprises a single torso along with femur, tibia and tarsal for each leg. All the joints are controlled by torque. B. Musculoskeletal human model. It has 7 joints along with 9 muscles for each of the leg. The muscles are Gluteus maximus, iliopsoas, biceps femoris, rectus femoris, hamstring, vastus, gastrocnemius, soleus and tibialis anterior. The reference trajectories that obtained with 7-linked planar biped robot are used to train the neural network that controls the muscles of the musculoskeletal system.

written as follows;

$$x = [q, \dot{q}] \tag{3.6}$$

$$\dot{x} = f(x, u) = [\dot{q}, \ddot{q}] \tag{3.7}$$

whereas the 7-link biped model is a hybrid dynamical system, therefore the equations of motion are written as a combination of two underlying dynamics for each of the states of movement; single stance $F_S(q, \dot{q}, u)$ and heel-strike dynamics $F_H(q, \dot{q}, u)$

$$M_S(q)\ddot{q} = F_S(q,\dot{q},u), \quad x^- \notin S \tag{3.8}$$

$$M_H(q^-)\dot{\ddot{q}}^+ = F_H(q^-,\dot{q}^-,u), \quad x^- \in S \tag{3.9}$$

where superscripts $(.)^- and (.)^+$ denote the states before or after an event, for instance in biped locomotion it corresponds to a mechanical collusion of the foot with ground. Here, the mass matrix, $M \in \Re^{q_n,q_n}$, is written with respect to the generalized coordinates of the state information, q where g_n is the number of state variables.

The first part of the hybrid system dynamics is the single stance dynamics given with Eq. 3.11. It describes the ballistic dynamics of the swing leg while one foot touches the ground. In order to simplify the derivations of the single stance dynamics, several assumptions have been considered, for instance an absolute angle instead of a relative angle to describe the joint position and symmetry between each leg. Considering the absolute angle of each state information instead of the relative angles, allows us to end up with individual equations for each of the degrees of freedom of the multi-body system. Then the dynamics of each of the seven degrees of freedom can be written with seven linearly independent equations while constructing the system as a balance of angular momentum around each joints in the kinematic chain of the 7-linked biped robot. The first equation of motion of the 7-linked biped robot is going to be given for the first joint in the kinematic chain of the 7-linked biped robot which is the ankle joint of the stance foot as given;

$$u_1 + \hat{k}\sum_{i=1}^{7}\left((C_i - Z_0)\cdot\left(-m_i g\,\hat{j}\right)\right) = \hat{k}\sum_{i=1}^{7}\left((C_i - Z_0)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i \hat{k}\right) \tag{3.10}$$

where $\hat{k},\hat{j}$ represents the unit vectors, C_i, Z_i are the center of mass of each link and the positions of the tip points of the links respectively, with mass and inertia of each joints and gravity m_i, I_i, g. This formulation is written based on the equality of the dynamics with an external torque applied to the ankle joint of the stance food (left side of the equation) and the change of the angular momentum around the stance food, Z_0, (right side of the equation). To obtain balance of angular momentum equation of the following joint in the kinematic chain, for instance for the knee of the stance food, we subtract the first angular momentum equation due to linear independence. Therefore the angular momentum around the knee of the stance food with corresponding external torque exerted to this joint, u_2, can be written as follows;

$$u_2 + \hat{k}\sum_{i=2}^{7}\left((C_i - Z_1)\cdot\left(-m_i g\hat{j}\right)\right) = \hat{k}\sum_{i=2}^{7}\left((C_i - Z_1)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right) \tag{3.11}$$

We can continue along the further joints of the kinematic chain by following the same derivations to obtain the rest of the angular momentum equations. Except the joint of the torso since it is linked to stance and swing legs therefore this difference needs to be taken into account while deriving the angular momentum balance for the torso joint.

$$\begin{aligned}
u_3 + \hat{k}\sum_{i=3}^{7}\left((C_i - Z_2)\cdot\left(-m_i g\hat{j}\right)\right) &= \hat{k}\sum_{i=3}^{7}\left((C_i - Z_2)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right)\\
u_4 + \hat{k}\sum_{i=4}^{7}\left((C_i - Z_3)\cdot\left(-m_i g\hat{j}\right)\right) &= \hat{k}\sum_{i=4}^{7}\left((C_i - Z_3)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right)\\
u_5 + \hat{k}\sum_{i=5}^{7}\left((C_i - Z_4)\cdot\left(-m_i g\hat{j}\right)\right) &= \hat{k}\sum_{i=5}^{7}\left((C_i - Z_3)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right)\\
u_6 + \hat{k}\sum_{i=6}^{7}\left((C_i - Z_5)\cdot\left(-m_i g\hat{j}\right)\right) &= \hat{k}\sum_{i=6}^{7}\left((C_i - Z_5)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right)\\
u_7 + \hat{k}\sum_{i=7}^{7}\left((C_i - Z_6)\cdot\left(-m_i g\hat{j}\right)\right) &= \hat{k}\sum_{i=7}^{7}\left((C_i - Z_6)\times\left(m_i\ddot{C}_i\right) + \ddot{q}_i I_i\hat{k}\right)
\end{aligned} \tag{3.12}$$

Until now, we have been describing the stance leg dynamics of the hybrid system of the 7-link biped robot. The second part of the system dynamics comprises the Heel-strike dynamics which is based on the assumption that there is no time difference between timing of the swing foot lands and stance foot takes off. This assumption leads a locomotion where always only one foot touches the ground as stance food, this is called single stance dynamics. The immediate transition of stance food is known as Heel-strike map and this is what we consider while modeling the 7-link biped model in order to pursue the simplicity of the model of the biped locomotion where the biped model is required to be symmetric to hold this assumption.

The Heel-strike map describes the dynamics of the swap between the stance and swing leg as well as the impulsive collusion where the joint velocities are calculated without any joint angle change. The equations of motion that describe the Heel-strike dynamics is given in a compact form in Eq.3.12. This dynamics creates a symmetric biped locomotion, an identical left and right leg behavior in each step. Assuming that the angular momentum has to be conserved at the contact point along with all the joint positions, we can write down the linearly independent equations of the Heel-strike dynamics for all joints. Following the same principles used for the single stance dynamics allows us to solve those equations with numerical methods. We can start writing down the Heel-strike equations of the joints from the first joint in the kinematic

chain which is the stance ankle, the joint that touches the ground at the moment however about to leave the ground.

$$\hat{k}\sum_{i=1}^{7}\left((C_i^- - Z_7^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)=\hat{k}\sum_{i=1}^{7}\left((C_i^+ - Z_0^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right) \tag{3.13}$$

This equation expresses the equality between the angular momentum before and after the Heel-strike. The left hand side describes the total angular momentum of biped before the Heel-strike considering the change of leg dynamics from swing to stance. The right hand side of the equation indicates the total angular momentum of the robot after the Heel-strike while considering that the stance foot is becoming the swing foot. From the Heel-strike equation of the ankle of the stance foot, we can move along the kinematic chain to write down the Heel-strike equation of the knee joint of the stance foot:

$$\hat{k}\sum_{i=1}^{6}\left((C_i^- - Z_6^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)=\hat{k}\sum_{i=2}^{7}\left((C_i^+ - Z_1^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right) \tag{3.14}$$

We can derive the Heel-strike equations for the remaining joints in the 7-link biped robot while moving along the kinematic chain which will provide us the rest of the Heel-strike dynamics to be solved numerically in the optimal control formulation.

$$\begin{aligned}
\hat{k}\sum_{i=1}^{5}\left((C_i^- - Z_5^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)&=\hat{k}\sum_{i=3}^{7}\left((C_i^+ - Z_2^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right)\\
\hat{k}\sum_{i=1}^{4}\left((C_i^- - Z_4^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)&=\hat{k}\sum_{i=4}^{7}\left((C_i^+ - Z_3^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right)\\
\hat{k}\sum_{i=1}^{3}\left((C_i^- - Z_3^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)&=\hat{k}\sum_{i=5}^{7}\left((C_i^+ - Z_4^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right)\\
\hat{k}\sum_{i=1}^{2}\left((C_i^- - Z_2^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)&=\hat{k}\sum_{i=6}^{7}\left((C_i^+ - Z_5^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right)\\
\hat{k}\sum_{i=1}^{1}\left((C_i^- - Z_1^-)\times\left(-m_i\dot{C}_i^-\right)+\dot{q}_i^- I_i\hat{k}\right)&=\hat{k}\sum_{i=7}^{7}\left((C_i^+ - Z_6^+)\times\left(m_i\dot{Z}_i^+\right)+\dot{q}_i^+ I_i\hat{k}\right)
\end{aligned} \tag{3.15}$$

where q^- and q^+ denote the state information before and after the Heel-strike respectively. We consider the fact that there are swaps among joints due to the Heel-strike map, for instance during a single step the foot of the swing leg Z_0^- turns into the foot of the stance leg Z_7^+. The assumption of the symmetry in the legs then allows us to write down a diagonal matrix to describe the swaps between states before and after Heel-strike;

$$q^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} q^- \tag{3.16}$$

The state equation can then be written as follows;

$$x^- = \begin{bmatrix} q^- \\ \dot{q}^- \end{bmatrix}, x^+ = \begin{bmatrix} q^+ \\ \dot{q}^+ \end{bmatrix} \tag{3.17}$$

with the relationship between the states before and after heel-strike in the system dynamics is;

$$x^+ = F_H(x^-) \tag{3.18}$$

Objective Function

There are several approaches to assign a cost function to be minimized for biped locomotion; integral of torque-squared, integral of absolute work done by actuators, integral of the cost of transportation and total energy consumption during a horizontal movement are among those possible functions. Here, we are interested in a biologically plausible cost function to be able to generate a human-like movements. At the same time, we have to consider the numerical implementation of the cost functions. Therefore we study a cost function which provides both plausibility and the numerical efficiency.

Experimental studies show, that animals tend to minimize the energy consumption during locomotion (Hoyt and Taylor, 1981). Hoyt and Taylor conducted an experiment where a horse on a treadmill was forced to walk, trot and gallop with different speeds. They measured the oxygen consumption of the horse for each speed enforced by the treadmill. The experiment showed that there exists an optimum level of energy consumption for each gait. Although the authors observed that a change of the speed of the treadmill caused different gaits, the level of optimum energy consumption was more or less equal for each of the gait patterns which can be seen in Figure 3.19 where the optimum level of energy consumption is equal for each

different gaits.

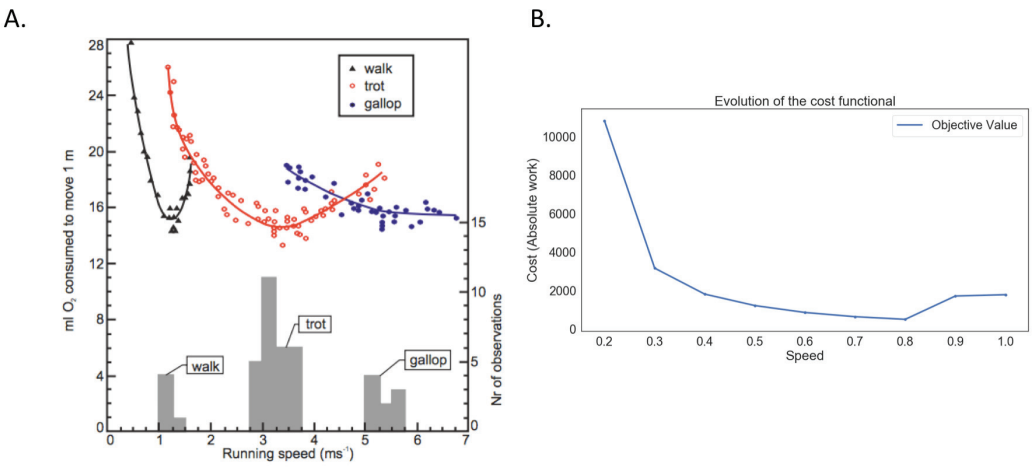


Figure 3.19: A. Hoyt and Taylor (Hoyt and Taylor, 1981) showed that there exists an optimum energy consumption level for walk, trot and gallop. B. The change of cost function according to step time while step size being fixed. The value of cost function is reaching an optimum value of "0.8".

We will use cost of transport as of our objective function to be minimized, that based on the energy to move the joints over a complete step cycle.

The objective function was to respect the discontinuties of the hybrid locomotion dynamics. The dynamics of one gait is continuous from the point that foot is leaving the ground until it hits the ground. Then the dynamics is switching. This discrete event interrupts and changes the behavior of the continuous dynamics. To study this event switches, there are mainly two possibilities, one of them is to incorporate this hybrid mechanism in to the objective function which becomes a mixed-integer nonlinear programming (Bazaraa et al., 2013). The other possibility is that instead of using entire multi-body dynamics of the system, we can conceptualize the problem where the main idea of the objective function is to find a stabilized limit cycle since the locomotion can be regarded as a periodic movement and considering that the goal is moving forward while minimizing the entire energy consumption. This assumption required to write down nature of the hybrid dynamics as a boundary condition instead of objective function. Having this idea of the objective, it can be argued that we might not need to observe the locomotion continuously, instead we can use repetitive motion of the locomotion

and consider the dynamics as a step-to-step movement. To do that, we can find a function that maps the states from the beginning of the periodic movement to the end of it, for instance we can start from the take-off and observe till the contact to the ground and expect to see the same pattern during the whole locomotion.

The solutions that we obtained with this objective function show smooth trajectories which increased the speed of convergence of the solutions. With these solutions, we also avoid having large fluctuations in the actuators therefore only those solutions are eligible for real time robotic implementations. It is called integral of smoothed absolute work done by actuators and given below;

$$\int_{t=0}^{Ts} \left(\sum_{i=1}^{7} \frac{u_i(\tau)\, v_i(\tau) \tanh\left(\frac{u_i(\tau) v_i(\tau)}{\alpha}\right)}{mgd_s} \right) d\tau \tag{3.19}$$

where T_s, d_s denotes the step size and length of the step, $u(.), v(.)$ indicates the joint torques and velocities and α is the smoothness parameter of the tangent hyperbolic function. With this approach, we avoid using an absolute function which would introduce a discontinuity to the optimization problem. This method is also called exponential smoothing of absolute work.

To generate a plausible biped locomotion, we need to consider the mechanical and numerical properties of the 7-link biped locomotion, for instance the periodicity of the movement trajectory, manipulation of the speed and the step size of the biped, and the control of switch between swing and stance leg. These properties can be integrated into the formulation of nonlinear programming as nonlinear inequality and equality constraints.

Periodicity constraint

The first boundary condition we will that the condition of periodicity: The locomotion patterns should be periodic and symmetric. It can be summarized as the equality of the initial position of the biped robot to the final position of the biped robot after a single step except the role of the legs. And at the end of a single step, the swing leg should become the stance leg and vice versa. Thus, the positions of joints must be equal at the beginning and the end of a step.

The Heel-strike marks transition from one step to another and a condition that we need to express as a boundary condition. It can be described as a Poincare map. The Heel-strike was defined as;

$$x^+ = F_H(x^-) \tag{3.20}$$

where x^+, x^- denote the state of the joints before and after the Heel-strike. Since we are required to observe periodicity in the gaits, we can also refer to x^+ as the initial position of the joints and x^- as the final position of the joints after one step of the robot. To obtain the Heel-strike map, F_H, one can integrate the multi-body dynamics from a set of initial condition and solve it numerically until the end of the single step. It is convenient to solve this equation numerically since analytical solution of the Heel-strike of a 7-link biped cannot be solved analytically in all cases. Then we can write a mathematical desciption of the first boundary condition for a stable locomotion pattern, which can be described as a forward movement and state has to map onto itself;

- For forward motion, state transition from x^+ to x^- has to be in the forward direction
- For mapping, periodicity can be given as

$$x^+_{\text{per}} = F_H\left(x^+_{\text{per}}\right) = x^-_{\text{per}} \tag{3.21}$$

To be able to create a stable gait the boundary condition as a nonlinear equality constraint can be given as finding a root of the Poincare equation

$$F^*_H(x^*_{\text{per}}) - x^*_{\text{per}} = 0 \tag{3.22}$$

This equation describes a Poincare section of the continuous system, which can be considered as one of the solution of the stable gait of a 7-link biped robot. A periodic and stable gait that obtained while considering the Eq. 3.22 is given in Figure 3.20 where the joint trajectories of the swing and stance leg are highlighted with the corresponding snapshot of the biped locomotion at the given points of the limit cycle.

Forward movement constraint

We add another constraint into the optimal control formulation to create a periodic forward movement and to avoid solutions that correspond to the standing position. The goal of this equality constraint is to create the periodic steps to generate forward movement therefore we identify the final position of the swing foot as a state control constraint which also intrinsically

allows us to control the speed of the biped locomotion. By setting up a desired position as an equality state constraint at a given time, we manage to steer the nonlinear programming to place the swing foot to the prescribed position in the ground within given end time hence the control of the speed of the biped locomotion. The equality state constraint for the speed control and forward movement can be written as follows;

$$Z_{\text{sa}}(T) = \begin{bmatrix} x_d \\ y_d \end{bmatrix} \tag{3.23}$$

where $Z_{\text{sa}}(T)$ denotes the tip position of the ankle in the swing foot at time T, and x_d, y_d correspond to the desired position at x and y coordinates respectively.

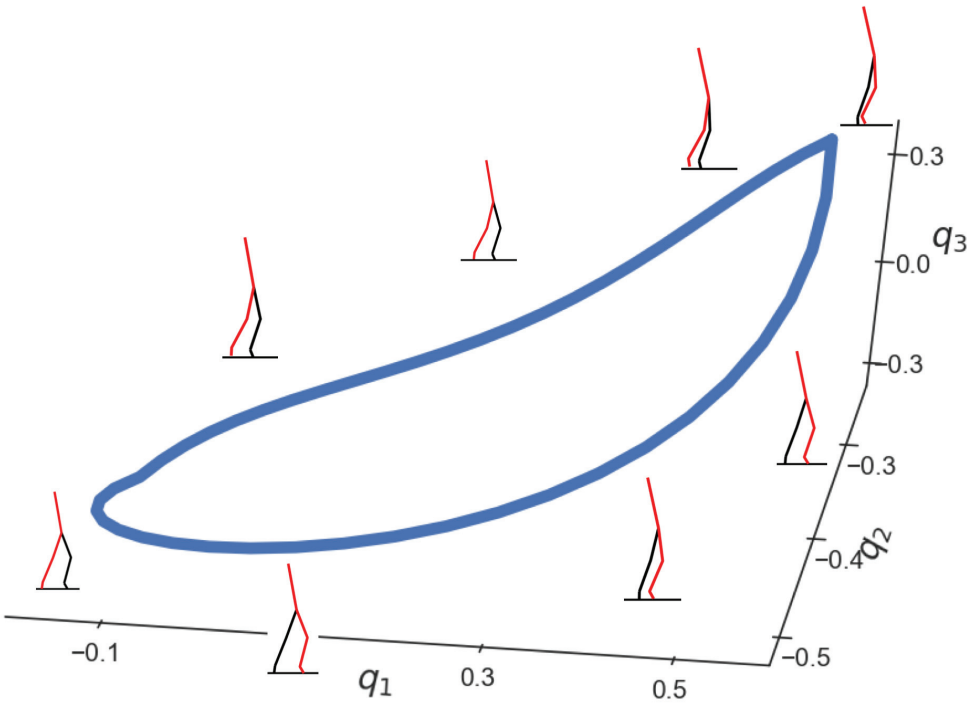


Figure 3.20: Evolution of the biped locomotion with respect to three joint positions, q_1, q_2, q_3 indicates the periodicity of the biped locomotion. In stable walking, biped locomotion follows a periodic orbit. Different states of the biped locomotion are indicated in the figure.

Timing of swing and stance legs constraint

To ensure that the solution of the optimal control is periodic and continuous, in the sense that subsequent step will be identical to the previous step, the velocity profiles of the swing and stance leg of the subsequent step have to be identical. We derive the Heel-strike equations based on the assumption that at the time the swing leg touches the ground, the stance leg must take off. Therefore an additional inequality constraint is needed which takes into account the precise timing of the swing leg's touch down and swing leg's take off. We therefore investigated the velocity profile of the swing leg along the vertical axis.

We start from the following observation: At the beginning of a cycle, when the swing leg takes off from the ground, its velocity must be positive. However, at the end of the cycle, just before the leg touches the ground again, the velocity must be negative. This can be written as a constraint for the vertical velocity profile to ensure that the swing leg trajectory will satisfy the instantaneous change from swing leg to stance leg. Since we consider the ground as an even terrain, the normal vector of the ground can be expressed as $\hat{n} = [0,1]^T$. Then, the inequality constraint of the velocity profile of the swing leg can be written as:

$$\begin{aligned} \dot{Z}_{\text{sa}}(0) \cdot \hat{n} &> 0 \\ \dot{Z}_{\text{sa}}(T_f) \cdot \hat{n} &< 0 \end{aligned} \tag{3.24}$$

Flight of the swing leg on even terrain constraint

Since the numerical solution is evaluated on a time grid, it is possible that the Heel-strike point is missed and the leg 'penetrates' the ground. This will cause to numerical solver to abort. To avoid this situation and to create a swing leg movement above the ground, we need to add another inequality constraint. Here, the creation of the flight for the swing leg above the ground can be ensured by constraining the vertical position of the swing leg joint to the positive range during the entire swing:

$$Z_{\text{sa}}(t) \cdot \hat{n} > 0 \qquad \forall\, t \in (0, T_f) \tag{3.25}$$

Flight of the swing leg on uneven terrain constraint

To incorporate also uneven terrain, we extend constraints for the position of the swing ankle joint position for even terrain. Instead of limiting the inequality constraint to the positive range of the vertical position, one can estimate the ground position and write a function that describes the relative position of the ground. This estimate can be used as an additional constraint. It enables to calculate the touch down coordinates and ensures that the swing leg's trajectory will be avoiding the ground collusion. In another word, such an inequality constraint limits the trajectory of the swing leg above the ground. Assuming that the position of the ground can be written, then the inequality constraint for the flight of the swing leg on uneven terrain is as follows:

$$Z_{\text{sa}}(t) \cdot \hat{n} > Z_{\text{gy}}(t) \qquad \forall\, t \in (0, T_f) \tag{3.26}$$

where $Z_{\text{gy}}(t)$ is the vertical position of the ground relative to a reference which can be taken as zero.

State and torque constraint

The final constraints for the optimal control formulation are imposed by the anatomical structure of biped locomotion. The viable range of motion of the joints and also the limits of possible torques in the joints have to be taken into account. Since the constraints are describing a possible ranges of the states and torques. This can be done as follows:

$$\begin{aligned} Z_{\text{i,min}} &\leq Z_i \leq Z_{\text{i,max}} \\ u_{\text{i,min}} &\leq u_i \leq u_{\text{i,max}} \end{aligned} \tag{3.27}$$

3.3.2 Point-to-Point Control of a Human Model

Similar to the Arm Control, we first test the locomotion model on point-to-point control. The goal is again to obtain the desired trajectory for only one joint while keeping the other joints in a stable position. The target trajectory is randomly generated for the right hip joint. The initial state of all the joints of the musculoskeletal system is set according to selected target trajectory of the right hip joint. This experiment is a preparation for locomotion problems where all joints need to be controlled. The objective of this experiment is to show that not only

a given trajectory can be mapped to musculoskeletal control but also joints can be kept in their initial positions while integrating this condition into the reward function. The Figure 3.21A illustrates the result of this experiment. It shows five snapshots of the right hip trajectory.

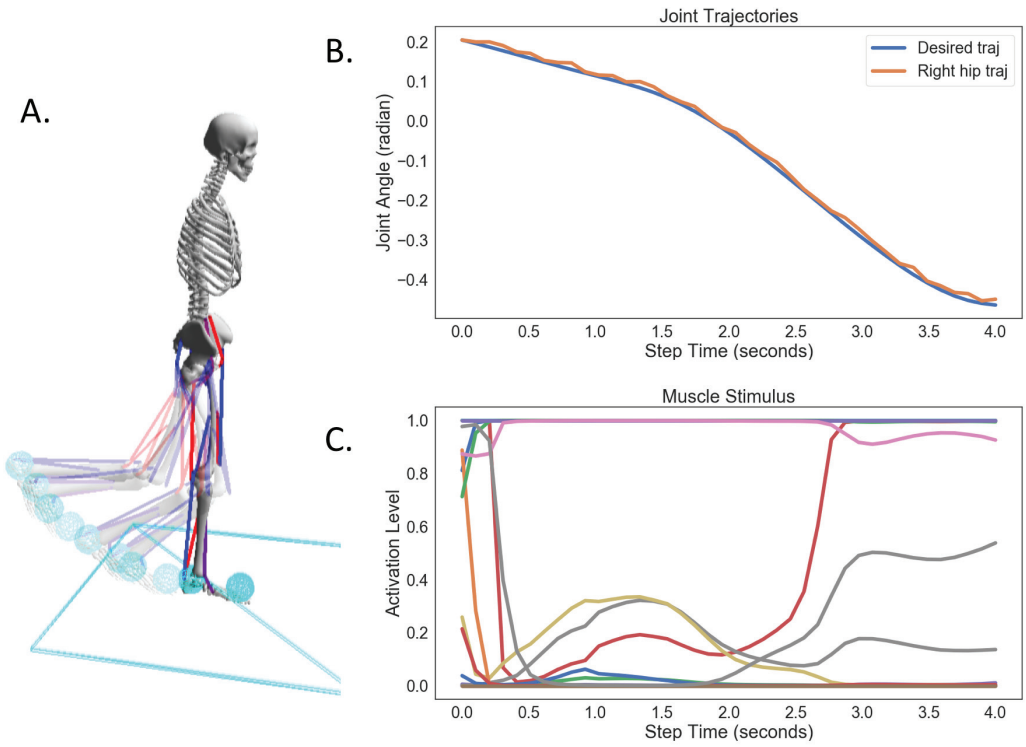


Figure 3.21: A. Movement of the musculoskeletal human model for point-to-point control problem. 7 snapshots of the movement is given. B. Desired trajectory is given by blue whereas the right hip trajectory of musculoskeletal human model is indicated by red. The desired trajectory is followed by the model after training with RL. C. Motor stimulus are given for each of the muscle. The maximum level of stimulus for a given muscle can reach "1" whereas the minimum value is "0".

Figure 3.21AB shows the trajectory of the right hip joint and the target. The movement of the hip joint deviates slightly from the target path and is also not as smooth as desired trajectory. However, these deviations are small enough to be neglected. In Figure 3.21C, stimulus that is exerted onto muscles is given. We observe that several muscles receive either the highest possible activity, 1, while others are not activated at all and the other muscles time changing stimulus. One can observe that shortly after onset of movement, the extensor muscles get

activated before being deactivated again, right after that the flexor muscles get activated. This coordinated activation-deactivation-activation sequence is aligned with the target. The initial state of the musculoskeletal system was assigned to be in forward movement at the right hip joint, while the movement continues with pulling the right leg back therefore flexor muscles are getting activated in order to follow the desired trajectory. As it can be seen, near the ending of the movement, due to fact that the goal is stabilizing the right leg at the end of the movement for the determined point in state space, those muscles that are responsible for flexor are also getting stabilized which yields a stable control of the right leg through the desired trajectory.

In this experiment, we applied our approach to a musculoskeletal model with more degrees-of-freedom than a human arm. We showed that our learning framework can be scaled up to systems with many DOF. Moreover, we observed that the activation of muscles was consistent with the expectation of orderly muscle recruitment.

However, as the control is overactuated, there may be other combinations of muscle recruitment that can lead to the same trajectories. Similar to the possibility of creating the same output from different synaptic weight distributions of the same recurrent neural network. Thus, we claimed that the proposed learning framework can produce solutions that can precisely recruit the different muscles such that coordinated motor behavior of a musculoskeletal human model can be learned.

3.3.3 Locomotion Control of a Human Model

The optimal control is a promising approach to study human biped locomotion. Majority of optimal control implementations have been considered for real-time control since the solutions are not only robust but also mathematically optimal. It has been used not only for biped locomotion but also quadrupedal locomotion as well and the solutions are implemented for the real-time hardware control (Xi and Remy, 2014; Hutter et al., 2011). In this section, we started to examine musculoskeletal walking where we used a musculoskeletal human model that has 7 joints and 18 muscles. We explored the trajectory mapping of 7-linked planar biped robot onto 7-joints musculoskeletal human model. The optimal gait trajectories were found by the optimal control implementation of the 7-linked biped robot and these reference trajectories were used as desired trajectories to be obtained in the musculoskeletal human model.

The solution we obtained with optimal control is based on single-stance dynamics where only one foot is on the ground and aimed at following this desired trajectory in the musculoskeletal walking. Three instances of the solutions can be found in Figure 3.22, where depending on the parameter setup, we obtained fast, optimum and slow walking patterns. In order to obtain different walking patterns, we used the step size and step time parameters as equality constraints of the optimal control in which optimization searches for a periodic solution which guarentees the desired step size and time. In order to obtain the fast walking pattern, see Figure 3.22a, we increased the step size and decreased the step time whereas the walking pattern is obtained with smaller step size and time. In order to find out the optimum walking gait, we run the optimization with different parameter values for the step size and time. As it is given in Figure 3.19, the optimum gait is obtained with a step length of "0.6m" and step time of "0.5s".

Walking Patterns

The reference trajectories were found by numerical integration of optimal control, and the methods has been explained in Section 3.2.1. Since the step length and step size are the parameters of the optimal control formulation, we conducted several experiments to obtain different speed in the walking pattern of 7-linked biped robot. Ranging these parameters from 0.2 to 1.0 for both of them, we obtained several walking patters, such as slow, optimum and fast walking. In all cases, the main constraint was that the solutions were to be periodic, as shown in Figure 3.23. The objective function that has been tied to the system dynamics allowed us to evaluate the optimality of the solutions. Since the change of the parameters can be observed in the change of value of the objective function, we can evaluate the performance of different parameter setup. Such as, the final value of objective function for fast walking was "1.0826e+04" given the step length was "0.8m" and step time was "0.3s", and similarly the final value of objective function for slow walking was "1.2723e+03" given the step length was "0.3m" and step time was "0.6s". However we obtained a final value of objective function, "205.1425" for the step length of "0.6m" and step time of "0.5s".

(a) Fast Walking

(b) Optimum Walking

(c) Slow walking

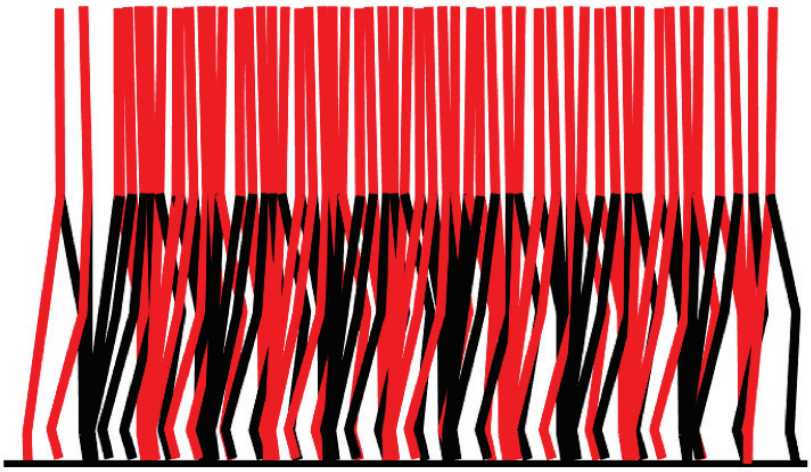


Figure 3.22: Three different walking patterns of 7-linked planar biped robot with optimal control. a. A fast walking pattern is obtained with the step length of "0.8m" and step time of "0.3s". b. One of the optimum walking pattern is obtained with step length of "0.6m" and step time of "0.5s". c. A slow walking pattern is obtained with step length of "0.3m" and step time of "0.6s". In all figures, two different colors are used to differentiate legs and color stays same during illustration and torso is always indicated by the color of red.

Learning the walking patterns

In Figure 3.23, the corresponding target joint trajectories are shown. We used only the optimum walking (Figure 3.23b) trajectory as the reference trajectory. The resulting walking patterns of musculoskeletal system are shown in Figure 3.24.

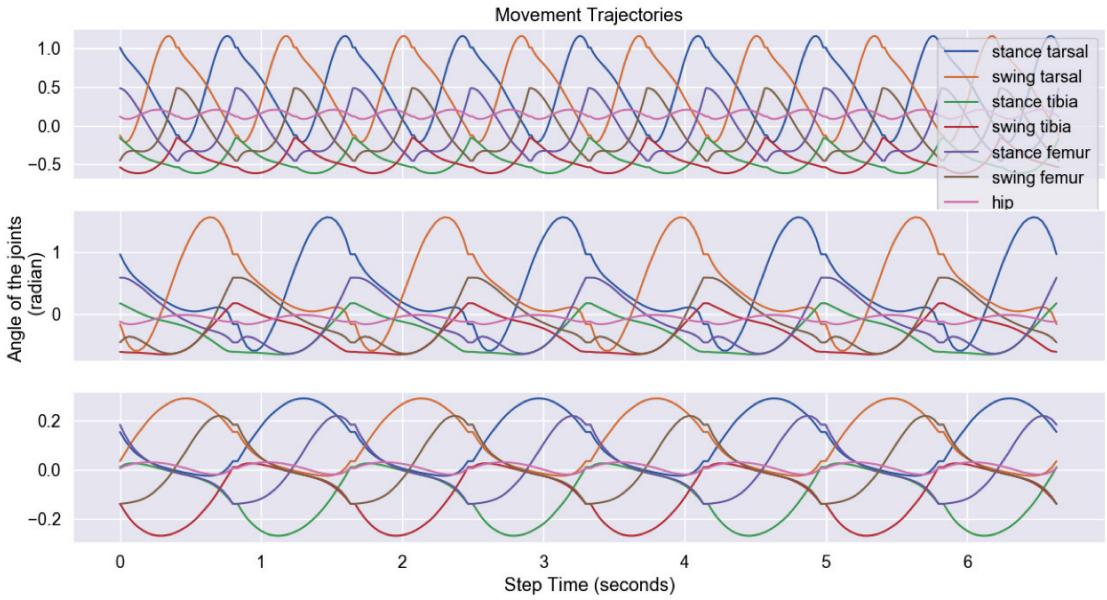


Figure 3.23: The reference trajectories of each slow, optimum and fast walking patterns. The optimum trajectory is used as reference trajectory for the musculoskeletal locomotion.

Although walking patterns similar to the solution of optimal control are observed, the solutions were only stable enough for 2 steps. In the musculoskeletal model, the robot was allowed to start the forward movement, even though initial configuration of the system is in vertical standing position. However musculoskeletal robot had tendency to lift the swing leg higher than the desired movement trajectory and perform the similar movement for the next step which caused the torso of the musculoskeletal robot to lean forward while initiating the next step. Consequently, the distance between two legs increased in each step therefore it required higher muscle contraction to initiate the swing movement in the following step which ended up being not possible.

The implementation of the 7-linked biped robot is based on single-stance dynamics, however musculoskeletal simulation shows that 2D single-stance dynamics is creating a hard constraint onto the musculoskeletal simulation where double-stance dynamics is observed. We believe

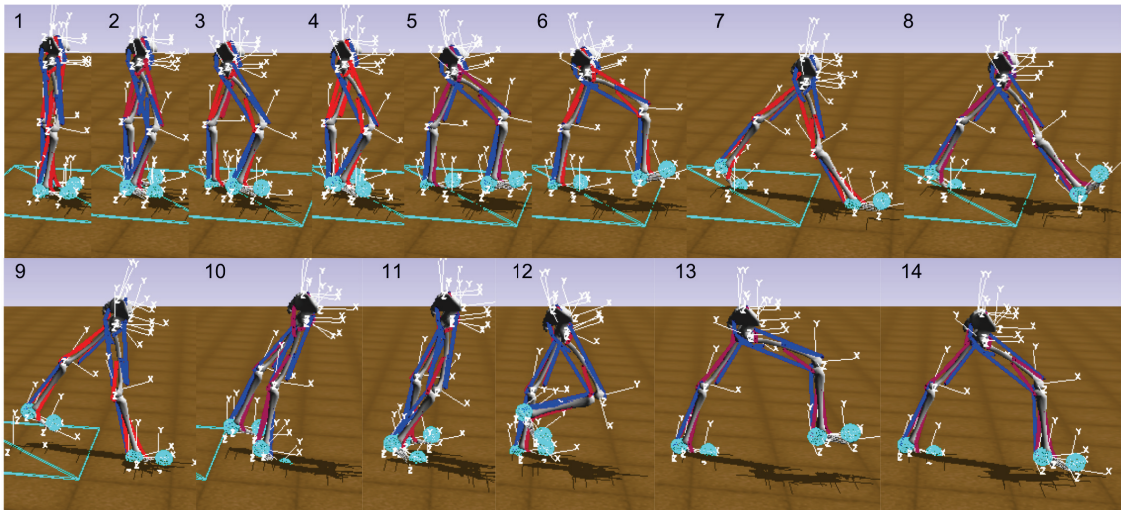


Figure 3.24: 14 Snapshots of the musculoskeletal walking after training the neural network with optimum reference trajectory are given. The orders of the snapshots are indicated with numbers at the corner of each snapshot.

that integration of the double-stance dynamics into the optimal control formulation as one of the constraint will improve the simulations of the musculoskeletal system. Apart from that the impulse forces between the ground and on the initiation of single-stance are not captured by the reference trajectories since it is not considered in optimal control formulation. Additionally, it has to be required that velocities of the contact points either in stance-dynamics or double-stance dynamics after the collusion have to set zero in order to keep the movement periodic. The major effect of lacking double-stance dynamics and impulse forces can be seen in the transition of successive steps. In order to maintain the successive steps, musculoskeletal system is needed to be informed about the impulse forces since they take part in the initiation of following steps. One another possible improvement can be achieved by studying correlation between parameter setup of 7-linked biped robot and the outcome of those parameters on impulse forces such that stability of the locomotion can be observed.

We used an actor-critic network with three hidden layers with [256, 512] neurons for each layers. The critic network is trained with temporal difference algorithm whereas the actor network parameters are learned with policy gradient. The solution we obtained shows that the actor-critic network with this parameter and structure setup managed to converge a local optimum albeit it was not sufficient to obtain a stable locomotion pattern. After training, the musculoskeletal system managed to solve the first initial problem which is the initiation of

the movement since the musculoskeletal system's initial position is vertical standing position. It managed to push itself upward to lift one of the leg then create a momentum to push the system forward. After the initial stepping, musculoskeletal system solved the initiation of the swing phase for standing leg and finishing the swing phase of the moving leg. This shows that we managed to map the periodic and symmetric movement of the optimum solution into the musculoskeletal system. Although there has been two stable steps during the test case, the musculoskeletal system stayed stable at the end of these two steps. It can be claimed that the training of the actor-critic network has finished prematurely.

3.4 Conclusion

In this section, we focused on the musculoskeletal walking problem where we used the reference trajectories of the optimal control to train an actor-critic network for muscle control. The objective of the optimal control was to obtain periodic walking gaits with different speed and step size for a 7-linked biped planar model. These biped models are commonly referred as reference models to study the bipedal locomotion problems (Yang et al., 2009; Pratt et al., 2001; Grizzle et al., 2009; Westervelt et al., 2003). The solutions that obtained show periodicity and symmetry, see Figure 3.23 due to the equality constraints of the optimal control formulation. Based on the equality constraints of the optimal control formulation, the gait patterns were left-right symmetric and periodic. We used torque-squared objective function to obtain a solution which has minimal energy consumption.

The solution that we obtained for the fast walking has a pattern of swing foot which is not similar to human-like walking. The trajectory of the swing foot indicates a movement where the robot first lifts the leg as much as possible and then swings it to the stance position. Although this solution satisfies all constraints and system dynamics, it is not a human-like movement. In order to avoid such solutions, an additional constraint is needed. We formulate this constraint for the hip and knee joint in a way that the swing foot cannot move beyond these boundary conditions of hip and knee joints. For slow walking, we observed that the foot was almost vertical to the ground with very little changes of angle. Although this pattern can be seen in human locomotion, slow walking doesn't necessarily require walking on the tip of the feet. Also this solution can be suppressed by introducing another boundary constraint for the foot joint. One particular difference between these walking patterns is that the torso position is aligned with the speed of the walking. For instance during fast walking, the angle of

the torso joint is always positive, indicating that body is leaning forward to speed up walking. However, walking patterns that can be considered optimal show a slight backward leaning. In addition, the torso of the planar robot almost always stays vertical in slow walking.

After the obtaining the optimum reference trajectories for biped locomotion, we trained an actor-critic network to map these solutions into musculoskeletal human model. The actor-critic network we used is identical to arm control experiments. Even though, we successfully obtained walking patterns in human musculoskeletal system, our learning and optimization framework has severa several limitations.One of the outcome of these limitations is that the gait we obtained was not stable. To tackle with this unstable walking gait, the actor-critic network and the simulation properties have to be adapted with respect to the complexity of the system. For instance, longer training iterations and an actor-critic network with more than three hidden layers as well as higher neuron numbers will enhance the quality of the musculoskeletal locomotion. In addition to the network properties and learning setup, a human locomotion with 7 muscles in each leg without active control of torso decreases the probability of a successful training. Therefore, a more realistic musculoskeletal design is the first necessary step to improve the quality of the solutions.

Another caveat of our method is the dependency of the domain-specific knowledge such that the simplified 2D planar biped model is used to train a 3D and higher degrees-of-freedom musculoskeletal system. Any difference between these two models has potential to create an unsuccessful solution for the musculoskeletal locomotion. Apart from the difference between the simplified and musculoskeletal system, the parameter setup of the muscle model is also required to be a part of optimization procedure. It has been shown that muscle parameters have important effect on the convergence of the RL and also the obtained locomotion pattern (Geijtenbeek et al., 2013; Peng and van de Panne, 2017). Therefore, one either needs to consider a muscle parameter search in addition to trajectory learning or the model of the musculoskeletal system with muscles needs to be adapted to the anatomical available human data.

4 Reverse engineering the motor circuit

4.1 Motor Circuit

The goal of this chapter is to reverse engineer the basic components of the neural motor control loop; including high-level motor circuits, spinal cord and the musculoskeletal system. We will focus on the control of the human arm. We will explain the components of our motor control model, starting from the muscle dynamics and how muscles are recruited, the generation of the motor commands by pools of alpha motor neurons in spinal cord to the integration of proprioception signals that carry the sensory information of the system dynamics. Then we will explain the experimental design we used to study sensory integration and motor learning in the spinal cord. We will then examine the functional role of the closed-loop between spinal cord and high-level cortical areas. The sketch of the main components of motor control system that is considered in this chapter is given in Figure 4.1.

4.1.1 Computational Muscle Model

Incorporating the properties of muscle dynamics into mathematical models pave the way for musculoskeletal simulations where different hypothesis of the underlying neural motor control can be examined. There exists a large number different muscle models, that differ in the level of description, their complexity and behavior being modelled. The difference of most models are either focus on how microscopic processes at the level of the tissue generate function or generic functional models that describe the input-output relationship between muscle activation and force generation (Zajac, 1989). The microscopic level of modeling approach of muscles takes into account the cross-bridge dynamics and the sliding activity of the filaments, e.g. Huxley muscle (Huxley, 1974) whereas the generic macroscopic models

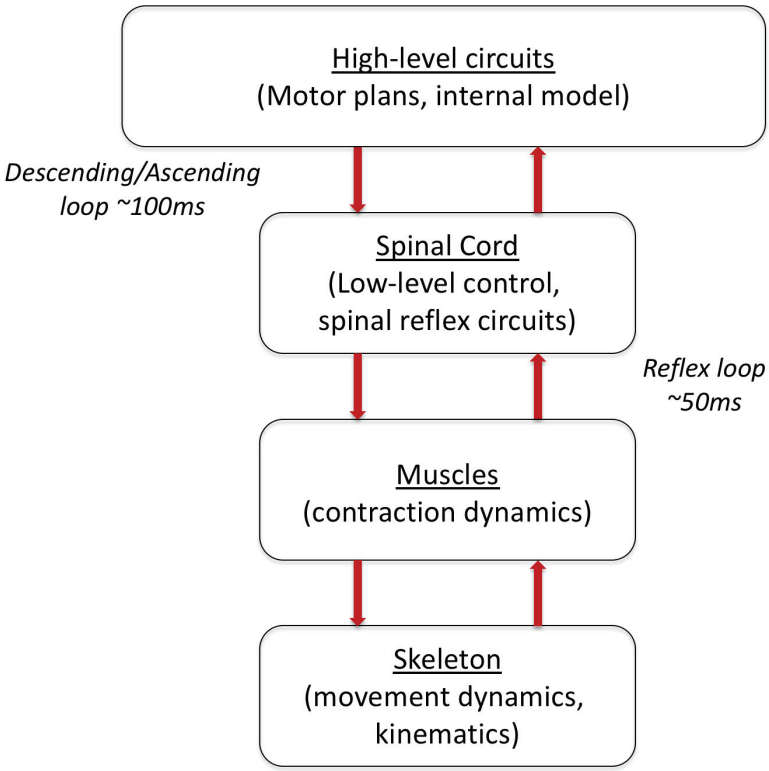


Figure 4.1: The closed-loop motor control circuit. Three main sub-components of the motor control circuit is considered, musculoskeletal system, low-level circuit (muscle-spinal cord), high-level circuits (spinal cord-high-level motor circuits).

focus on more high level properties of the muscles such as the force-length and the force-velocity relationships e.g. Hill-type muscles (Geyer and Herr, 2010; Millard et al., 2013; Thelen, 2003; van den Bogert et al., 1998). Since the microscopic properties of the muscle tissue are not relevant to biomechanical simulation studies, so-called Hill-type muscle models are widely used instead of microscopic models. Sensory information can also be extracted from these models due to the fact that the length, velocity and force are calculated within these models.

Muscle model consists of several so-called muscle-tendon units (MTU), which can be seen in Figure 4.2. The MTU define the dynamic properties of a muscle. A muscle tendon-units has two parts: an elastic part, the tendon (Figure 4.2A) and contractile part, muscle unit (Figure 4.2B). These two parts are serially connected. The muscle unit is again composed of two parts: a contractile part and an elastic element, but this time they are in parallel

(see Figure 4.2). The combined activity of those sub-units generates the force that causes the movement in the skeletal system. Therefore, muscle force is generated by these serial and parallel elements of the MTU. As it can be seen in Figure 4.2, the contractile element $f^L(l^M)f^V(l^M)$ with parallel elasticity element $f^{PE}(l^M)$ composes the muscle unit which in turn generates the muscle force f^M. The parallel elasticity element becomes active in case of excessive stretches in the contractile element or shrinkage of the contractile element beyond the acceptable range in order to prevent muscles tendon unit to be collapsed. In addition, serially connected tendon unit provide series elasticity $f^T(l^T)$ which in turn creates tendon force f^T and it takes part in the force-length and force-velocity profile of the muscle.

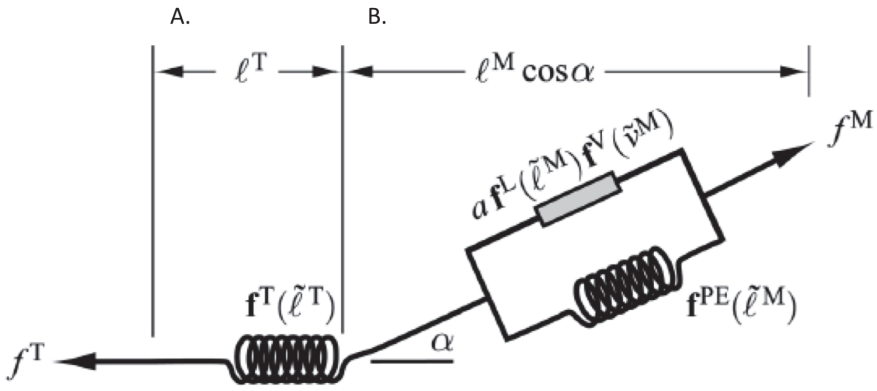


Figure 4.2: Muscle-tendon unit is comprised of muscle and tendon unit. In muscle unit, there exists a contractile element and passive parallel element. Tendon is connected to muscle unit in series. The relationship between the maximum active force and length can be given with active-force curve, described by $f^L(l^M)$. The nonlinear effect of the rate of lengthening on the force is calculated by force-velocity profile, $f^V(l^M)$. Passive-force relattionship is obtained by $f^{PE}(l^M)$ whereas $f^T(l^T)$ represents the tendon-force relationship. Figure from (Millard et al., 2013)

As is sketched in Figure 4.3, the stimulus of an alpha motor neuron is transformed to muscle activation through its nonlinear activation dynamics which in turn causes the muscle to contract. The activation of the muscle is driven by the stimulus of the motor neurons as a first order differential equation which takes into account the neural delay with parameter τ:

$$\dot{a} = \frac{u - \hat{a}}{\tau} \tag{4.1}$$

The contraction of the muscle depends on the joint states of the musculoskeletal dynamics, which in turn depends on the states of the muscles. Therefore we are faced with a recurrent nonlinear dynamical system. According to the Hill-type muscle modelling, the force is calculated while considering the force-velocity, force-length and the active stimulation of the muscle as given below;

$$f^M = f_0^M \left(a f^l \left(l^M \right) f^v \left(v^M \right) + f^{PE} \left(l^M \right) + \beta v^M \right) \cos \alpha \tag{4.2}$$

where f_0^M is the peak force at length l_0^M. After the integration of the muscle and joint positions and velocities, the continuous time-series of the stimulation is translated into the force, generated by the muscle sub-units. Hence, the activation of the muscle by a stimulus generates the active tension in the muscle model with the evaluation of the active contractile element, a passive elastic element and the tendon dynamics.

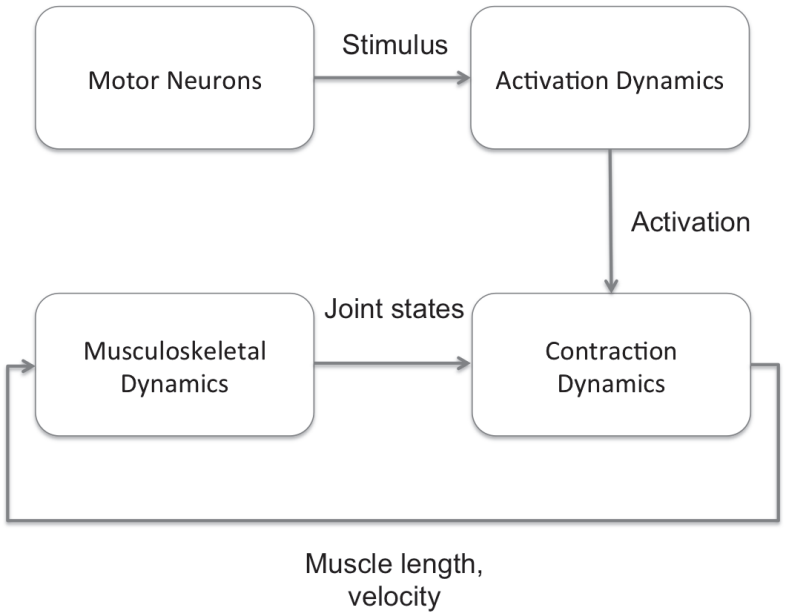


Figure 4.3: The overall schema of the force generation by muscle, given a stimulus of a alpha motor neuron through the activation, contraction and musculoskeletal dynamics

The actuation of the joints by the muscle force nonlinearly depends on the force-velocity and force-length relationships of the muscle which are modeled based on the studies of Archibald

Hill (Hill, 1938). The force-length relationship of the muscle is a result of the properties of sarcomeres which is the main cause of the active force generated by the muscles and it has bell-shaped curve and it creates passive force after a certain point of stretching. The force-length relationship of the muscle is given in Figure 4.4. The equation that represents the force-length relationship of the muscle is given by;

$$f^{l}\left(l^{M}\right)=\exp\left(c\left|\frac{l^{M}-l^{\text{opt}}}{l^{\text{opt}}w}\right|^{3}\right) \tag{4.3}$$

where l^{opt} is the optimum length of the muscle and w is the parameter for the force-length relationship of the sarcomere.

It can be seen that the total force exerted by the muscle is the combination of the active and passive forces generated by the contractile element and the passive elastic element. Muscles have also dynamic properties such that there is a relationship between the shortening velocity of the contractions and the force generation which represents the actin-myosin cycle. It has the effect on the force generation during nonisometric contractions. Instead of a bell-shape curve, force-velocity has an inverted sigmoidal property, which can be seen in Figure 4.4B. One important aspect of the force generation during the nonisometric contraction is that the force is generated if the muscle is stretched more than the threshold length regardless of the situation of the muscle, whether it is activated or not.

The force-velocity relationship of the muscle is given by $f^{v}(v^{M})$ which represents the concentric contraction, represented as inverted sigmoid.

$$f^{v}(v^{m})=\begin{cases}\frac{v^{\max}-v^{M}}{v^{\max}+Kv^{M}}\\ N+(N-1)\frac{v^{\max}+v^{M}}{7.56Kv^{M}-v^{\max}}\end{cases} \tag{4.4}$$

where $v^{\max}$ is the maximum possible velocity and with parameters N and K. The passive force-length $f^{PE}(l^{M})$ curve is given by;

$$f^{PE}(l^{M})=\begin{cases}k_{1}\left(\exp\left(k_{2}l^{M}\right)-1\right)\\ 0\end{cases} \tag{4.5}$$

The new length of the muscle after one time interval $[t_n, t_{n+1}]$ depends on the contraction velocity of the muscle. In order to calculate the numerical new length, length-velocity equation is solved with trapezoidal rule;

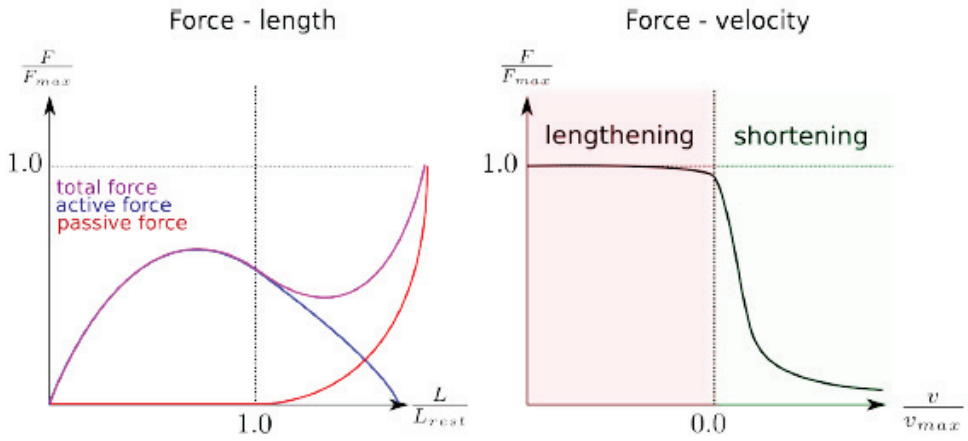


Figure 4.4: The force-length and force-velocity profile of the muscle tendon unit. Left: Force-length relationship of the muscle tendon unit shows that the force is generated by the combination of the active and passive force Right: Force-velocitiy relationship of the muscle tendon unit has two distict phase, either during the muscle is lengthening or the muscle is shortning. Figure from (Dzeladini, 2019)

$$l^M(n+1) = l^M(n) + 1/2\,(t_{n+1} - t_n)\left(v^M(n+1) + v^M(n)\right) \tag{4.6}$$

and the velocity of the muscle is found by the inverse of the muscle force-velocity equation;

$$v^M\left(f^v\right) = \begin{cases} v^{\max}\frac{1-f^v}{1+Kf^v} \\ v^{\max}\frac{f^v-1}{7.56K(f_v-N)+1-N} \end{cases} \tag{4.7}$$

Above mentioned equations that describe the MTU functionality provide not only the force generation but also information about the intrinsic properties of the muscles to be used as proprioception signals such as length, velocity and generated force of the muscle. Based on the dynamics of the muscle tendon units, we can examine the low-level closed-loop neural circuit in spinal cord. The first priority of the neural motor control circuit development is to identify the signal flow between sensory organs and alpha motor units which leads us to understand primitive structure of the spinal cord circuits, such reflexes.

4.2 Reflex Circuit Model

The transmission of the knowledge of the external stimulus is encoded in the organizational schema of the spinal cord circuits which can be seen in the wiring patterns of the spinal cord circuitry. The synaptic transmission of the reflex signals and the commanding signals of the high-level motor circuits form synapses in the neuronal circuitry of the spinal cord. The interneurons in the spinal cord receive the sensory signals through the reflex pathways and they also have synapses with the descending signal pathways. This integration of the spinal reflex and descending motor commands shows that the spinal reflex pathways have important roles in not only automatic responses of the external stimulus but also in participating the complex behaviors such as walking, jumping, running and so on.

Reflexes can be described as involuntary responses of motor circuits to external stimulation. Stimulation of the sensory organs of the muscles triggers movements in the skeletal system in order to prevent harm to the joints and muscles. Reflexes are also used by the CNS to generate voluntary movements. In these movements, reflex circuits are activated by the motor circuits according to the task or the behavioral goal. Although the details of spinal circuits are yet to be discovered, their role has been abstractly identified in experiments.

4.2.1 Basic Reflexes

The cutaneous flexion withdrawal reflex is an example of a protective reflex, where an external stimulus causes an abrupt movement of the joints. To create abrupt movements, all the flexor muscles must be recruited directly, via the connections between sensory organs and the motor neurons. Excitation from the motor neurons cause the flexor muscles to contract while the extensor muscles relax. The principle schema of the connectivity of a spinal reflex circuit is shown in Figure 4.5. Ia and II afferent sensory pathways have excitatory synapses with motor neurons, the one that originates from the same muscle has direct excitatory connections of the alpha motor neuron whereas the activity of the alpha motor neuron which has connection to the synergistic muscle is inhibited due to inhibitory neurons in between sensory signal and the alpha motor neuron.

The stretch reflex is the most studied reflex mechanism. It directly links sensory signals of the contraction dynamics of the muscle to the alpha motor neurons. The reflex circuit for the stretch reflex has largely the same architectural properties as the flexion withdrawal reflex,

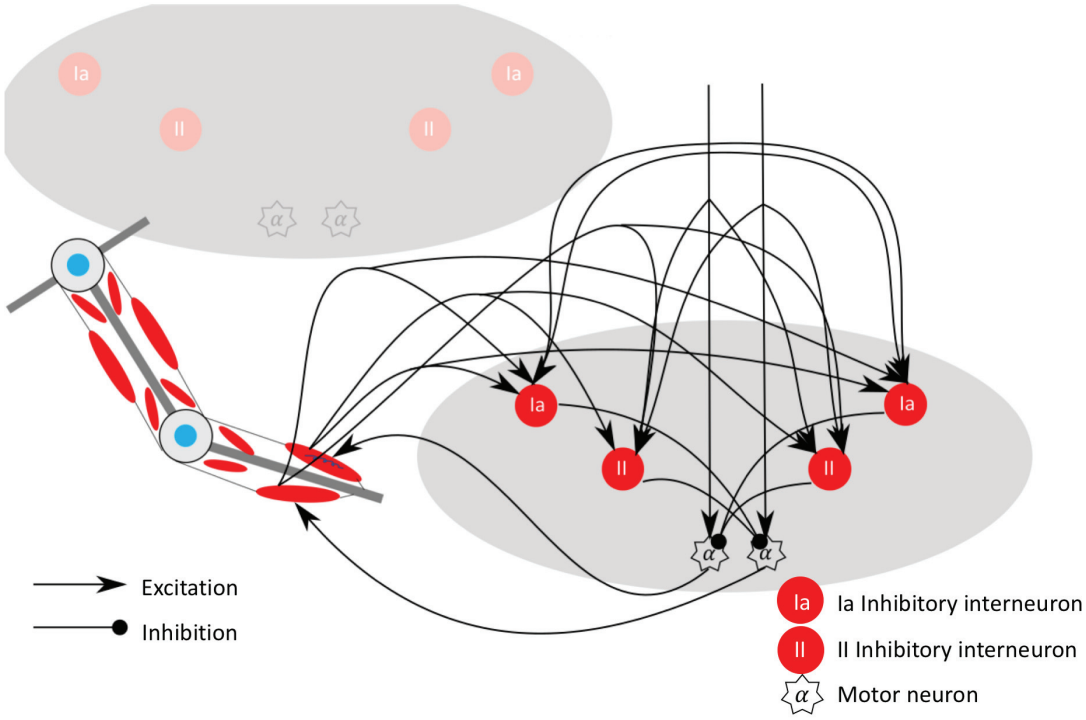


Figure 4.5: Representation of connections within a spinal reflex circuit. Ia and II sensory pathways form excitatory synaptic connections with interneurons, while interneurons have inhibitory connections to alpha motor neurons. The modulatory descending pathways have connections with inhibitory neurons and alpha motor neurons to adjust the spinal reflex circuit activity which is trasmitted to the muscles with alpha motor neurons

with the main difference that the pathways in the stretch reflex circuits are monosynaptic. The sensory information for the stretch reflex originates at the muscle spindles which are distributed between the extrafusal muscle fibers. The fibers that make up the muscle spindles are called intrafusal fibers and contract parallel to the extrafusal muscle fibers. Due to the anatomical arrangements of the muscle spindles, they encode the length change of the muscle fibers. They also provide information about the relative position of the joints. Thus, muscle spindles play a very important role in sensory motor control. The activity of the intrafusal fibers is correlated with the contraction of the muscle fibers: the more the muscle fibers are stretched, the more the firing rates of sensory nerve endings increase and the opposite in the case of muscle shortening. This activity is transmitted to the interneurons of the spinal cord.

From the computational point of view, a change of the angle of a joint corresponds to a change in muscle lengths which in turn changes the activity of the muscle spindles which is then

transmitted to the interneurons of the reflex circuit. The activity of the alpha motor neurons controls the contraction of the extensor and flexor muscles which causes a torque at joints which in turn changes the angle, velocity and the acceleration of the joint. The changes of the muscle length are integrated by the reflex circuit and contribute the activity of the alpha motor neurons which in turn stimulate the extensor and flexor muscles.

4.2.2 Spinal Reflex Model

We constructed the spinal reflex model based on stretch reflex circuit given in Figure 4.5. Model consists of a two-layer feedforward neural network where the first layer represents the spinal inhibitory neurons and the second layer represents the alpha motor neurons as output layer. There is excitatory connections between sensory signals and spinal inhibitory neurons, whereas connections between spinal interneurons and alpha motor neurons are placed as inhibitory in order to align with stretch reflex connectivity. The training phase of spinal reflex circuit is conducted without any descending modulatory signals. Based on the architecture of the spinal reflex circuit, alpha motor neuron activities are written as follow;

$$m_i = \sum_{j=1}^{n} W^I_{Ia,j} \left(\sum_{i=1}^{n} \left(W^E_{Ia,j} S_{Ia,i} \right) \right) + \sum_{j=1}^{n} W^I_{II,j} \left(\sum_{i=1}^{n} \left(W^E_{II,j} S_{II,i} \right) \right) \tag{4.8}$$

where $W^I_{Ia,j}$ and $W^I_{II,j}$ are the inhibitory weights between spinal interneurons and alpha motor neurons for Ia and II interneurons respectively and S_{Ia} and S_{II} denotes afferent sensory signals. In order to train the network, we designed a twitching experiment. We used an Hebbian-type learning rule to obtain the weights of the connections. Hebbian learning rules are known as correlative learning rules such that modification of the connections is proportional to the activity of the pre-synaptic activity v_j^{pre} and the post-synaptic activity v_i^{post}. The compact form of the learning equation takes on the following form:

$$\frac{d}{dt} w_i = F\left(v_i^{\text{post}}, v_j^{\text{pre}} \right) \tag{4.9}$$

Depending on choice of the function F, Hebbian-type learning can lead to detecting positive or negative correlation between pre- and post-synaptic activities. Due to the connectivity pattern of the reflex circuits, we deployed Adaptive principal components extraction algorithm which is comprised by two different hebbian type learning: an anti-Hebbian-type learning rule for the connections between alpha motor neurons and spinal interneurons and a hebbian

type learning rule for the connections between spinal interneurons and sensory signals.

The anti-hebbian learning rule for inhibitory connections is given as follows:

$$\frac{d}{dt} w^I_{x,j} = \eta^I \left(v_i^{\text{post}} v_{i-1}^{\text{post}} - \left(v_i^{\text{post}} \right)^2 w^I_{x,i} \right) \tag{4.10}$$

where η^I is the learning rate, $v_i^{post} = m_i$ and x represents either Ia or II connections. Connections between sensory signals and spinal inhibitory neurons are trained by a Hebbian-type learning which is given below:

$$\frac{d}{dt} w^E_{x,j} = \eta^E \left(v_i^{\text{post}} v_i^{\text{pre}} - \left(v_i^{\text{post}} \right)^2 w^E_{x,i} \right) \tag{4.11}$$

where η^E is the learning rate and post-synaptic activity is given by:

$$v_i^{post} = \sum_{j=1}^{n} \left(W^E_{x,j} S_{x,i} \right) \tag{4.12}$$

and pre-synaptic activity is $v_i^{pre} = S_{x,i}$.

4.2.3 Twitching Experiments

In order to find the connectivity pattern within the spinal reflex circuit, we designed a twitching experiment where the objective is to identify the correlation between the activity of inhibitory neurons and the changes of the sensory information. Our hypothesis is that function of the alpha motor neurons is encoded in the connectivity of the spinal reflex circuit and the recruitment of the alpha motor neurons provide a correlative control of the antagonistic muscles. A straightforward consequence of this experiment is to find the role of each muscle according to the activity of the corresponding sensory organs such as muscle spindles, for instance it enables us to identify the extensor and flexor muscles individually.

The experiment consists of activating individual alpha motor neurons with maximum value of stimulus which is "1". When a single alpha motor neuron is activated with twitches, it contracts the corresponding muscle which in turn causes skeletal joints to generate movement. Due to the change of the position of the joints, all the other muscle lengths and velocities change therefore it activates sensory organs. Figure 4.6 shows a result of activity changes in sensory organs due to twitching signals.

Our objective is to identify the pattern of connectivity between sensory receptors and in-

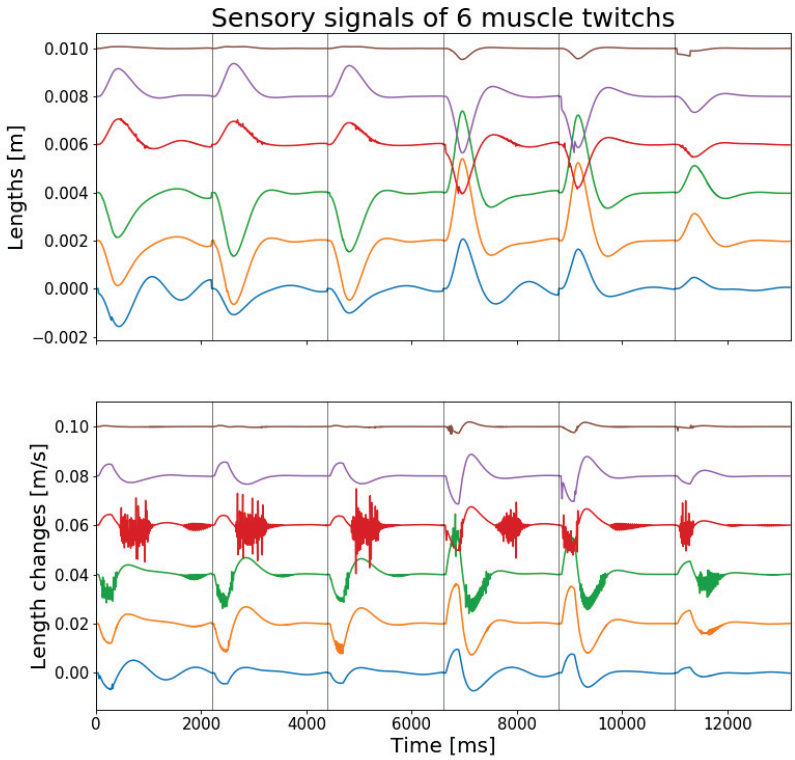


Figure 4.6: The activation of sensory signals in all muscles due to the recruitment of a single alpha motor neuron. In this experiment, two different alpha motor neurons are recruited, each receives three unit pulses consecutively.

terneurons/alpha motor neurons in the spinal cord circuitry. To this end, we assume that the sensory signals coming from different muscles convey the information about the connectivity pattern of the spinal cord reflex circuits. When an individual motor neuron is activated, it leads to a movement directly correlated with the activity of a motor neuron which in turn encodes the activity of the sensory organs. This information is then transmitted to the spinal reflex circuitry and is distributed to the motor neurons via this connectivity pattern.

We used the Hebbian learning rule that are given in Eq.4.10 and Eq.4.11 which is a self-organizing/unsupervised learning algorithm that can discover correlation patterns in the activity of the sensory receptors correlated and the alpha motor neuron activities. We used this Hebbian type learning rule to create the sensory mapping of the proprioception. This

learning rule is a correlation learning algorithm (Kung and Diamantaras, 1990) with the aim to identify the statistical properties of the proprioception signals. The data obtained through the twitching experiments is used to train the spinal reflex circuit with above mentioned hebbian learning rule to obtain the corresponding weights of the network, which is given in Figure 4.7.

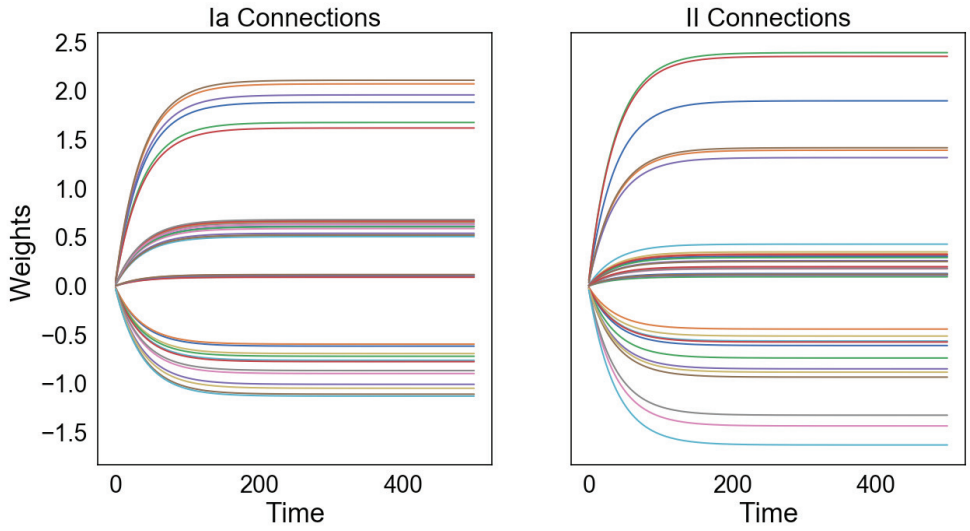


Figure 4.7: The connections obtained during the training of the spinal reflex circuit with hebbian type learning rule. Right: Weights of the connections related to the Ia sensory organs. Left: Weights of the connections related to the II sensory organs

As a result of the twitching experiment, the inhibitory and excitatory connections of spinal reflex circuit are found by hebbian learning rule due to the intrinsic statistical properties of the sensory signals. As it can be seen in Figure 4.5, an alpha motor neuron receives inhibitory connections not only from Ia afferent sensory signal of the muscle it activates but also from synergistic muscles, as well as inhibitory connections from II afferent signals. After training spinal reflex circuit model with twitching experiment, we managed to assign the weights of all these connections. Similar results are also obtained for II afferent sensory signals.

Figure 4.8 and Figure 4.9 show the identification of the synergistic and antagonistic muscles for Ia and II afferent sensory signals after training. The activity of the spinal interneurons due to the sensory signals shows that individual role of the muscles are encoded in the connections between sensory organs and spinal interneurons. Since both position and velocity profile of the muscles are driven by the same stimulus, the corresponding results show similarity. Covariance matrix of all possible pairs among spinal interneurons show that the distinction

between flexor and extensor muscles and also the identification of the synergistic muscles can be observed. While Long head biceps, short hand Biceps and Brachialis are identified as flexor muscles, long head Triceps, lateral head Triceps and medial head Triceps are assigned as extensor muscles. It also reveals the relative role of each muscles in movements, for instance there is high correlation between short hand Biceps and Brachialis.

At the end of the training with APEX learning rule, given in Eq. 4.11 and Eq.4.12, we obtained a weight distribution within the spinal reflex circuit which corresponds to the principal components of the input signal. Therefore, we managed to find out the individual contributions of each sensory information for the α-motoneuron activities. It is proven that APEX learning rule yields the principal components of the sensory signal with an orderly fashion. The proof of the APEX learning rule and its ability to obtain the principal components of the sensory signal can be found in (Haykin, 1994), pg. 446-451.

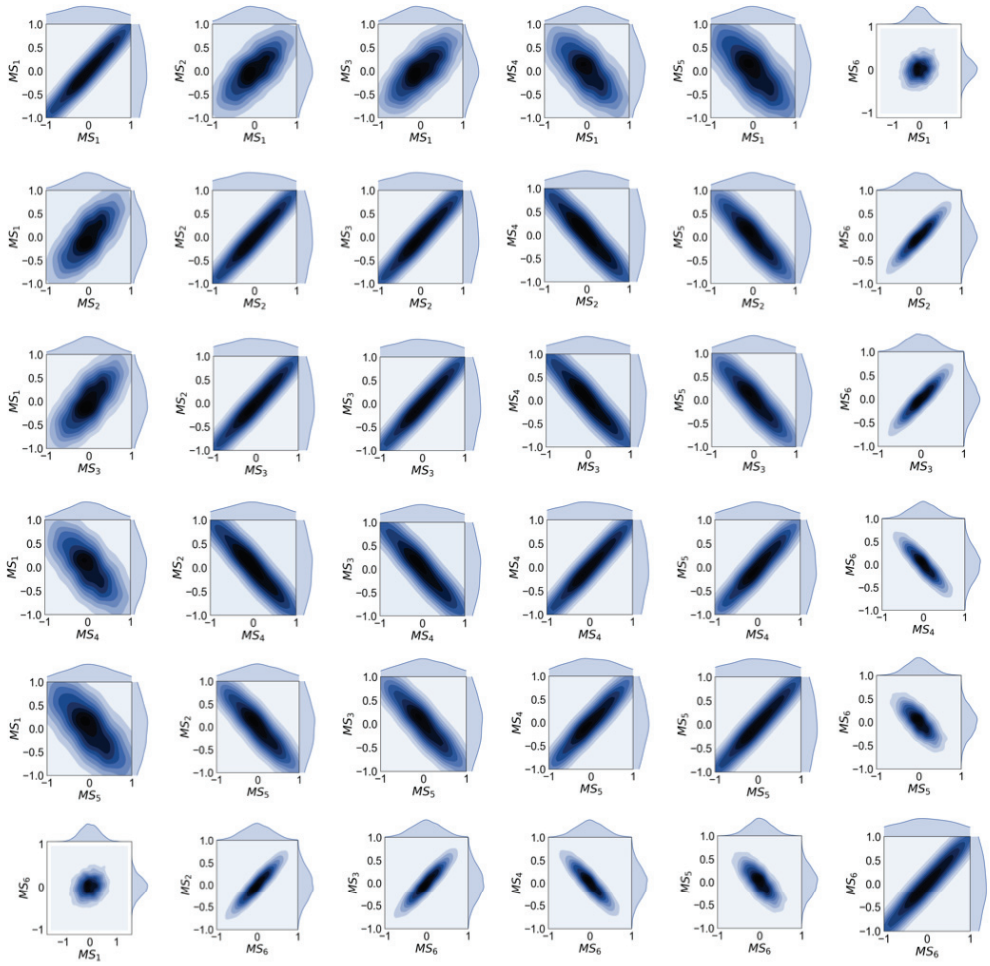


Figure 4.8: Pairwise covariance matrix of spinal interneuron activities that are connected to the Ia afferent sensory signals after the training of spinal reflex circuit with Hebbian Learning.

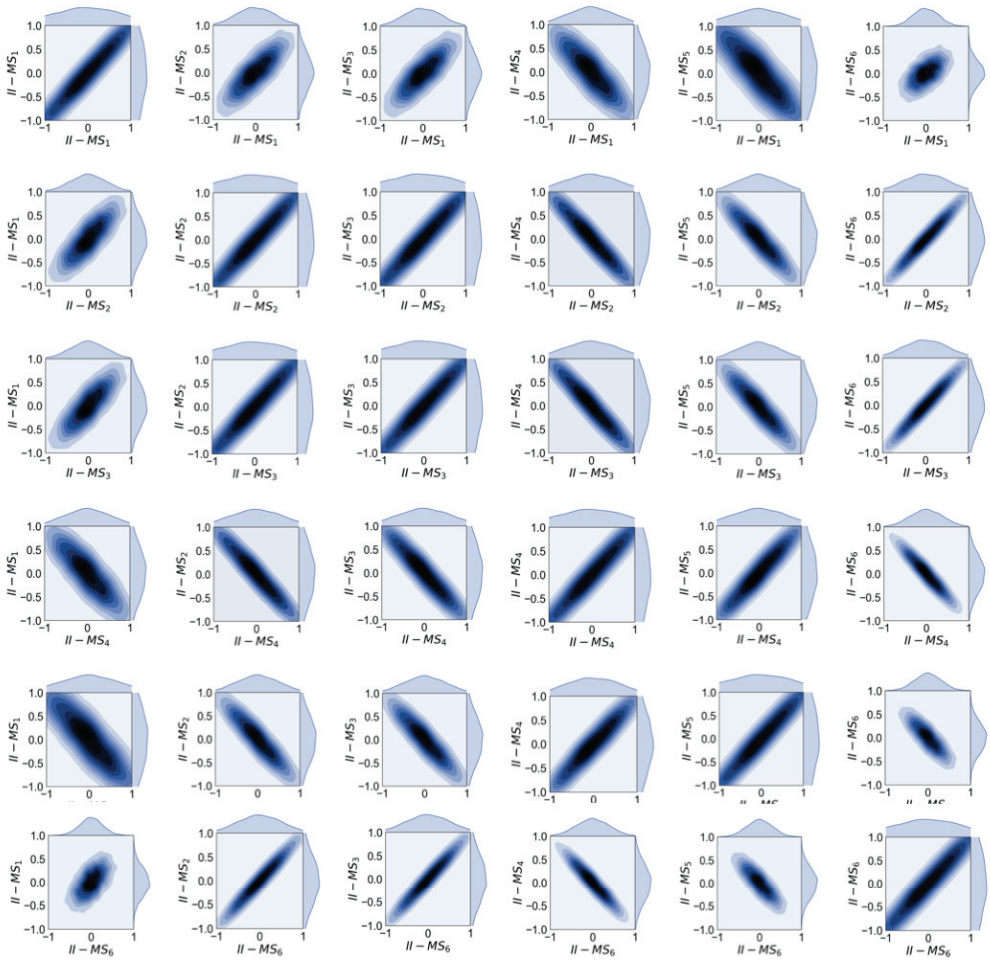


Figure 4.9: Pairwise covariance matrix of spinal interneuron activities that are connected to the II afferent sensory signals after the training of spinal reflex circuit with Hebbian Learning.

4.3 Closed-loop between cerebellum and spinal cord

Coordinated behavior requires timely decision of the movement trajectories in each time step. Depending on the position of the body, the transformation from one state of the position to the next one needs to be calculated in order to achieve the next position. Then the sensory information about the body position also need to be sent to the high-level motor control circuits for the adjustment of the motor commands. In each time step, information about the new position of the body is evaluated in order to create smooth movement trajectories. The cerebellum is thought to be involved in this precise timing of the movement generations.

Although the cerebellum only contributes approximately 10% of the brain volume, it comprises more than half of the total number of neurons in the brain. Highly ordered, repeated units of microcircuits builds up the structure of the cerebellum. The cerebellum does not only form a closed-loop with different brain regions and the spinal cord, but also distributes the activity from different brain regions to the different parts of the motor control circuits. Due to the similarity between the microcircuits of the cerebellum, one hypothesis about its function is that these microcircuits are responsible for the calculation of precise timing of motor actions. This hypothesis is supported by experimental studies in animals and human (Braitenberg et al., 1997). Patients or animals with a lesion in the cerebellum share common symptoms such as having difficulties of obtaining the desired control of movement (Bastian et al., 1996).

The cerebellum is divided by three sub-regions with respect to the difference of their involvements in different motor tasks; vestibulocerebellum, cerebrocerebellum and spinocerebellum. The vestibulocerebellum receives inputs from the vestibular and visual systems and sends information to the vestibular nuclei of the brainstem. Its main role is being involved in the evaluation of the balance of the body but it also takes part in vestibular reflexes and eye movements. Whereas cerebrocerebellum makes the closed-loop circuit between cerebellum and the cerebral cortex in order to be part of the planning and execution of the movement. The other closed-loop circuit is the spinocerebellum where the somatosensory and proprioceptive input of the spinal cord is evaluated and this computation is transmitted to the spinal cord to be able to maintain the control of the muscles. It takes main roles in the posture control, proximal and distal muscle control and also locomotion. The connections between spinocerebellum and spinal cord is the one which is focused in this study.

The complementary hypothesis of precise timing calculation and the functional role of the

cerebellum is that the "internal models" of the body is represented in the cerebellum. The idea of the internal models is that the repertoire of external world and the representations of the body is embedded in cerebellum. Given the internal models, cerebellum can adjust the motor behavior in case of changing environmental conditions or the motor control units through the learning capabilities of cerebellum. The internal model represents the structures of the motor control units which includes the information about the dynamic model of the limbs and kinematic properties. In another interpretation, cerebellum compares the desired and actual motor movements while considering the current internal model and adapts the internal model if there is mismatch. With the internal models, a timely and adjusted muscle contractions can be modulated due to fact that the model of the system dynamics could be used to calculate the desired end point of the movement. In addition, knowing the kinematic model of the joints with the endpoint goal can provide the sequence of motor commands to obtain desired joint position trajectories and muscle contractions. Experimental studies support the idea that kinematics and system dynamics is evaluated in the cerebellum functionality. It is shown that the governing dynamic properties of the multi-body systems is represented in the cerebellum. The importance of the internal multi-body dynamics representation can be observed in the arm movement control. Given that a movement that is generated in the elbow joint causes counter movement in the shoulder joint due to the dynamics of the arm, neural controller needs to come up with the evaluation of the counter movements to create precise movement trajectories. Thus, in order to achieve coordinated movement both in the elbow and shoulder joint, muscles located in the upper arm are needed to be controlled in a way to compensate this counter moment generation which stabilizes the both joint trajectories. Patients with cerebellar disorders show the lack of these stabilizing muscle coordination which could be interpreted as the internal model representation of the cerebellum.

4.3.1 High-level descending commands

We extended the equation of alpha motor neuron activities in a way that it incorporates the effect of modulatory descending signal of cerebellum as follows;

$$m_i = \sum_{j=1}^{n} W^I_{Ia,j} \left(\sum_{i=1}^{n} \left(W^E_{Ia,j} S_{Ia,i} \right) \right) + \sum_{j=1}^{n} W^I_{II,j} \left(\sum_{i=1}^{n} \left(W^E_{II,j} S_{II,i} \right) \right) + d_i(t) \tag{4.13}$$

where $d_i(t)$ denotes the timely descending commands to corresponding alpha motor neurons. In order to obtain these descending commands, we designed a babbling experiment and train

a feedforward neural network with data that is obtained during these experiments.

4.3.2 Babbling Experiment

The idea behind babbling experiments is to simulate α-motoneurons with stimulation selected from the arm control experiments that are explained in Chapter 3. We used a feedforward neural network to find relationship between descending commands and their contribution to the α-motoneuron activities. After training, the relationship between the descending commands and corresponding joint positions are encoded within the feedforward neural network. The range of the descending commands are selected withing the boundary of possible joint positions. Since there exists six muscles in musculoskeletal arm model, the output of the feedforward neural network has six neurons which have excitatory connections to alpha motor neurons. The input layer of the feedforward neural network has two input neurons which receive the desired goal position of the joints. The network has 2 hidden layers each with 64 neurons. The connections within the feedforward network is selected all-to-all. The schema of the feedforward neural network is shown in Figure 4.10.

The formulation of the alpha motor neuron activities resembles the PD-controller. The difference between the desired state information encoded in cerebellum's internal model and actual sensory knowledge is used as a stimulus to be given to the muscle dynamics whereas in PD-control, the control signal to be exerted on actuator, u, is calculated as follows;

$$u = P\left(s_d(t) - s_c(t)\right) + D\left(\dot{s}_d(t) - \dot{s}_c(t)\right) \tag{4.14}$$

where P, D represents the coefficients, $s_d(t), s_c(t)$ denotes the desired and actual state information. The main difference between a PD-controller and the activation of alpha motor neurons is that the state of the muscle is used as actual state information instead of state of the skeletal dynamics. The internal model of the feedforward neural network encodes the nonlinearity of this transformation intrinsically in the weights.

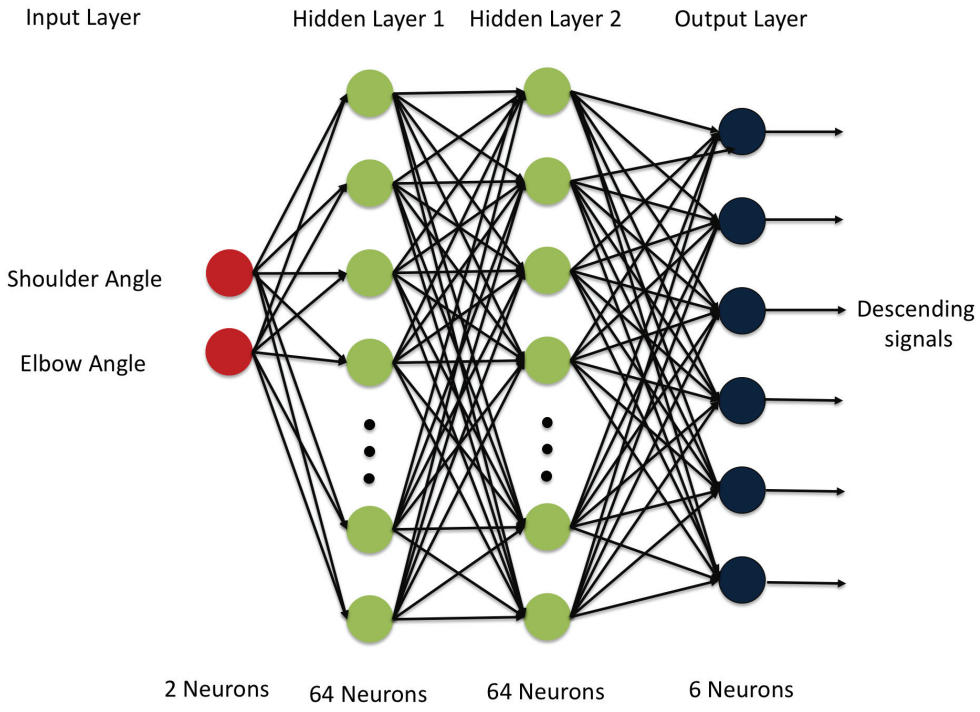


Figure 4.10: The feedforward neural network corresponds to cerebellum's internal model representation with 2 hidden layers each with 64 neurons

4.4 High-level commands between Cortical models and Cerebellum

To control the high-level descending commands from cerebellum model, we trained a recurrent neural network which provides information about the desired trajectory to be generated by the closed-loop between cerebellum and spinal-cord network. We aimed at integrating several optimal trjactories into same recurrent neural network, such that any of these integrated trajectories can be called and executed to control the cerebellum dynamics. However, it is known that a recurrent neural network will forget about the learned trajectory upon training the same network for another trajectory. This phenomena is called catastrophic interference or catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990). Although there exists studies to overcome this problem in recurrent neural networks (Kirkpatrick et al., 2017), we focused on an idea based on correlative learning which yields a possibility of biologically feasi-

ble implementation. Correlation based training for recurrent neural networks to overcome catastrophic forgetting is suggested by Jaeger (Jaeger, 2014) and it is called conceptors. Using conceptors allowed us to load different optimal trajectories into the same recurrent neural network.

We used a recurrent neural network that is composed of 500 neurons with random connections among them that are sampled from sparse random matrix with 10% density. Network has two input and two output neurons, one for elbow joint and one for shoulder joint. The input neurons are randomly connected to internal neurons and output neurons have connections from all internal neurons that are initially assigned with small random values. The recurrent network structure is given in Figure 4.11.

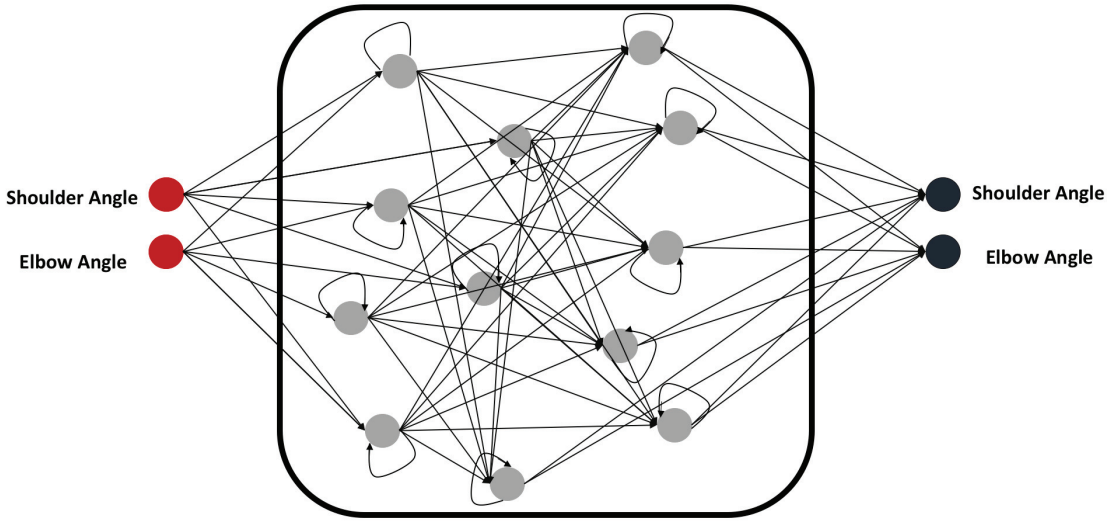


Figure 4.11: The recurrent neural network which stores optimal trajectories to be given to the cerebellum model has two input and two output neurons, for shoulder and elbow joints. After learning, network is capable of providing desired trajectory information to the closed-loop cerebellum-spinal cord network to be executed.

The activity of an internal neuron is calculated as follows;

$$x(n+1) = \tanh\left(W^* x(n) + W^{in} p(n+1) + b\right) \tag{4.15}$$

where x denotes level of activity of an internal neuron, W^* is initial internal weights, W^{in}

is weights between input and internal neurons, p represents input and b is called bias. The activity of output neurons is calculated as follows:

$$y(n) = W^{out} x(n) \tag{4.16}$$

here W^{out} represents the output weights that need to be learned. We trained the network with 16 trajectories (elbow and shoulder joints respectively) that are found with solution of different setup in optimal control formulation. These trajectories are used as input trajectories, $p^j(n)$ to be learned by recurrent neural network and resulted state of internal neurons are recorded $x^j(n)$ with 500 steps. Resulting state values comprises a matrix $X = \left[x^1|x^2|...|x^{16}\right]$, one time step shifted matrix $\tilde{X} = \left[\tilde{x}^1|\tilde{x}^2|...|\tilde{x}^{16}\right]$ along with input values $P = \left[p^1|p^2|...|p^{16}\right]$. Then the output weights are calculated with ridge regression formula:

$$W^{out} = \left(\left(X\tilde{X} + \sigma^{out} I_{NxN}\right)^{-1} XP'\right)' \tag{4.17}$$

where σ is a regularizer , 0.01. Then the conceptor matrix is given as follows:

$$C = R(R + \alpha^2 I)^{-1} \tag{4.18}$$

where R is the correlation matrix of each state values, $x^j(n)$ and α is called aperture. The idea behind the conceptor is to project the state trajectories into main directions in the sample data, similar to Principal Component Analysis. Once the output weights are trained, then internal weights is required to be calculated. Here the assumption is that after loading all input trajectories, the following equality has to be satisfied:

$$Wx^j(n) = W^* x^j(n) + W^{in} p^j(n+1) \tag{4.19}$$

for all input trajectories $p^j(n)$ which requires to minimize the squared error:

$$\epsilon(n+1) = \left(\left(\tanh^{-1}\left(x^j(n+1)\right) - b\right) - Wx^j(n)\right)^2 \tag{4.20}$$

therefore initial weights are then obtained with a ridge regression:

$$W = \left(\left(\tilde{X}\tilde{X}' + \sigma^W I_{NxN}\right)^{-1} \tilde{X}\left(\tanh^{-1}(X) - B\right)\right)' \tag{4.21}$$

where B represents a matrix whose columns are identical to b. To obtain one of the learned op-

timal trajectory in output neurons, we need to multiply the output activity with pre-calculated conceptor matrix such as:

$$y^j(n) = W^{out} C^j \tanh(W x(n-1) + b) \tag{4.22}$$

In Figure 4.12, four out of sixteen optimal trajectories that are loaded to the recurrent neural network are given as an example of loaded optimal trajectories. Right of the figure shows elbow trajectories whereas shoulder trajectories are given on the left of the figure.

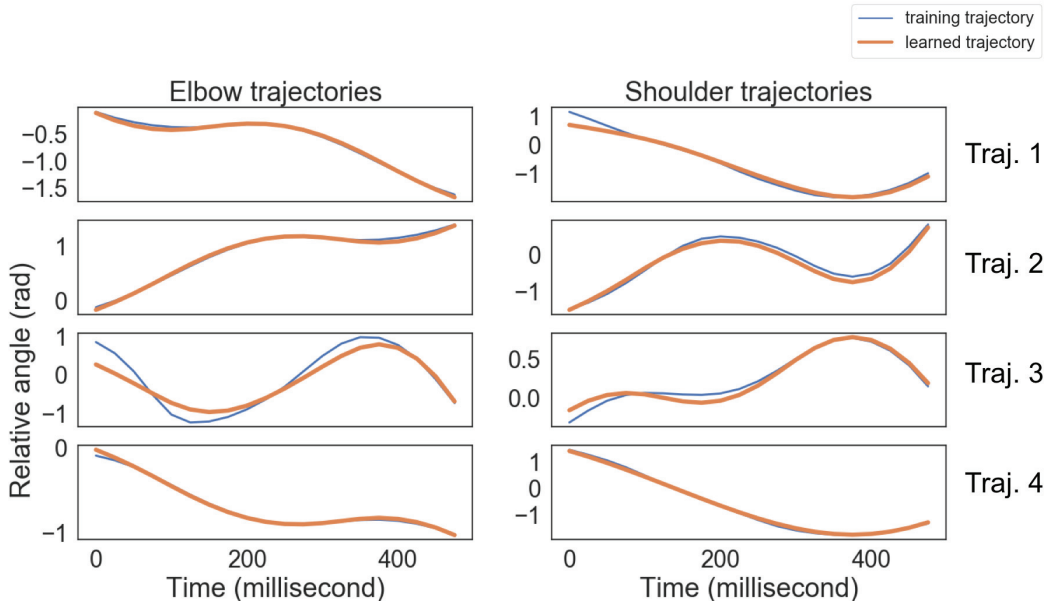


Figure 4.12: Examples of loaded optimal trajectories to the recurrent neural network. Four different trajectories for elbow (Left) and shoulder (right) joints are given. Thick orange line represents the learned whereas thin blue lines represent training trajectories in each figure.

4.4.1 Motor Learning

After training the spinal reflex circuit with twitching experiments and also the training of the recurrent neural network with optimum joint trajectories (see Chapter 3), we fixed all these weights in their corresponding networks and initiate the training of the feedforward neural network which is located in between these two circuits, see Figure 4.13. In order to train the feedforward network, we used the output activities of the actor-critic networks that are found with the arm control experiments (see Chapter 3, Figure 3.5, 3.6, 3.8, 3.10, 3.11) as well as

additional experiments that are not reported in this thesis albeit similar setup. We only used the first 0.5 seconds of the output of each actor-critic activity as a teaching signal to train the feedforward network. The goal of the training of the feedforward neural network is to obtain descending commands to α-motoneurons that would make the activity of the α-motoneurons identical to the output of the corresponding actor-critic network.

In Chapter 3, we showed that the problem of musculoskeletal control is an ill-posed problem, however the solution that we obtain with our learning and optimization framework (see Chapter 2) allows us to turn this ill-posed problem into a regularization problem where we can use the output of the actor-critic networks as of our teaching signal. Therefore, a feedforward neural network can be trained with a back-propagation algorithm since there exists 6 teaching signal for 6 α-motoneurons. To train the feedforward neural network, we calculate an error signal which is the difference between the activity of α-motoneurons and the activity of the output of the actor-critic networks in which the output of each actor-critic network is considered as the teaching signal. At the beginning of the trainning, the activity of the α-motoneurons are driven by the reflex circuit and untrained feedforward neural network which causes the difference between the desired activity and teaching signal. Then we used this error signal to train the output of the feedforward neural network with back-propagation, for details of the algorithm see (Hecht-Nielsen, 1992; Haykin, 1994). In Figure 4.13, the connections that are subject to learning are indicated with red color and the rest of the connections remain fixed during the training phase. After the training of the feedforward neural network, depending on the desired movement, high-level circuits stimulates the intermediate feedforward network with desired joint trajectories of the elbow and shoulder, see Figure 4.13. Based on this desired movement, then the feedforward neural network turns this joint positions into descending commands to α-motoneurons. After all, movement is obtained with the combined stimulus (see Eq. 4.13) of the descending commands of the feedforward neural network and the activity of the spinal reflex circuit where the sensory information is encoded.

To assess the robustness of the model, we examined the movement control with a reaching experiment. We assigned eight different goal positions, given by the specific joint angles in space. For each goal position, we run the simulations five times and final positions of the musculoskeletal arm were recorded. The simulation results of this experiment is shown in Figure 4.14. The goal positions for elbow and shoulder joints are $[0,\pi/8]$, $[0,\pi/4]$, $[\pi/8,\pi/8]$, $[\pi/8,\pi/4]$, $[\pi/4,\pi/4]$, $[-\pi/8,0]$, $[-\pi/8,\pi/8]$, $[-\pi/8,\pi/4]$.

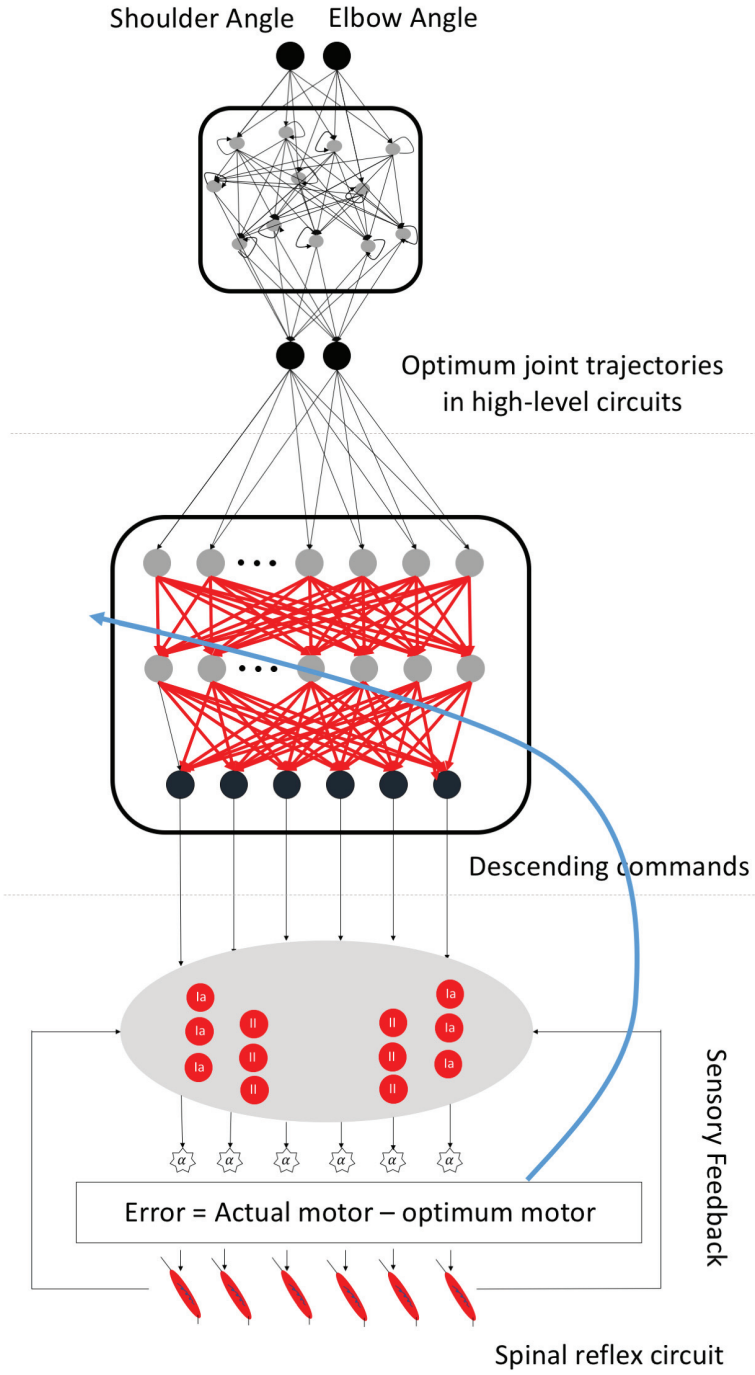


Figure 4.13: Proposed motor control model. A desired signal is given to the high-level circuits which is modelled as a recurrent neural network and trained with correlative learning. Then the desired descending commands are generated by the intermediate feedforward neural network. Along with these descending commands, the reflex circuit that integrate the sensory information generate the stimulus given to the α-motoneurons which in turn generate the movement. Only weights that are indicated with red color are subject to training. The error signal is given with a blue arrow.

From the initial position of the musculoskeletal arm to eight equidistant targets are tested in this experiment. In each experiment, the musculoskeletal arm reached a region rather than exact target location. The region of error is approximately sketched with a red rectangle in which the maximum target error was obtained 0.06 radian.

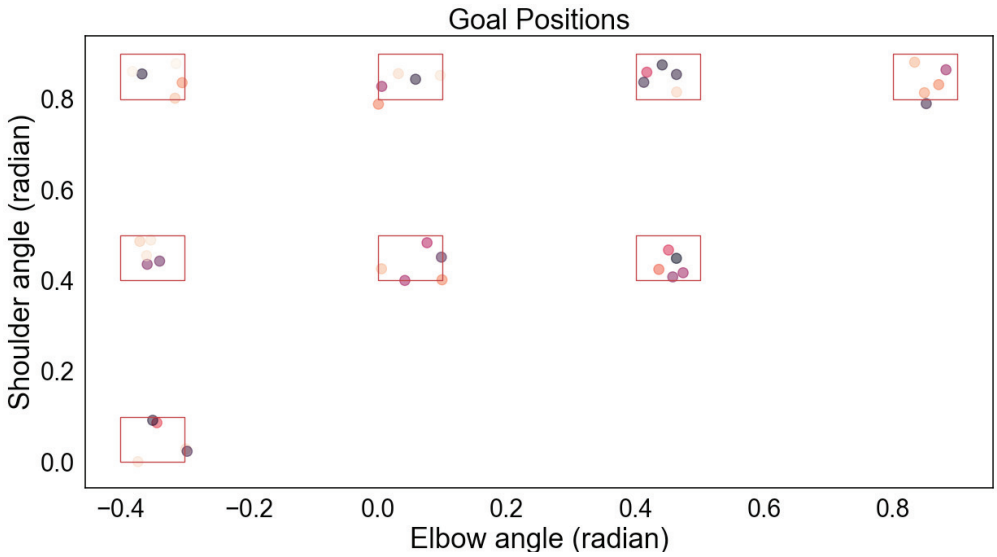


Figure 4.14: In order to test the ability of the model, eight different positions in space are defined as goal positions. The goal positions are given with red box while the dots represent the angles of the elbow and shoulder joints at the end of the simulations. Different colors are used to indicate the joint angles at the end of the simulations.

To test the ability of the proposed motor control system to generalize the movement control, we tested the model with three movements that were not part of the training data. We used the network that has been trained with optimum solutions of the learning and optimization framework, the details of the network can be seen in Figure 4.13. Since the problem has been turned into a well-defined regression problem, the trained network has the ability to generate a movement trajectory using the previously trained weight distributions. The purpose of the training was to create a map between joint angles and the necessary descending commands to generate those joint angles. Therefore after learning, network has the ability of transforming the joint angles to descending commands. Hence, in order to generate a movement that was not part of the learning, network associates the previously learned relationships between joint angles and descending commands.

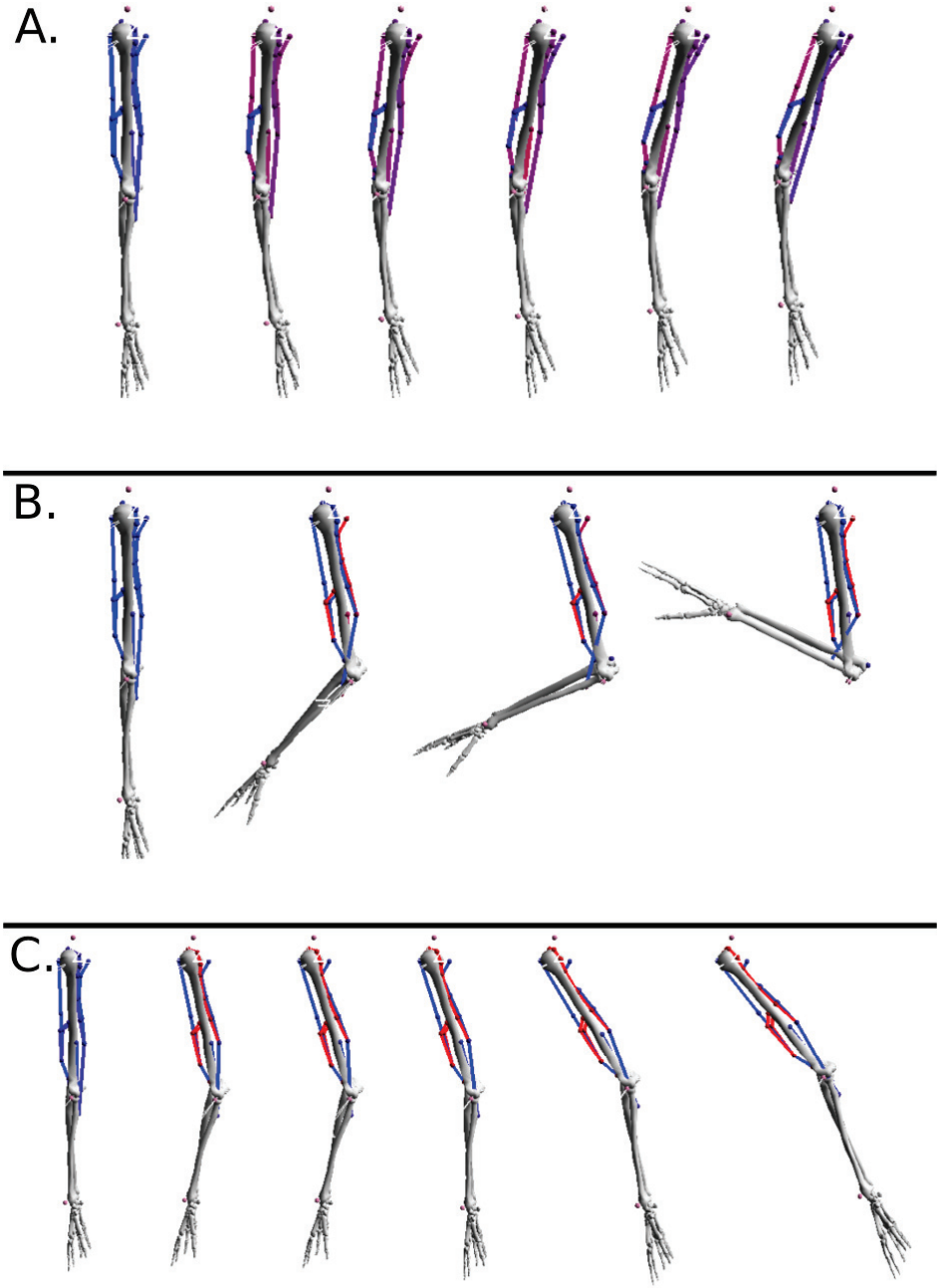


Figure 4.15: Examples of three trajectory generation with human arm model. The initial position of the arm is vertical. The movements are given with the snapshots of the trajectories. A. The desired trajectory is chosen such that a flexor behavior is required. B. The musculoskeletal arm is required to rise above the elbow joint in order to achieve the reaching goal position. C. A backward movement of arm is also obtained.

In order to show the capability of the network, we used trajectories that were not included in the training data to derive the feedforward neural network. The objective is to show that the trained feedforward neural network is not only capable of generating training trajectories but also generate a trajectory that was not a part of the training data. Figure 4.15 shows three distinct arm movements for given desired angles. In each experiments, an arbitrarily defined angle trajectory is given to the feedforward neural network which in turn translates these positions into descending commands. Consequently, descending signals drive the spinal reflex model in order to obtain the desired trajectories in the arm control.

4.5 Conclusion

Thus far, we have presented an abstract model of neural motor control system in which we investigated the one of the movement hypothesis, called OCT. We mainly focused on three subsystems of the neural motor control, to be specific, high-level motor execution with a recurrent neural network where we used correlative learning, internal model representation of the cerebellum with a feedforward neural network and spinal reflex circuit in which we used an unsupervised Hebbian type learning algorithm. We first obtained the reference trajectories with proposed learning and optimization framework where we used OCT and reinforcement learning. After that we used these reference trajectories to train the neural motor control model.

The identification of the weights of the spinal reflex network was based on an experiment, called twicthing. We then train the spinal reflex circuit with APEX learning rule in order to identify the functional role of the sensory information. After that we integrated the reference joint trajectories into a recurrent neural network with a correlative learning algorithm which is known as conceptor learning. It allowed us to solve the catastrophic forgetting phenomena since we aimed at memorizing multiple reference joint trajectories. To train the feedforward neural network in order to integrate the solutions of the learning and optimization framework. During this training, we fixed all the previously learned weights of the high-level and spinal network while training the feedforward neural network with the output of the actor-critic networks that we obtained for the reference trajectories. We showed that the reference trajectories are learned by the abstract model of the motor control system and we used the model to control a human arm model with 6 muscles and 2 joints.

5 Conclusion and Discussion

5.1 Conclusion

Throughout this thesis, we have examined the functional properties of the human neural motor control system. In Chapter 2, we presented a learning and optimization framework to obtain a desired movement trajectory for a musculoskeletal system. Chapter 3, we presented the experiments with a model of a human arm control as well as human locomotion along with the corresponding results while using our learning and optimization framework. We showed how a variety of different reference trajectories can be obtained with optimal control and how these reference trajectories are mapped to the musculoskeletal control with reinforcement learning.

Based on the results of these experiments, we then focused on the development of a neural motor control system for a human arm control, which we then presented in Chapter 4 with corresponding results. We showed that after training, the proposed neural motor control model is able to generate movement trajectories for randomly determined goal positions in space.

In the introduction of this thesis, we posed four questions to be studied throughout this thesis:

1. Can we solve the redundancy in musculoskeletal control problems, in such a way that the solution is optimum?

2. What are the principle roles of reflex circuits in spinal cord, how can we obtain these circuits with learning rules that have biological relevance?

3. How can we integrate proprioception signals intio motor control circuit to create a closed-loop system?

4. Can we use the idea of reverse engineering to model the neural motor control system for movement generations?

In the following, we discuss our results in the context of these questions.

5.1.1 Reformulating of RL provides a solution to redundancy in musculoskeletal systems

Our experiments show that reformulating the reward function of reinforcement learning provide a solution of mapping the optimal trajectories to the muscle control. Instead of using a global reward function that determines the desired goal position, the reward function is written as a minimization of the difference between a reference trajectory and the current state positions. This enables us not only to obtain the goal position but also to control the joints in each time step. As a result of this precise control, we have managed to solve the ill-posed problem of muscle control where the activation of multiple antagonistic muscles results in a one dimensional joint trajectory. We have also obtained the time coarse of activation for the alpha motor neurons for each muscle. A similar idea has been gaining attention in the computer-animation where hand-crafted sequences of a musculoskeletal system are used as reference motions to be learned with reinforcement learning (Coros et al., 2011; Geijtenbeek et al., 2013; Peng et al., 2017). The caveat of this approach is that reference motions need to be created for all frames in a sequence. In addition, contrary to the solutions of reinforcement learning where the learned trajectory is one of many possible trajectories, we have shown that an exact trajectory can be obtained with our approach. We showed that the error range of the learned trajectories is approximately around 0.05 which indicates the ability of the controller to track the desired trajectory. As can be seen in one of the recent implementation by Heess et.al. (Heess et al., 2017), the movements that have been found by reinforcement learning can only achieve forward movement while controlling the joints in an obscure way. However, by using the trajectories from optimal control as a reference signal for reinforcement learning implementation has allowed us to acquire optimum and human like behavior of the musculoskeletal system.

It can be said that OCT not only satisfies the optimality condition but also consider the dynamics of the system itself and therefore provides physically realistic reference trajectories. The other advantage of using OCT is that all feasible movement trajectories can be obtained with reformulating the objective function and the related state and path constraints.

5.1.2 Self-organizing learning rule and reflex circuits

After we have prepared the learning and optimization framework, we have started to work on the modelling of the neural motor control system. First, we have focused on the structure of the spinal cord circuits where proprioception signals are integrated. We have used the idea of twitching and motor babbling in the musculoskeletal system to obtain data to be used in self-organizing learning. Our reflex circuit model and also the experimental design is similar to studies in (Marques et al., 2014). The learning rule, (called anti-Oja rule) has been suggested by (Marques et al., 2014) is a variation of Oja's rule (Oja, 1982). It was shown that Oja's rule detects the first principal component of data (Oja, 1982), however our goal was not only to find all principal components of the data, but also to find them in the right order. We used an Hebbian-type learning rule, Kung and Diamantaras (Kung and Diamantaras, 1990), to decompose complex twitching signals into motor basis functions (principle components) that are then learned by the connections within spinal reflex circuit. It is proven that learning rule developed by Kung and Diamantaras (Kung and Diamantaras, 1990) developed a learning rule that yields the connectivity between excitatory and inhibitory neurons along with ordered principle components of the data. Using this rule, we have then demonstrated that it is possible to obtain the correlated activity among muscles, such that the roles of antagonistic muscles and correlated activity among corresponding muscle can be separated.

The spinal reflex circuit that we have obtained with the Kung and Diamantaras rule allowed us to integrate the statistical properties of the proprioception signals into the circuit and also to identify the connectivity within the circuit. The activity of the alpha motor neurons also allowed us to transmit the "muscle coordinates", represented by sensory signals, into joint coordinates. We have also managed to modulate the alpha motor neuron activity with descending signals based on real-time sensory information.

5.1.3 Closed-loop motor control

Sherrington (Sherrington, 1952) defines proprioception as a stimulus that represents the body itself. Proprioception signals originate in sensory organs, such as muscle spindles, golgi-tendon organs etc, and provide sensation of the relative position of body and limbs. They also encode limb velocity, muscle tension and force. This information is used by spinal cord circuits for the real-time control of movements. Moreover, reflexes depend almost exclusively on proprioception signals. Based on this knowledge, we have integrated sensory information in our spinal cord circuit model and our closed-loop control architecture for muscle control. Proprioceptive information is not only used as feedback signal but also for the coordinate transformation from muscle coordinates to joint coordinates, that are evaluated in our cerebellum model.

Apart from detecting the changes of body orientation and position, proprioception signals carry information about relative differences between the state of muscles. We have designed an experimental setup (twitching and motor babbling) to exploit this information in order to train the connectivity pattern within the spinal cord circuitry, as mentioned above. With the integration of the statistical properties of proprioception signals into our spinal cord model, we have demonstrated that the individual contribution of muscles for a specific movement is encoded in the connectivity.

5.1.4 Neural motor control model

In the second part of the thesis, we have shown that the integration of reference trajectories to the neural motor control model provided a promising approach to study the functional roles of the different sub-units of the motor control system. Integrating optimal trajectories into the model allows us to study the modulatory control of descending signals. Although we have achieved a coherent control scheme for the musculoskeletal system, the complex architecture of the motor control could only be partially studied. Our approach represents an abstraction of the full biological system, since we have used rather abstract artificial neural network. However, we believe that understanding the global structure of the motor control system provides a useful constraint for detailed modelling approaches because the structure and connectivity of the abstract model could be transferred to detailed models, also maintaining the functionality of the different sub-units.

5.2 Discussion

From the point of views of robotics, there are two main methodologies to identify the controller of the joints: a metaheuristic optimization or model-based optimization methods. The advantage of model-free metaheuristic optimization methods is that one does not need to know the details of the robot's kinematic and dynamic to solve the control problem. Thanks to the recent improvements in computation power, one can solve a model-free optimization problem on a compute cluster and achieve promising results in acceptable time.

There has been several successful studies to solve the musculoskeletal control problem with metaheuristic optimization methods and evolutionary algorithms (Coros et al., 2011; Geijtenbeek et al., 2013; Dura-Bernal et al., 2017). In these studies, it has been shown that a solution of the musculoskeletal control can be obtained with these algorithms. However, these studies only focus on the solution of the control problem without biological concerns.

In addition, the solutions are restricted to the mechanical properties of the system, such that a change in the musculoskeletal system requires a training from the beginning since these methods don't incorporate the dynamics of the model into the optimization procedure. However, we focused on one of the motor control hypotheses, OCT, to address the biological relevance of the movement trajectories as well as the methodology of the OCT allowed us to incorporate the system dynamics of the musculoskeletal system in which we showed the motor control model has the ability to generalize the movement trajectories while satisfying the minimum energy consumption.

In particular for locomotion, we depend on a high accuracy of the solutions, in order to obtain realistic movements. For the human arm model we could fully demonstrate the potential of our approach, however for the locomotion problem, we could so far only find a partial solution. To limit the initial complexity of the human locomotion (7-linked biped robot), we have omitted several important aspects of the system dynamics, such as the ground-reaction force, asymmetric and non-periodic solutions of locomotion therefore obtained only a partial solution for the musculoskeletal locomotion control.

Underactuated biped locomotion has been an ongoing research area in which there has been different approaches studied, such as rule-based (Yin et al., 2007), learning and optimization (Han et al., 2014; Lee and Terzopoulos, 2006), human motion-captured data (Sok et al., 2007). In these studies, the main objective was to obtain a stable and torque-based controller to

generate human-like movements. The stochastic optimization methods accelerated the scaling of the musculoskeletal studies. In (Geijtenbeek et al., 2013), it has been shown that a stochastic optimization method has the ability of scaling up the problem of musculoskeletal control. Variety of different robot morphologies have been succesfully controlled with the help of a stochastic optimization method.

Recent developments in Reinforcement Learning and deep neural networks initiated an interest in musculoskeletal control problems (Peng and van de Panne, 2017; Peng et al., 2018; Lee et al., 2019; Kidziński et al., 2018). It has been shown that Reinforcement learning can also be applied to musculoskeletal systems due the availability of the computing powers. In this thesis, we also focused on Reinforcement Learning to address the musculoskeletal learning. However, to improve the quality of the learning and optimization framework, a comparison of these different methods has been left as future studies.

An abstract model of the neural motor control system allowed us to study the possible functional role of the different sub-units of the system. This approach is quite common and has been suggested in several studies, they have suggested particular functional roles of the sub-units of motor control and have tried to map their abstract models to more detailed ones (DeWolf et al., 2016; Eliasmith et al., 2012). However, our conclusion is that the complexity of mammalian motor control systems require a more comprehensive modelling approach. To study the concerted behavior of whole system we had to focus on the main functionality of the system and its sub-units. Consequently, we have focused on only the high-level descending commands, cerebellum's internal model and spinal cord reflex circuits among all the motor control system and there are still many uncovered sub-systems.

In this thesis, we wanted to show that a bottom approach can be used to understand the structure and functionality of the motor control system. We developed a learning and optimization framework to study the ill-posed musculoskeletal control problem and then we used this framework to establish the weight distribution of the motor control model that we proposed. The solutions that we obtained with optimal control and reinforcement learning allowed us to constraint the mathematical model of the motor control system.

The motivation behind this methodology is based on the fact there is no available tool to dissect the outcome of a long evolutionary and skill learning process that took place in motor control development. Therefore constraining the model with a well established motor control hypothesis, OCT, allowed us to study the motor control system rigorously. A complementary

study of this thesis is to test experimental data with the proposed motor control model and the learning framework since this comparison would pave the way to improve our understanding of the motor control system.

5.3 Outlook

The proposed learning and optimization framework for motor control can be adapted to different musculoskeletal control problems; by changing e.g. the morphological details of a limb model and adapting the system dynamics accordingly. It is therefore also possible to study different animals, since the framework only requires to write equations of motions its physiological constraints for the desired animal model. For example, Therefore, the results for the human musculoskeletal system can be modified to study not only mouse models but also another mammalian musculoskeletal control problem.

The accuracy of the solution of optimal control depends highly on how detailed the system dynamics are formulated. In order to improve the quality of the locomotion model, one has to extend the equation of motion of the planar robot model to include at least ground-reaction force. Another caveat of our implementation is that we trained a 3D musculoskeletal system on reference trajectories that were obtained from a 2D model.

To further investigate limb control or locomotion patterns, one could also use motion-capture data in addition to synthetic data to train the musculoskeletal system. Motion-capture data would not only allow us to replicate human movements but also allow us to compare the solutions of our optimal control framework to human reference data. Based on such a comparison, it would be possible to adjust the formulation of the optimal control framework as well as the system dynamics. After validating the solution of the optimal control formulation, we could then study the conditions for more advanced movement.

Apart from the fact that our approach depends on an abstract model of the neural motor control system, more comprehensive models of muscle path wrapping, more accurate models for proprioception signals and excluded sub-units of the motor control could be integrated into our framework. Finally, our abstract model of the motor system can be mapped onto network models that are based on biological data. This would allow us to improve the abstract model, by using the detailed model as reference. Such biologically constrained models depend in turn on accurate and detailed experimental data.

A Appendix

A.1 Euler-Lagrangian equation of 2D arm model

The Lagrangian mechanics is one of the methods to obtain the equations of motion for the behavior of a physical system. Solutions of a lagrangian mechanics provide the trajectory of a system. In summary, Lagrangian can be stated as a function that takes the generalized coordinates of the system and their time derivatives to describe the motion of system. We used Euler-Lagrangian equation to write down the equations of motion for 2D arm model which resembles the double pendulum dynamics. The arm model in Figure 3.1 is a two-link planar system with point mass in the middle of each link. An external torque that is exerted in shoulder joint, u_1, and in elbow joint, u_2, control the rotational movement of each link. The 2D arm model has four state variables, position of shoulder and elbow, θ_1, θ_2, along with velocities of these joints, $\dot{\theta}_1, \dot{\theta}_2$. The illustration of 2D arm model is given in Figure A.1

The cartesian coordinates of each link X_1, Y_1, X_2, Y_2 can be written as follows:

$$\begin{aligned} X_1 &= -\frac{1}{2} l_1 \sin(\theta_1) \\ Y_1 &= \frac{1}{2} l_1 \cos(\theta_1) \\ X_2 &= -l_1 \sin(\theta_1) - \frac{1}{2} l_2 \sin(\theta_2) \\ Y_1 &= ł_1 \cos(\theta_1) + \frac{1}{2} l_2 \cos(\theta_2) \end{aligned} \tag{A.1}$$

and the velocities are;

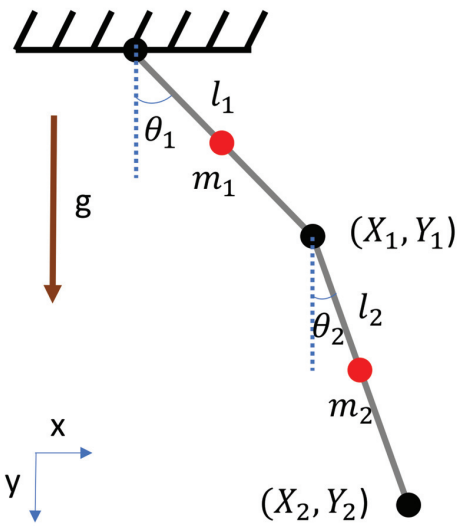


Figure A.1: The sketch of 2D arm model

$$\begin{aligned}
\dot{X}_1 &= -\frac{1}{2}\dot{\theta}_1 l_1 \sin(\theta_1) \\
\dot{Y}_1 &= \frac{1}{2}\dot{\theta}_1 l_1 \cos(\theta_1) \\
\dot{X}_2 &= -l_1\dot{\theta}_1 \sin(\theta_1) - \frac{1}{2} l_2 \dot{\theta}_2 \sin(\theta_2) \\
\dot{Y}_1 &= ł_1\dot{\theta}_1 \cos(\theta_1) + \frac{1}{2} l_2 \dot{\theta}_2 \cos(\theta_2)
\end{aligned} \tag{A.2}$$

therefore the squared velocities of each link at the midpoint is:

$$v_1^2 = \dot{X}_1^2 + \dot{Y}_1^2 = \frac{1}{4} l_1^2 \dot{\theta}_1^2 \tag{A.3}$$

$$v_2^2 = \dot{X}_1^2 + \dot{Y}_1^2 = l_1^2\dot{\theta}_1^2 + \frac{1}{4} l_2^2 dot\theta_2^2 + l_1 l_2 \dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) \tag{A.4}$$

The Lagrangian equation is defined as difference between kinetic, T, and potential energy, P which are given as follows:

$$\begin{aligned}
L &= T - V \\
&= \frac{1}{2} m_1^2 v_1^2 + \frac{1}{2} m_2^2 v_2^2 + \frac{1}{2} I_1^2 \dot{\theta}_1^2 + \frac{1}{2} I_2^2 \dot{\theta}_2^2 - m_1 g Y_1 - m_2 g Y_2
\end{aligned}$$

where $I_1 = \frac{1}{12} m_1 l_1^2$, $I_2 = \frac{1}{12} m_2 l_2^2$ are the angular moment of inertia for each joint. Inserting the velocities, A.3 and A.4 into Lagrangian equation;

$$L = \frac{1}{8} m_1 l_1^2 \dot{\theta_1}^2 + \frac{1}{2} m_2 (l_1^2 \dot{\theta_1}^2 + \frac{1}{4} l_2^2 \dot{\theta_2}^2 + l_1 l_2 \dot{\theta_1} \dot{\theta_2} \cos(\theta_1 - \theta_2)) + \frac{1}{2} I_1 \dot{\theta_1}^2 + \frac{1}{2} I_2 \dot{\theta_2}^2 - \frac{1}{2} m_1 g l_1 \cos(\theta_1) - m_2 g (l_1 \cos(\theta_1) + \frac{1}{2} l_2 \cos(\theta_2)) \tag{A.5}$$

The definition of the equation of motion in Lagrangian equation is given as follows:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta_i}} - \frac{\partial L}{\partial \dot{\theta_i}} = Q_i \tag{A.6}$$

where Q_i denotes external forces and q_i denotes the state variables. Therefore we need to write down all the partial derivatives with respect to state variables:

$$\frac{\partial L}{\partial \dot{\theta_1}} = l_1^2 \dot{\theta_1} (\frac{1}{4} m_1 + m_2) + \frac{1}{2} m_1 l_1 l_2 \dot{\theta_2} \cos(\theta_1 - \theta_2) + I_1 \dot{\theta_1} \tag{A.7}$$

$$\frac{\partial L}{\partial \theta_1} = -\frac{1}{2} m_2 l_1 l_2 \dot{\theta_1} \dot{\theta_2} \sin(\theta_1 - \theta_2) + (\frac{1}{2} m_1 + m_2) g l_1 \sin(\theta_1) \tag{A.8}$$

$$\frac{\partial L}{\partial \dot{\theta_2}} = m_2 l_2 (\frac{1}{4} m_2 \dot{\theta_2} + \frac{1}{2} l_1 \dot{\theta_1} \cos(\theta_1 - \theta_2)) + I_2 \dot{\theta_2} \tag{A.9}$$

$$\frac{\partial L}{\partial \theta_2} = \frac{1}{2} m_2 l_2 (l_1 \dot{\theta_1} \dot{\theta_2} \sin(\theta_1 - \theta_2) g \sin(\theta_2)) \tag{A.10}$$

then the equation of motion for 2D arm model can be written as follows:

$$\begin{aligned} u_1 &= \ddot{\theta_1} \left(l_1^2 (0.25 m_1 + m2) + I_1 \right) + \ddot{\theta_2} 0.25 m_2 l_1 \cos(\theta_1 - \theta_2) \\ &\quad + l_1 \left(0.5 m_2 l_2 \dot{\theta_2}^2 \sin(\theta_1 - \theta_2) - g \sin(\theta_1) (0.5 m_1 + m_2) \right) \\ u_2 &= \ddot{\theta_1} 0.5 l_1 l_2 m_2 \cos(\theta_1 - \theta_2) + \ddot{\theta_2} \left(0.25 m_2 l_2^2 + I_2 \right) \\ &\quad - 0.5 m_2 l_2 \left(l_1 \dot{\theta_1}^2 \sin(\theta_1 - \theta_2) - g \sin(\theta_2) \right) \end{aligned} \tag{A.11}$$

Bibliography

Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8, 2007.

Giovanni Abbruzzese and Alfredo Berardelli. Sensorimotor integration in movement disorders. *Movement disorders*, 18(3):231–240, 2003.

Marko Ackermann and Werner Schiehlen. Dynamic analysis of human gait disorder and metabolical cost estimation. *Archive of Applied Mechanics*, 75(10-12):569–594, 2006.

ED Adrian. Afferent areas in the cerebellum connected with the limbs. *Brain*, 66(4):289–315, 1943.

Garrett E Alexander and Michael D Crutcher. Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in neurosciences*, 13(7):266–271, 1990.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 146–155. JMLR. org, 2017.

Richard A Andersen and Christopher A Buneo. Intentional maps in posterior parietal cortex. *Annual review of neuroscience*, 25(1):189–220, 2002.

Frank C Anderson and Marcus G Pandy. Dynamic optimization of human walking. *Journal of biomechanical engineering*, 123(5):381–390, 2001.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

Rika Antonova, Akshara Rai, and Christopher G Atkeson. Sample efficient optimization for learning controllers for bipedal locomotion. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 22–28. IEEE, 2016.

Hj Asanuma and I Rosen. Topographical organization of cortical efferent zones projecting to distal forelimb muscles in the monkey. *Experimental brain research*, 14(3):243–256, 1972.

David G Asatryan. Functional tuning of the nervous system with control of movement or maintenance of a steady posture. 1. mechanographic analysis of the work of the joint on execution of a postural task. *Biophysics*, 10:925–935, 1965.

James Ashe and Apostolos P Georgopoulos. Movement parameters and neural activity in motor cortex and area 5. *Cerebral Cortex*, 4(6):590–600, 1994.

Amy J Bastian, TA Martin, JG Keating, and WT Thach. Cerebellar ataxia: abnormal control of interaction torques across multiple joints. *Journal of Neurophysiology*, 76(1):492–509, 1996.

Amy J Bastian, KM Zackowski, and WT Thach. Cerebellar ataxia: torque deficiency or torque mismatch between joints? *Journal of neurophysiology*, 83(5):3019–3030, 2000.

Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience*, volume 2. Lippincott Williams & Wilkins, 2007.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Richard Ernest Bellman. *Dynamic Programming*. Courier Dover Publications, 1957.

Nikolai Bernstein. The co-ordination and regulation of movements. *The co-ordination and regulation of movements*, 1966.

Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific, 1995.

Armin Biess, Dario G Liebermann, and Tamar Flash. A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *Journal of Neuroscience*, 27(48): 13045–13064, 2007.

Mathew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in cognitive sciences*, 2019.

Valentino Braitenberg, Detlef Heck, and Fahad Sultan. The detection and generation of sequences as a key to cerebellar function: experiments and theory. *Behavioral and Brain Sciences*, 20(2):229–245, 1997.

T Graham Brown. On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system. *The Journal of physiology*, 48(1):18–46, 1914.

Thomas Graham Brown. The intrinsic factors in the act of progression in the mammal. *Proceedings of the Royal Society of London. Series B, containing papers of a biological character*, 84(572):308–319, 1911.

Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. Routledge, 2018.

Christopher A Buneo and Richard A Andersen. The posterior parietal cortex: sensorimotor interface for the planning and online control of visually guided movements. *Neuropsychologia*, 44(13):2594–2606, 2006.

Richard R Carrillo, Eduardo Ros, Christian Boucheny, and J-MD Coenen Olivier. A real-time spiking cerebellum model for learning robot control. *Biosystems*, 94(1-2):18–27, 2008.

Claudia Casellato, Alberto Antonietti, Jesus A Garrido, Richard R Carrillo, Niceto R Luque, Eduardo Ros, Alessandra Pedrocchi, and Egidio D'Angelo. Adaptive robotic control driven by a versatile spiking cerebellar network. *PLoS one*, 9(11):e112265, 2014.

Umberto Castiello. The neuroscience of grasping. *Nature Reviews Neuroscience*, 6(9):726, 2005.

Ettore Cavallaro, Jacob Rosen, Joel C Perry, Stephen Burns, and Blake Hannaford. Hill-based model as a myoprocessor for a neural controlled powered exoskeleton arm-parameters optimization. In *Proceedings of the 2005 IEEE international Conference on Robotics and Automation*, pages 4514–4519. IEEE, 2005.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Stephen I Ryu, and Krishna V Shenoy. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68(3):387–400, 2010.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51, 2012.

Paul Cisek and John F Kalaska. Neural mechanisms for interacting with a world full of action choices. *Annual review of neuroscience*, 33:269–298, 2010.

Yale E Cohen and Richard A Andersen. A common reference frame for movement plans in the posterior parietal cortex. *Nature Reviews Neuroscience*, 3(7):553, 2002.

Carol L Colby and Michael E Goldberg. Space and attention in parietal cortex. *Annual review of neuroscience*, 22(1):319–349, 1999.

JD Cooke and Virginia A Diggles. Rapid error correction during human arm movements: evidence for central monitoring. *Journal of motor behavior*, 16(4):348–363, 1984.

Stelian Coros, Andrej Karpathy, Ben Jones, Lionel Reveret, and Michiel Van De Panne. Locomotion skills for simulated quadrupeds. In *ACM Transactions on Graphics (TOG)*, volume 30, page 59. ACM, 2011.

Kenneth James Williams Craik. *The nature of explanation*, volume 445. CUP Archive, 1952.

Donald J Crammond and John F Kalaska. Prior information in motor and premotor cortex: activity during the delay period and effect on pre-movement activity. *Journal of neurophysiology*, 84(2):986–1005, 2000.

Roy D Crowninshield and Richard A Brand. A physiologically based criterion of muscle force prediction in locomotion. *Journal of biomechanics*, 14(11):793–801, 1981.

He Cui and Richard A Andersen. Posterior parietal cortex encodes autonomously selected motor plans. *Neuron*, 56(3):552–559, 2007.

Andrea d'Avella and Emilio Bizzi. Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences*, 102(8):3076–3081, 2005.

Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, 6(3):300, 2003.

Eran Dayan and Leonardo G Cohen. Neuroplasticity subserving motor skill learning. *Neuron*, 72(3):443–454, 2011.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Hugo Delivet-Mongrain, Hugues Leblond, and Serge Rossignol. Effects of localized intraspinal injections of a noradrenergic blocker on locomotion of high decerebrate cats. *Journal of neurophysiology*, 100(2):907–921, 2008.

Mahlon R DeLong and Thomas Wichmann. Circuits and circuit disorders of the basal ganglia. *Archives of neurology*, 64(1):20–24, 2007.

Michel Desmurget and Robert S Turner. Testing basal ganglia motor functions through reversible inactivations in the posterior internal globus pallidus. *Journal of neurophysiology*, 99(3):1057–1076, 2008.

Travis DeWolf, Terrence C Stewart, Jean-Jacques Slotine, and Chris Eliasmith. A spiking neural model of adaptive arm control. *Proceedings of the Royal Society B: Biological Sciences*, 283 (1843):20162134, 2016.

M. Diehl and S. Gros. *Linear and Nonlinear Programming*. Draft, 2016.

H Chris Dijkerman and Edward HF De Haan. Somatosensory processing subserving perception and action: Dissociations, interactions, and integration. *Behavioral and brain sciences*, 30(2):224–230, 2007.

Jonathan B Dingwell, Christopher D Mah, and Ferdinando A Mussa-Ivaldi. Experimentally confirmed mathematical model for human control of a non-rigid object. *Journal of Neurophysiology*, 91(3):1158–1170, 2004.

John P Donoghue and Jerome N Sanes. Motor areas of the cerebral cortex. *Journal of clinical neurophysiology: official publication of the American Electroencephalographic Society*, 11(4): 382–396, 1994.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12 (1):219–245, 2000.

T Drew. Motor cortical cell discharge during voluntary gait modification. *Brain research*, 457 (1):181–187, 1988.

Trevor Drew, Jacques-Etienne Andujar, Kim Lajoie, and Sergiy Yakovenko. Cortical mechanisms involved in visuomotor coordination during precision walking. *Brain research reviews*, 57(1):199–211, 2008.

Salvador Dura-Bernal, Samuel A Neymotin, Cliff C Kerr, Subhashini Sivagnanam, Amit Majumdar, Joseph T Francis, and William W Lytton. Evolutionary algorithm optimization of biological learning parameters in a biomimetic neuroprosthesis. *IBM journal of research and development*, 61(2/3):6–1, 2017.

Jacques Duysens and Henry WAA Van de Crommert. Neural control of locomotion; part 1: The central pattern generator from cats to humans. *Gait & posture*, 7(2):131–141, 1998.

Florin Dzeladini. From neuromechanical simulation to controllers for orthoses and lower-limb exoskeletons. *EPFL PhD Thesis*, 2019.

Arne D Ekstrom, Michael J Kahana, Jeremy B Caplan, Tony A Fields, Eve A Isham, Ehren L Newman, and Itzhak Fried. Cellular networks underlying human spatial navigation. *Nature*, 425(6954):184, 2003.

Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *science*, 338(6111): 1202–1205, 2012.

Roger M Enoka. *Neuromechanics of human movement*. Human kinetics, 2008.

EDWARD V Evarts, C Fromm, J Kroller, and VA Jennings. Motor cortex control of finely graded forces. *Journal of Neurophysiology*, 49(5):1199–1215, 1983.

Michael W Eysenck and Mark T Keane. *Cognitive psychology: A student's handbook*. Psychology press, 2015.

Anatol G Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture-ii. controllable parameters of the muscle. *Biofizika*, 11: 565–578, 1966.

Anatol G Feldman. Once more on the equilibrium-point hypothesis (λ model) for motor control. *Journal of motor behavior*, 18(1):17–54, 1986.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Tamar Flash and Binyamin Hochner. Motor primitives in vertebrates and invertebrates. *Current opinion in neurobiology*, 15(6):660–666, 2005.

Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

Leonardo Fogassi, Pier Francesco Ferrari, Benno Gesierich, Stefano Rozzi, Fabian Chersi, and Giacomo Rizzolatti. Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667, 2005.

H Forssberg. Ontogeny of human locomotor control i. infant stepping, supported locomotion and transition to independent locomotion. *Experimental Brain Research*, 57(3):480–493, 1985.

Jason Friedman and Tamar Flash. Trajectory of the index finger during grasping. *Experimental brain research*, 196(4):497–509, 2009.

Vittorio Gallese, Luciano Fadiga, Leonardo Fogassi, and Giacomo Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

Brian A Garner and Marcus G Pandy. A kinematic model of the upper limb based on the visible human project (vhp) image dataset. *Computer methods in biomechanics and biomedical engineering*, 2(2):107–124, 1999.

Brian A Garner and Marcus G Pandy. Musculoskeletal model of the upper limb based on the visible human male dataset. *Computer methods in biomechanics and biomedical engineering*, 4(2):93–126, 2001.

Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):206, 2013.

Israel M Gelfand and Mark L Latash. On the problem of adequate language in motor control. *Motor control*, 2(4):306–313, 1998.

Apostolos P Georgopoulos, John F Kalaska, Roberto Caminiti, and Joe T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(11):1527–1537, 1982.

Apostolos P Georgopoulos, Roberto Caminiti, John F Kalaska, and Joseph T Massey. Spatial coding of movement: a hypothesis concerning the coding of movement direction by motor cortical populations. *Experimental Brain Research*, 7(32):336, 1983.

Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

Apostolos P Georgopoulos, Ronald E Kettner, and Andrew B Schwartz. Primate motor cortex and free arm movements to visual targets in three-dimensional space. ii. coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, 8(8):2928–2937, 1988.

Apostolos P Georgopoulos, Hugo Merchant, Thomas Naselaris, and Bagrat Amirikian. Mapping of the preferred direction in the motor cortex. *Proceedings of the National Academy of Sciences*, 104(26):11068–11072, 2007.

Samuel J Gershman and Nathaniel D Daw. Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annual review of psychology*, 68:101–128, 2017.

PETER A Getting, PAUL R Lennard, and RICHARD I Hume. Central pattern generator mediating swimming in tritonia. i. identification and synaptic interactions. *Journal of neurophysiology*, 44(1):151–164, 1980.

Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *IEEE Transactions on neural systems and rehabilitation engineering*, 18(3):263–273, 2010.

Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1529): 2173–2183, 2003.

Gary Goldberg. Supplementary motor area structure and function: Review and hypotheses. *Behavioral and brain Sciences*, 8(4):567–588, 1985.

Patricia S Goldman-Rakic. Topography of cognition: parallel distributed networks in primate association cortex. *Annual review of neuroscience*, 11(1):137–156, 1988.

Michael S Graziano, Gregory S Yap, and Charles G Gross. Coding of visual space by premotor neurons. *Science*, 266(5187):1054–1057, 1994.

Michael SA Graziano, Dylan F Cooke, and Charlotte SR Taylor. Coding the location of the arm by sight. *Science*, 290(5497):1782–1786, 2000.

Sten Grillner. Locomotion in vertebrates: central mechanisms and reflex interaction. *Physiological reviews*, 55(2):247–304, 1975.

Sten Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228(4696): 143–149, 1985.

Sten Grillner. Control of locomotion in bipeds, tetrapods, and fish. *Comprehensive Physiology*, pages 1179–1236, 2011.

Sten Grillner, Peter Wallén, Nicholas Dale, Lennart Brodin, James Buchanan, and Russell Hill. Transmitters, membrane properties and network circuitry in the control of locomotion in lamprey. *Trends in Neurosciences*, 10(1):34–41, 1987.

Sten Grillner, Tanja Deliagina, A El Manira, RH Hill, GN Orlovsky, P Wallén, Ö Ekeberg, and A Lansner. Neural networks that co-ordinate locomotion and body orientation in lamprey. *Trends in neurosciences*, 18(6):270–279, 1995.

Valeriya Gritsenko, Sergiy Yakovenko, and John F Kalaska. Integration of predictive feedforward and sensory feedback signals for online control of visually guided movement. *Journal of Neurophysiology*, 102(2):914–930, 2009.

Jessy W Grizzle, Jonathan Hurst, Benjamin Morris, Hae-Won Park, and Koushil Sreenath. Mabel, a new robotic bipedal walker and runner. In *2009 American Control Conference*, pages 2030–2036. IEEE, 2009.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838, 2016.

Emmanuel Guigon, Pierre Baraduc, and Michel Desmurget. Computational motor control: redundancy and invariance. *Journal of neurophysiology*, 97(1):331–347, 2007.

Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801, 2005.

Daseong Han, Junyong Noh, Xiaogang Jin, Joseph S. Shin, and Sung Y. Shin. On-line real-time physics-based predictive motion control with balance recovery. In *Computer Graphics Forum*, volume 33, pages 245–254. Wiley Online Library, 2014.

R Happee and FCT Van der Helm. The control of shoulder muscles during goal directed movements, an inverse dynamic analysis. *Journal of biomechanics*, 28(10):1179–1191, 1995.

David C Harding, Kenneth D Brandt, and Ben M Hillberry. Finger joint force minimization in pianists using optimization techniques. *Journal of biomechanics*, 26(12):1403–1412, 1993.

Christopher M Harris and Daniel M Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695):780, 1998.

Richard F Hartl, Suresh P Sethi, and Raymond G Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM review*, 37(2):181–218, 1995.

Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

Jiping He, William S Levine, and Gerald E Loeb. Feedback gains for correcting small perturbations to standing posture. *IEEE Transactions on Automatic Control*, 36(3):322–332, 1991.

Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

Nicolas Heess, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, Martin Riedmiller, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Stewart Heitmann, Tjeerd Boonstra, and Michael Breakspear. A dendritic mechanism for decoding traveling waves: principles and applications to motor cortex. *PLoS computational biology*, 9(10):e1003260, 2013.

Stewart Heitmann, Tjeerd Boonstra, Pulin Gong, Michael Breakspear, and Bard Ermentrout. The rhythms of steady posture: Motor commands as spatially organized oscillation patterns. *Neurocomputing*, 170:3–14, 2015.

Okihide Hikosaka, Kae Nakamura, Katsuyuki Sakai, and Hiroyuki Nakahara. Central mechanisms of motor skill learning. *Current opinion in neurobiology*, 12(2):217–222, 2002.

Archibald Vivian Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, 126(843):136–195, 1938.

Mark R Hinder and Theodore E Milner. The case for an internal dynamics model versus equilibrium point control in human movement. *The Journal of Physiology*, 549(3):953–963, 2003.

Bruce Hoff and Michael A Arbib. Models of trajectory formation and temporal interaction of reach and grasp. *Journal of motor behavior*, 25(3):175–192, 1993.

Jeffrey R Hollerman and Wolfram Schultz. Dopamine neurons report an error in the temporal prediction of reward during learning. *Nature neuroscience*, 1(4):304, 1998.

Clay B Holroyd and Michael GH Coles. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological review*, 109(4):679, 2002.

Fay B Horak and Marjorie E Anderson. Influence of globus pallidus on arm movements in monkeys. i. effects of kainic acid-induced lesions. *Journal of Neurophysiology*, 52(2):290–304, 1984.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Donald F Hoyt and C Richard Taylor. Gait and the energetics of locomotion in horses. *Nature*, 292(5820):239, 1981.

Yildirim Hurmuzlu. Dynamics of bipedal gait: Part i—objective functions and the contact event of a planar five-link biped. *Journal of Applied Mechanics*, 60(2):331–336, 1993a.

Yildirim Hurmuzlu. Dynamics of bipedal galt: Part ii-stability analysis of a planar five-link biped. *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF APPLIED MECHANICS*, 60:337–337, 1993b.

Marco Hutter, C David Remy, Mark A Hoepflinger, and Roland Siegwart. Scarleth: Design and control of a planar running robot. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 562–567. IEEE, 2011.

AF Huxley. Muscular contraction. *The Journal of physiology*, 243(1):1–43, 1974.

Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *science*, 315 (5817):1416–1420, 2007.

Andrew Jackson, Jaideep Mavoori, and Eberhard E Fetz. Correlations between the same motor cortex cells and arm muscles during a trained task, free behavior, and natural sleep in the macaque monkey. *Journal of neurophysiology*, 97(1):360–374, 2007.

Herbert Jaeger. Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*, 2014.

Wojciech Jaśkowski, Odd Rune Lykkebø, Nihat Engin Toklu, Florian Trifterer, Zdeněk Buk, Jan Koutník, and Faustino Gomez. Reinforcement learning to run… fast. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 155–167. Springer, 2018.

IH Jenkins, DJ Brooks, PD Nixon, RS Frackowiak, and RE Passingham. Motor sequence learning: a study with positron emission tomography. *Journal of Neuroscience*, 14(6):3775–3790, 1994.

John F Kalaska. From intention to action: motor cortex and the control of reaching movements. In *Progress in Motor Control*, pages 139–178. Springer, 2009.

John F Kalaska and Donald J Crammond. Deciding not to go: neuronal correlates of response selection in a go/nogo task in primate premotor and parietal cortex. *Cerebral Cortex*, 5(5): 410–428, 1995.

Morton I Kamien and Nancy Lou Schwartz. *Dynamic optimization: the calculus of variations and optimal control in economics and management*. Courier Corporation, 2012.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Department of Biochemistry, Molecular Biophysics Thomas Jessell, Steven Siegelbaum, and AJ Hudspeth. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

Risa Kawai, Timothy Markman, Rajesh Poddar, Raymond Ko, Antoniu L Fantana, Ashesh K Dhawale, Adam R Kampff, and Bence P Ölveczky. Motor cortex is required for learning but not for executing a motor skill. *Neuron*, 86(3):800–812, 2015.

Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

Daniel Kernell. The motoneurone and its muscle fibres. *Journal of the Neurological Sciences*, 2006.

Łukasz Kidziński, Sharada Prasanna Mohanty, Carmichael F Ong, Zhewei Huang, Shuchang Zhou, Anton Pechenko, Adam Stelmaszczyk, Piotr Jarosik, Mikhail Pavlov, Sergey Kolesnikov, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 121–153. Springer, 2018.

Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

J Zico Kolter, Christian Plagemann, David T Jackson, Andrew Y Ng, and Sebastian Thrun. A probabilistic approach to mixed open-loop and closed-loop control, with application to extreme autonomous driving. In *2010 IEEE International Conference on Robotics and Automation*, pages 839–845. IEEE, 2010.

Konrad P Körding and Daniel M Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427(6971):244, 2004.

Vijaya Krishnamoorthy, Mark L Latash, John P Scholz, and Vladimir M Zatsiorsky. Muscle synergies during shifts of the center of pressure by standing persons. *Experimental brain research*, 152(3):281–292, 2003.

Sun-Yuan Kung and KI Diamantaras. A neural network learning algorithm for adaptive principal component extraction (apex). In *International Conference on Acoustics, Speech, and Signal Processing*, pages 861–864. IEEE, 1990.

Arthur D Kuo. An optimal control model for analyzing human postural balance. *IEEE transactions on biomedical engineering*, 42(1):87–101, 1995.

Kim Lajoie, Jacques-Étienne Andujar, Keir Pearson, and Trevor Drew. Neurons in area 5 of the posterior parietal cortex in the cat contribute to interlimb coordination during visually guided locomotion: a role in working memory. *Journal of neurophysiology*, 103(4):2234–2254, 2010.

Andrew D Lawrence. Error correction and the basal ganglia: similar computations for action, cognition and emotion? *Trends in Cognitive Sciences*, 4(10):365–367, 2000.

Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. Dexterous manipulation and control with volumetric muscles. *ACM Transactions on Graphics (TOG)*, 37(4):57, 2018.

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Trans. Graph.*, 38(4), 2019.

Sung-Hee Lee and Demetri Terzopoulos. Heads up!: biomechanical modeling and neuromuscular control of the neck. *ACM Transactions on Graphics (TOG)*, 25(3):1188–1198, 2006.

Yoonsang Lee, Moon Seok Park, Taesoo Kwon, and Jehee Lee. Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics (TOG)*, 33(6):218, 2014.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Andrew Levy, Robert Platt, and Kate Saenko. Hierarchical reinforcement learning with hindsight. *arXiv preprint arXiv:1805.08180*, 2018.

Chiang-Shan Ray Li, Camillo Padoa-Schioppa, and Emilio Bizzi. Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron*, 30(2):593–607, 2001.

Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2011.

Benjamin Libet. *Mind time: The temporal factor in consciousness*. Harvard University Press, 2009.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

CL Lim, NB Jones, Sarah K Spurgeon, and JJA Scott. Modelling of knee joint muscles during the swing phase of gait—-a forward dynamics approach using matlab/simulink. *Simulation Modelling Practice and Theory*, 11(2):91–107, 2003.

David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*, volume 2. Springer, 1984.

Andreas R Luft and Manuel M Buitrago. Stages of motor skill learning. *Molecular neurobiology*, 32(3):205–216, 2005.

Jack Macki and Aaron Strauss. *Introduction to optimal control theory*. Springer Science & Business Media, 2012.

Hugo Gravato Marques, Arjun Bharadwaj, and Fumiya Iida. From spontaneous motor activity to coordinated behaviour: a developmental model. *PLoS computational biology*, 10(7): e1003653, 2014.

Mathworks. Matlab optimization toolbox. *The MathWorks, Natick, MA, USA*, 2018.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Daniel McNamee and Daniel M Wolpert. Internal models in biological control. *Annual review of control, robotics, and autonomous systems*, 2:339–364, 2019.

David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review*, 95(3):340, 1988.

Chris Miall. Motor control: correcting errors and learning from mistakes. *Current Biology*, 20 (14):R596–R598, 2010.

R Chris Miall and Daniel M Wolpert. Forward models for physiological motor control. *Neural networks*, 9(8):1265–1279, 1996.

Matthew Millard, Thomas Uchida, Ajay Seth, and Scott L Delp. Flexing computational muscle: modeling and simulation of musculotendon dynamics. *Journal of biomechanical engineering*, 135(2):021005, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Ryan J Monti, Roland R Roy, and V Reggie Edgerton. Role of motor unit structure in defining function. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine*, 24(7):848–866, 2001.

Daniel W Moran and Andrew B Schwartz. Motor cortical activity during drawing movements: population representation during spiral tracing. *Journal of neurophysiology*, 82(5):2693–2704, 1999.

Pietro Morasso. Spatial control of arm movements. *Experimental brain research*, 42(2):223–227, 1981.

Franck Multon, Laure France, Marie-Paule Cani-Gascuel, and Giles Debunne. Computer animation of human walking: a survey. *The journal of visualization and computer animation*, 10(1):39–54, 1999.

Akira Murata, Vittorio Gallese, Giuseppe Luppino, Masakazu Kaseda, and Hideo Sakata. Selectivity for the shape, size, and orientation of objects for grasping in neurons of monkey parietal area aip. *Journal of neurophysiology*, 83(5):2580–2601, 2000.

Venkatesh N Murthy and Eberhard E Fetz. Coherent 25-to 35-hz oscillations in the sensorimotor cortex of awake behaving monkeys. *Proceedings of the National Academy of Sciences of the United States of America*, 89(12):5670, 1992.

Yoshihisa Nakayama, Tomoko Yamagata, Jun Tanji, and Eiji Hoshi. Transformation of a virtual action plan into a motor plan in the premotor cortex. *Journal of Neuroscience*, 28(41): 10287–10297, 2008.

Georg Northoff, R Steinke, D Nagel, C Czerwenka, O Grosser, P Danos, A Genz, R Krause, H Böker, HJ Otto, et al. Right lower prefronto-parietal cortical dysfunction in akinetic catatonia: a combined study of neuropsychology and regional cerebral blood flow. *Psychological Medicine*, 30(3):583–596, 2000.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

David J Ostry and Anatol G Feldman. A critical evaluation of the force control hypothesis in motor control. *Experimental brain research*, 153(3):275–288, 2003.

Marcus G Pandy, Felix E Zajac, Eunsup Sim, and William S Levine. An optimal control model for maximum-height human jumping. *Journal of biomechanics*, 23(12):1185–1198, 1990.

Liam Paninski, Matthew R Fellows, Nicholas G Hatsopoulos, and John P Donoghue. Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *Journal of neurophysiology*, 91(1):515–532, 2004.

Se-Woong Park, Tjeerd Dijkstra, and Dagmar Sternad. Learning to never forget—time scales and specificity of long-term memory of a motor skill. *Frontiers in computational neuroscience*, 7:111, 2013.

Xue Bin Peng and Michiel van de Panne. Learning locomotion skills using deeprl: Does the choice of action space matter? In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 12. ACM, 2017.

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4):41, 2017.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):143, 2018.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Richard Poppele and Gianfranco Bosco. Sophisticated spinal contributions to motor control. *Trends in neurosciences*, 26(5):269–276, 2003.

Jerry Pratt, Chee-Meng Chew, Ann Torres, Peter Dilworth, and Gill Pratt. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 20(2):129–143, 2001.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2827–2836. JMLR. org, 2017.

Carlos Quental, João Folgado, Jorge Ambrósio, and Jacinto Monteiro. A multibody biomechanical model of the upper limb including the shoulder girdle. *Multibody System Dynamics*, 28 (1-2):83–108, 2012.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Bryan L Riemann and Scott M Lephart. The sensorimotor system, part i: the physiologic basis of functional joint stability. *Journal of athletic training*, 37(1):71, 2002a.

Bryan L Riemann and Scott M Lephart. The sensorimotor system, part ii: the role of proprioception in motor control and functional joint stability. *Journal of athletic training*, 37(1):80, 2002b.

Giacomo Rizzolatti and Giuseppe Luppino. The cortical motor system. *Neuron*, 31(6):889–901, 2001.

Giacomo Rizzolatti and Corrado Sinigaglia. The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations. *Nature reviews neuroscience*, 11(4):264, 2010.

Giacomo Rizzolatti, M Gentilucci, RM Camarda, V Gallese, G Luppino, M Matelli, and L Fogassi. Neurons related to reaching-grasping arm movements in the rostral part of area 6 (area 6aβ). *Experimental brain research*, 82(2):337–350, 1990.

Jinsook Roh, William Z Rymer, Eric J Perreault, Seng Bum Yoo, and Randall F Beer. Alterations in upper limb muscle synergy structure in chronic stroke survivors. *Journal of neurophysiology*, 109(3):768–781, 2012.

Oliver Röhrle, Michael Sprenger, Ellankavi Ramasamy, and Thomas Heidlauf. Multiscale skeletal muscle modeling: from cellular level to a multi-segment skeletal muscle model of the upper limb. In *Computer models in biomechanics*, pages 103–116. Springer, 2013.

Jennifer C Romano, James H Howard Jr, and Darlene V Howard. One-year retention of general and sequence-specific skills in a probabilistic, serial reaction time task. *Memory*, 18(4): 427–441, 2010.

David A Rosenbaum. *Human motor control*. Academic press, 2009.

David A Rosenbaum, Loukia D Loukopoulos, Ruud GJ Meulenbroek, Jonathan Vaughan, and Sascha E Engelbrecht. Planning reaches by evaluating stored postures. *Psychological review*, 102(1):28, 1995.

David A Rosenbaum, Ruud J Meulenbroek, Jonathan Vaughan, and Chris Jansen. Posture-based motion planning: applications to grasping. *Psychological review*, 108(4):709, 2001.

Ilya A Rybak, Natalia A Shevtsova, Myriam Lafreniere-Roula, and David A McCrea. Modelling spinal circuitry involved in locomotor pattern generation: insights from deletions during fictive locomotion. *The Journal of physiology*, 577(2):617–639, 2006.

Robert L Sainburg, C Ghez, and D Kalakanis. Intersegmental dynamics are controlled by sequential anticipatory, error correction, and postural mechanisms. *Journal of neurophysiology*, 81(3):1045–1056, 1999.

Jerome N Sanes and John P Donoghue. Oscillations in local field potentials of the primate motor cortex during voluntary movement. *Proceedings of the National Academy of Sciences*, 90(10):4470–4474, 1993.

Robert M Sanner and Makiko Kosha. A mathematical model of the adaptive control of human arm motions. *Biological Cybernetics*, 80(5):369–382, 1999.

Veronica J Santos and Francisco J Valero-Cuevas. Reported anatomical variability naturally leads to multimodal distributions of denavit-hartenberg parameters for the human thumb. *IEEE Transactions on Biomedical Engineering*, 53(2):155–163, 2006.

Luisa Sartori, Andrea Camperio, Maria Bulgheroni, and Umberto Castiello. Reach-to-grasp movements in macaca fascicularis monkeys: the isochrony principle at work. *Frontiers in psychology*, 4:114, 2013.

John P Scholz and Gregor Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental brain research*, 126(3):289–306, 1999.

John P Scholz, Ning Kang, David Patterson, and Mark L Latash. Uncontrolled manifold analysis of single trials during multi-finger force production by persons with and without down syndrome. *Experimental Brain Research*, 153(1):45–58, 2003.

Christian Schumacher and André Seyfarth. Sensor-motor maps for describing linear reflex composition in hopping. *Frontiers in computational neuroscience*, 11:108, 2017.

Stephen H Scott. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5(7):532, 2004.

Stephen H Scott. The computational and neural basis of voluntary motor control and planning. *Trends in cognitive sciences*, 16(11):541–549, 2012.

Stephen H Scott, Tyler Cluff, Catherine R Lowrey, and Tomohiko Takei. Feedback control during voluntary motor actions. *Current opinion in neurobiology*, 33:85–94, 2015.

Lauren E Sergio, Catherine Hamel-Pâquet, and John F Kalaska. Motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks. *Journal of neurophysiology*, 94(4):2353–2378, 2005.

Ajay Seth, John J McPhee, and Marcus G Pandy. Multi-joint coordination of vertical arm movement. *Applied Bionics and Biomechanics*, 1(1):45–56, 2003.

Reza Shadmehr, Steven P Wise, et al. *The computational neurobiology of reaching and pointing: a foundation for motor learning*. MIT press, 2005.

Philip Shaw, Noor J Kabani, Jason P Lerch, Kristen Eckstrand, Rhoshel Lenroot, Nitin Gogtay, Deanna Greenstein, Liv Clasen, Alan Evans, Judith L Rapoport, et al. Neurodevelopmental trajectories of the human cerebral cortex. *Journal of Neuroscience*, 28(14):3586–3594, 2008.

Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, 36:337–359, 2013.

Charles Sherrington. *The integrative action of the nervous system*. CUP Archive, 1952.

Charles Scott Sherrington. Flexion-reflex of the limb, crossed extension-reflex, and reflex stepping and standing. *The Journal of physiology*, 40(1-2):28–121, 1910.

M Lo Shik. Control of walking and running by means of electrical stimulation of the midbrain. *Biophysics*, 11:659–666, 1966.

Lior Shmuelof, John W Krakauer, and Pietro Mazzoni. How is a motor skill learned? change and invariance at the levels of task success and trajectory control. *Journal of neurophysiology*, 108(2):578–594, 2012.

Miguel PT Silva and Jorge AC Ambrósio. Solution of redundant muscle forces in human locomotion with multibody dynamics and optimization tools. *Mechanics Based Design of Structures and Machines*, 2003.

Samuel J Sober and Philip N Sabes. Flexible strategies for sensory integration during motor planning. *Nature neuroscience*, 8(4):490, 2005.

John F Soechting, Christopher A Buneo, Uta Herrmann, and Martha Flanders. Moving effortlessly in three dimensions: does donders' law apply to arm movement? *Journal of Neuroscience*, 15(9):6271–6280, 1995.

Kwang Won Sok, Manmyung Kim, and Jehee Lee. Simulating biped behaviors from human motion data. In *ACM Transactions on Graphics (TOG)*, volume 26, page 107. ACM, 2007.

Olaf Sporns and Gerald M Edelman. Solving bernstein's problem: A proposal for the development of coordinated movement by selection. *Child development*, 64(4):960–981, 1993.

Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

Kok Lay Teo, C Goh, and K Wong. *A unified computational approach to optimal control problems*. Longman Science and Technology, 1991.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

Darryl G Thelen. Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *Journal of biomechanical engineering*, 125(1):70–77, 2003.

Esther Thelen and Linda B Smith. *A dynamic systems approach to the development of cognition and action*. MIT press, 1996.

Lena H Ting and Jane M Macpherson. A limited set of muscle synergies for force control during a postural task. *Journal of neurophysiology*, 93(1):609–613, 2005.

Lena H Ting and J Lucas McKay. Neuromechanics of muscle synergies for posture and movement. *Current opinion in neurobiology*, 17(6):622–628, 2007.

Emanuel Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9): 907, 2004.

Emanuel Todorov. Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural computation*, 17(5):1084–1108, 2005.

Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226, 2002.

Emanuel Todorov and Weiwei Li. Optimal control methods suitable for biomechanical systems. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, volume 2, pages 1758–1761. IEEE, 2003.

Gelsy Torres-Oviedo, Jane M Macpherson, and Lena H Ting. Muscle synergy organization is robust across a variety of postural perturbations. *Journal of neurophysiology*, 96(3): 1530–1546, 2006.

Matthew C Tresch, Vincent CK Cheung, and Andrea d'Avella. Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets. *Journal of neurophysiology*, 95(4):2199–2212, 2006.

Brian R Umberger and Ross H Miller. Optimal control modeling of human movement. *Handbook of Human Motion*, pages 327–348, 2018.

Brian R Umberger, Karin GM Gerritsen, and Philip E Martin. A model of human muscle energy expenditure. *Computer methods in biomechanics and biomedical engineering*, 6(2):99–111, 2003.

Leslie G Ungerleider, Julien Doyon, and Avi Karni. Imaging brain plasticity during motor skill learning. *Neurobiology of learning and memory*, 78(3):553–564, 2002.

Yoji Uno, Mitsuo Kawato, and Rika Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics*, 61(2):89–101, 1989.

Francisco J Valero-Cuevas, Jae-Woong Yi, Daniel Brown, Robert V McNamara, Chandana Paul, and Hood Lipson. The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Transactions on Biomedical Engineering*, 54(6):1161–1166, 2007.

Antonie J van den Bogert, Karin GM Gerritsen, and Gerald K Cole. Human muscle modelling from a user's perspective. *Journal of Electromyography and Kinesiology*, 8(2):119–124, 1998.

Frans CT Van der Helm. Analysis of the kinematic and dynamic behavior of the shoulder mechanism. *Journal of biomechanics*, 27(5):527–550, 1994.

Frans CT Van der Helm, Alfred C Schouten, Erwin de Vlugt, and Guido G Brouwn. Identification of intrinsic and reflexive components of human arm dynamics during postural control. *Journal of neuroscience methods*, 119(1):1–14, 2002.

P Viviani and G McCollum. The relation between linear extent and velocity in drawing movements. *Neuroscience*, 10(1):211–218, 1983.

Jonathan D Wallis and Earl K Miller. From rule to response: neuronal processes in the premotor and prefrontal cortex. *Journal of neurophysiology*, 90(3):1790–1806, 2003.

Megan Wang, Christeva Montanede, Chandramouli Chandrasekaran, Diogo Peixoto, Krishna V Shenoy, and John F Kalaska. Macaque dorsal premotor cortex exhibits decision-related activity only when specific stimulus–response associations are known. *Nature communications*, 10(1):1793, 2019.

Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1):42–56, 2003.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Daniel B Willingham. A neuropsychological theory of motor skill learning. *Psychological review*, 105(3):558, 1998.

U Windhorst. Muscle proprioceptive feedback and spinal networks. *Brain research bulletin*, 73(4-6):155–202, 2007.

Jack M Winters, Savio LY Woo, and Idd Delp. *Multiple muscle systems: Biomechanics and movement organization*. Springer Science & Business Media, 2012.

Daniel M Wolpert and Zoubin Ghahramani. Computational principles of movement neuroscience. *Nature neuroscience*, 3(11s):1212, 2000.

Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7-8):1317–1329, 1998.

Jungdam Won, Jongho Park, Kwanyu Kim, and Jehee Lee. How to train your dragon: example-guided control of flapping flight. *ACM Transactions on Graphics (TOG)*, 36(6):198, 2017.

Stephen J Wright. *Primal-dual interior-point methods*, volume 54. Siam, 1997.

Gabriele Wulf, Charles Shea, and Rebecca Lewthwaite. Motor skill learning and performance: a review of influential factors. *Medical education*, 44(1):75–84, 2010.

Weitao Xi and C David Remy. Optimal gaits and motions for legged robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3259–3265. IEEE, 2014.

Yujiang Xiang, Jasbir S Arora, and Karim Abdel-Malek. Physics-based modeling and simulation of human walking: a review of optimization-based and other approaches. *Structural and Multidisciplinary Optimization*, 42(1):1–23, 2010.

T Yang, ER Westervelt, A Serrani, and James P Schmiedeler. A framework for the control of stable aperiodic walking in underactuated planar bipeds. *Autonomous Robots*, 27(3):277, 2009.

KangKang Yin, Kevin Loken, and Michiel Van de Panne. Simbicon: Simple biped locomotion control. In *ACM Transactions on Graphics (TOG)*, volume 26, page 105. ACM, 2007.

Felix E Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical reviews in biomedical engineering*, 17(4):359–411, 1989.

Berat Denizdurduran

Neurorobotics and Simplification Group, Blue Brain Project
Brain Mind Institute, École Polytechnique Fédérale de Lausanne
e-mail: berat.denizdurduran@epfl.ch,

Research Interests

Brain Inspired Computational Models, Neurorobotics, Sensorimotor Learning, Nonlinear Programming, Optimal Control, Reinforcement Learning, Gaussian Processes, Musculoskeletal Simulations.

Education

École Polytechnique Fédérale de Lausanne, Blue Brain Project, Neurorobotics Group

Ph.D. Candidate, Neuroscience — May 2013 - Present

- Thesis Title: "Reverse Engineering the Motor Control System"
- Advisors: Prof. Henry Markram and Dr. Marc-Oliver Gewaltig

Istanbul Technical University, Electronics and Communications Engineering Department

M.Sc. Degree, Electronics Engineering, — June 2012

- Thesis Title: "Learning How to Select an Action, From Bifurcation Theory to the Brain-Inspired Computational Model"
- Advisor: Assoc. Prof. Dr. Neslihan Serap Sengor

Istanbul Technical University, Electronics and Communications Engineering Department

B.Sc. Degree, Electronics Engineering — June 2010

- Thesis Title: "Dynamical Analysis of Neuron Models and Networks by XPPAUT"
- Advisor: Assoc. Prof. Dr. Neslihan Serap Sengor

Academic Experience

Teaching Assistant

- Unsupervised and reinforcement learning in neural networks Fall 2014, 2015, 2016, 2017, 2018
- EPFLx-RoboX-Neurorobotics MOOC 2019
- Sensorimotor neuroprosthetics — Spring 2014

Supervised Master Projects

- Jonny Quarta, "Ball-balancer control with continuous-time and space reinforcement learning in Spinnaker" , 2014
- Eleftherios Zisis, " CPG-based locomotion control of realistic mouse model", 2015
- Udaranga Wickramasinghe, " Musculoskeletal arm control via policy gradient learning", 2016
- Antoine Sauvage, " Musculoskeletal arm control with twitching and bubbling", 2017
- Loic Jeanningros, " Spiking neural network of spinal cord circuit", 2018

Invited Talk

- Denizdurduran, B. and Gewaltig, M-O., "Closed-loop motor learning of spiking neural networks", Autonomous agents and learning workshop, Tokyo, Japan, 2015.

Guest Assistant — June, 2012 - January, 2013

École Polytechnique Fédérale de Lausanne, Blue Brain Project, NeuroRobotics Group

- Project Title: "Parameter Extraction to Simplify the Conductance-based Neuron Models"
- Advisor: Dr. Marc-Oliver Gewaltig

Researcher February, 2010 - June, 2012

Istanbul Technical University, Electronics and Communications Engineering Department, Neuroscience Modeling and Research Group

- Project Title: "Modelling Neural Substructures Active in Decision Making, Reward Based Learning and Goal Directed Behavior"
- Grant: The Scientific and Technological Research Council of Turkey
- Advisor: Assoc. Prof. Dr. Neslihan Serap Sengor

Reviewer

- Bio-Inspired Computing: Theories and Application, (BIC-TA) 2014
- Frontiers in Neurorobotics
- The 24^{th} International Conference on Artificial Neural Networks, ICANN 2014

Publications

Denizdurduran, B., Sengor, N.S.:"Learning How to Select an Action: A Computational Model," *Proceedings of The* 22^{nd} *International Conference on Artificial Neural Networks (ICANN)*, 2012, Lausanne, Switzerland.

Yucelgen, C., **Denizdurduran, B.**, Metin, S., Elibol, R., Sengor, N.S.:"A Biophysical Network Model Displaying the Role of Basal Ganglia Pathways in Action Selection," *Proceedings of The* 22^{nd} *International Conference on Artificial Neural Networks (ICANN)*, 2012, Lausanne, Switzerland.

Denizdurduran, B., Sengor, N.S.: "A Realization of Goal-Directed Behavior, Implementing a Robot Model Based on Cortico-Striato-Thalamic Circuits," *Proceedings of The* 4^{th} *International Conference on Agents and Artificial Intelligence (ICAART)*, pp. 289-294, 2012, Vilamoura, Portugal.

Denizdurduran, B., Gewaltig, M-O, Markram, H: "Closed-loop motor control with Reinforcement Learning and Optimal Control," **in preparation**.

Denizdurduran, B., Gewaltig, M-O, Markram, H: "Computational motor circuit model of human arm control," **in preparation**.

Summer Schools

- Advanced Course in Computational Neuroscience 2014
- The 3^{rd} SpiNNaker Training Workshop, April 2014, Manchester, UK
- Barcelona Cognition, Brain and Technology Summer School 2011
- IBRO/UNESCO Cape Town School on Advanced Theoretical and Computational Neurosciences 2011
- Artificial Intelligence at Katholieke Universiteit Leuven 2010

References

Dr. Marc-Oliver Gewaltig
Ecole Polytechnique Federale de Lausanne, Blue Brain Project, Neurorobotics Group
Tel: 00 41 21 6931866, e-mail: marc-oliver.gewaltig@epfl.ch

Prof. Henry Markram
Ecole Polytechnique Federale de Lausanne, Blue Brain Project
Tel: 00 41 21 6939536, e-mail: henry.markram@epfl.ch

Assoc. Prof. Dr. Neslihan Serap Sengor
Istanbul Technical University, Electronics and Communications Department
Tel: 00 90 212 285 3610, e-mail: sengorn@itu.edu.tr

**Ce document a été imprimé au Centre d'impression EPFL,
imprimerie climatiquement neutre, certifiée myClimate.**