NOCH: A framework for biologically plausible models of neural motor control

by

Travis DeWolf

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis examines the neurobiological components of the motor control system and relates it to current control theory in order to develop a novel framework for models of motor control in the brain. The presented framework is called the Neural Optimal Control Hierarchy (NOCH). A method of accounting for low level system dynamics with a Linear Bellman Controller (LBC) on top of a hierarchy is presented, as well as a dynamic scaling technique for LBCs that drastically reduces the computational power and storage requirements of the system. These contributions to LBC theory allow for low cost, high-precision control of movements in large environments without exceeding the biological constraints of the motor control system.

Acknowledgements

I would like to thank my wife, Julie, my parents, both of them, my supervisor, Chris, the other members of my lab, the UW staff for making sure my thesis is well formatted, as well as Jim Steinman and Marvin Lee Aday, whose work has been an unending source of inspiration to me.

Dedication

This one goes out to the one I love.

Contents

List of Figures

Chapter 1

Introduction

Motor control affects all aspects of a person's life. The ability to efficiently and accurately carry out motor actions is heavily dependent on the proper functioning of the motor control system in the brain. This thesis examines the neurobiological components of this system and relates it to mathematical systems from current control theory in order to develop a novel comprehensive framework for models of motor control in the brain. The presented framework is entitled the Neural Optimal Control Hierarchy (NOCH).

1.1 Motivation

Currently in the field of motor control research there are very few general models of the motor control system. Much work has been done on the individual neurobiological components of the motor control system, such as the primary motor cortex [54, 108, 117] or basal ganglia [14, 50, 121], but there has been little done researching the structure and functioning of the motor control system as a whole. Similarly, in control theory there has been much focus on control of motor actions in systems with basic biological constraints, such as systems that can operate with the observed 150-250ms delay on visual feedback [63], but little has been done investigating potential neural sites for the performed functions.

The primary motivation for the work done here is to further the understanding of the motor control system by bringing together the current neurobiological research of the motor system components and control theory for motor actions into a biologically plausible framework that provides a context for the investigation of these neural areas

and further defines the required functions and constraints of mathematical models.

Understanding the systems and mechanisms underlying the ability to perform motor actions should provide insight into the development of skillful movements and effective methods of treatment for malfunctioning motor systems. For instance, many current medical treatments are based on our understanding of the function of the motor control system in the brain, including the most effective current treatments of Parkinson's and Huntington's [1, 33, 70, 102, 104]. As well, neural prosthetics have become an area of renewed research interest recently [5, 25, 40, 110, 115], and would benefit from a better understanding of the neural underpinnings of motor control. Additionally, the field of robotics is heavily inspired by biology; a clearer understanding of how nature accomplishes efficient and precise motor control is critical to the development of advanced robotic systems. As human interaction with technology continues to expand, ergonomic design and intuitive control based on the principles of human movement and motor control will also become increasingly important.

Together, these consideration highlight the importance of empowering our understanding of neural systems involved in motor control.

1.2 Past models

The MOSAIC model of neural motor control is perhaps the most prominent model of motor control currently, presented first in [55], though developed from a model presented in [64]. The MOSAIC is able to learn the internal models for operating in a given environment by combining feedforward and feedback sensorimotor information. Although this model assigns function to the different areas of the brain involved in motor control and gives a structure to the flow of information through the system, its implementation is centered around machine learning techniques. Similarly, a neural network approach to arm control is described in [15]. While the brain indeed is constantly learning and these techniques might therefore be valid representations of the operation of different areas, the machine learning approach provides very limited insight into the actual functions being calculated by the system. The problem with this then is that the applicability is restricted and any hopes of understanding or being able to fix errors in the system is greatly reduced. While the MOSAIC system is able to account for some of the phenomena surrounding motor control, the methods used to achieve these results are not conducive to a deep understanding of the system operation.

In [72], a hierarchal optimal control system is presented, governing the movement of a 7-DOF arm model. Building on research from [131], the model is able to successfully create realistic arm movement trajectories. Linear Bellman controllers are a recent development in control theory, reducing the problem of optimal control to the solution of a linear equation. In [133], an implementation of simple arm control using through linear Bellman control is briefly described. Both of these models are important contributions to the field of motor control research, but only implement basic biological constraints and give little consideration to the underlying neural structures.

The work done in [124], begins to bridge the gap between neurobiological processes and the functions of control theory, providing a basis for the examination of a biological implementation for several of the basic operations in mathematical models. This work, while important, is a very general low level model that does not address the function of the motor control system. Indeed, the majority of work done in engineering and robotics is only 'biologically inspired', drawing upon one or two of the basic principles of the motor control system and disregarding the actual neural processes [6, 32, 88, 124].

1.3 Thesis structure

The structure of this thesis is as follows: The next chapter will introduce the relevant neurobiology, providing the necessary context for the development of a biologically plausible model of motor control in the brain. The following chapter presents the elements of control theory used to form the mathematical underpinnings of the framework, beginning with a brief review and then expanding upon recent hierarchical and optimal control theory research. At the end of this chapter several novel control techniques are presented, forming the practical contributions of the work done here. The final chapter presents the theoretical contributions of the thesis, detailing a framework for biologically plausible models of motor control in the brain. Predictions of this work are then presented as well as empirical support for the framework. The thesis concludes by detailing directions for future research.

Chapter 2

The neuroanatomy of the motor control system

The motor system is made up of three primary components, the spine, brainstem, and forebrain, shown in Figure 2.1. The system is hierarchically structured such that each component handles increasingly complex aspects of motor control: the spine dealing with the most basic control, such as reflexes; and the forebrain controlling highly complicated volitional movement.

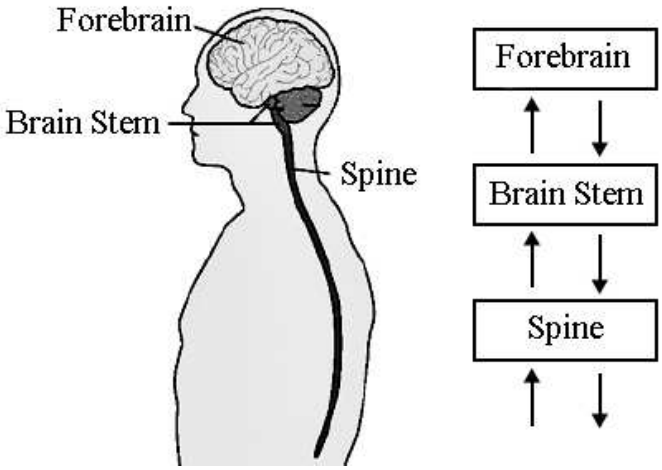

Figure 2.1: The motor control system, taken and modified from [44].

At each level of the motor hierarchy, the recieved peripheral sensory information is used to modify the motor control output signal. Distinct neural populations at each level project in parallel to sensory relay nuclei and other structures which keep the sensory processing centers informed about the ongoing motor commands. In addition,

these projections allow the higher motor control centers to filter the information that reaches them, restricting the flow to task relevant information only [44].

At the lowest level of the hierarchy, the spinal cord houses the neural circuitry for reflexes and automated rhythmic actions, such as locomotion. The most basic of these circuits consist of a single synapse connecting a primary sensory neuron and a motor neuron [93]. Most circuits, however, contain multiple connections and interneurons, which are neurons found between afferent and efferent neurons, allowing for more complex movements. These circuits also recieve input from the higher level motor control centers through descending pathways which have a large regulatory effect on the motor output [26]. Importantly, to reach the muscles, all motor commands must go through motor neurons, whose axons connect directly with the skeletal muscles.

The second level of the motor systems hierarchy is the brain stem. The brain stem can be divided into two subsystems, the medial and lateral descending systems, with both recieving input from the forebrain and projecting to the spinal cord [26]. The medial systems are involved in control of the proximal muscle systems through integrating the relevant visual, vestibular, and somatosensory information. The lateral descending systems include those that control the more distal limb muscles, such as the arm and hand. In addition to these systems the brain stem also houses circuits that control the movements of the eyes and head.

At the top of the hierarchy lies the forebrain, with several relevant cortical areas. The primary motor cortex and the premotor areas project both directly and indirectly to the spinal cord through corticomotoneural (CM) cells and the brain stem descending systems. From these areas complex sequences of motor actions are generated and sent to the lower levels for execution.

Overall, the areas of interest that will be detailed are the primary motor cortex (M1), the premotor areas (PM, SMA, CMA), the basal ganglia (BG), cerebellum (CB), and the thalamus (Th). The sensory procesing areas are also used in this model. However, they are represented as simply providing sensory feedback to the cortex. They are a highly complicated system and are beyond the scope of this thesis.

In 1909, Korbinian Brodmann published maps of the cortical areas of human, monkey, and the brains of other species, and distinguished the different areas in each based on the organization of the neurons revealed by Nissl staining [123]. In the century since the Brodmann areas map was first pusblished, there has been much debate and many proposed refinements. However, the Brodmann map remains the most well known and frequently cited cytoarchitecutural organization of the human

cortex, and will be referred to throughout this chapter.

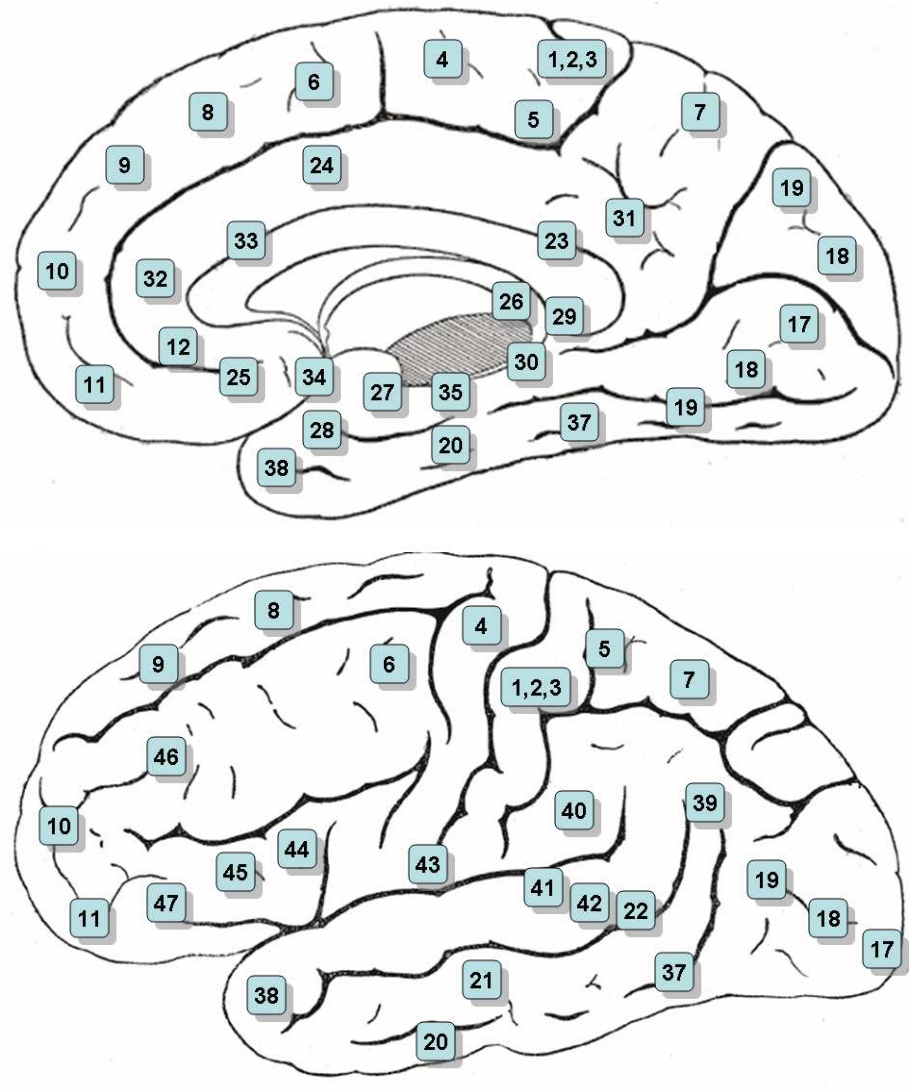

Figure 2.2: The Brodmann areas map of the cortex, taken from [126].

The remainder of this chapter is organized as follows: The next section describes the spine and brain stem to give a functional context for the higher processing centers. Then M1, PM, SMA, CMA, basal ganglia, cerebellum, and thalamus are discussed in turn. Each section will describe the area's function in the motor control system, as relevant for the remainder of this thesis.

2.1 The spine

The spinal cord is comprised of 33 separate vertebrae and is sectioned into four parts, the sacral, lumbar, thoracic, and cervical. The motor nuclei that run inside the vertebrae are arranged according to which muscles they innervate along a medial-lateral axis. The medial nuclei are involved in the activation of the muscles in the neck and back. Among the lateral nuclei the more medial motor neurons control proximal muscles, while the more lateral nuclei are involved in the control of distal muscles [44].

There is varying interconnection between the medial and lateral motor nuclei. Where the medial motor neurons are connected by propriospinal neurons with large axons and across several segments of the spinal cord, the lateral nuclei project to each other through propriospinal neurons with shorter axons across fewer segments, as shown in Figure 2.3. This reduced interconnectedness is critical for the independent control of distal muscles.

In addition, the spinal cord houses all of the circuitry for reflexes. Reflexes are automated responses to external stimuli that are performed without volitional control. They are realized by simple neural circuits consisting of at least one primary sensory neuron and motor neuron, and usually one or more interneurons.

A canonical example of a simple but important reflex is when your hand is quickly pulled away from a hot stove when placed there unknowingly: the stronger the intensity of the stimulus, the stronger the response of the reflex. This functional dependence shows that a reflex is not simply a stereotyped movement pattern, but is rather governed by the properties of the stimulus. The reason that a reflex has moved your hand from the hot stove before you are consciously aware that your hand is burning, is because of the neural circuit in the spine responding directly to the stimulus, which allows a fast response to be generated. The distance travelled and the amount of processing required for a reflex is significantly less than for a conscious response, operating in the range of 30-50ms [63] as compared to the 150-250ms range required for conscious responses [63]. This makes spinal reflexes essential for quick, basic movements in response to external stimuli [93].

Reflexes, automatic as they are, are not completely independent of higher level control centers and conscious control. They are regulated to a large degree by commands from descending pathways. This allows for flexibility which is taken advantage of by the higher control centers in the execution of complex motor actions, incorpo-

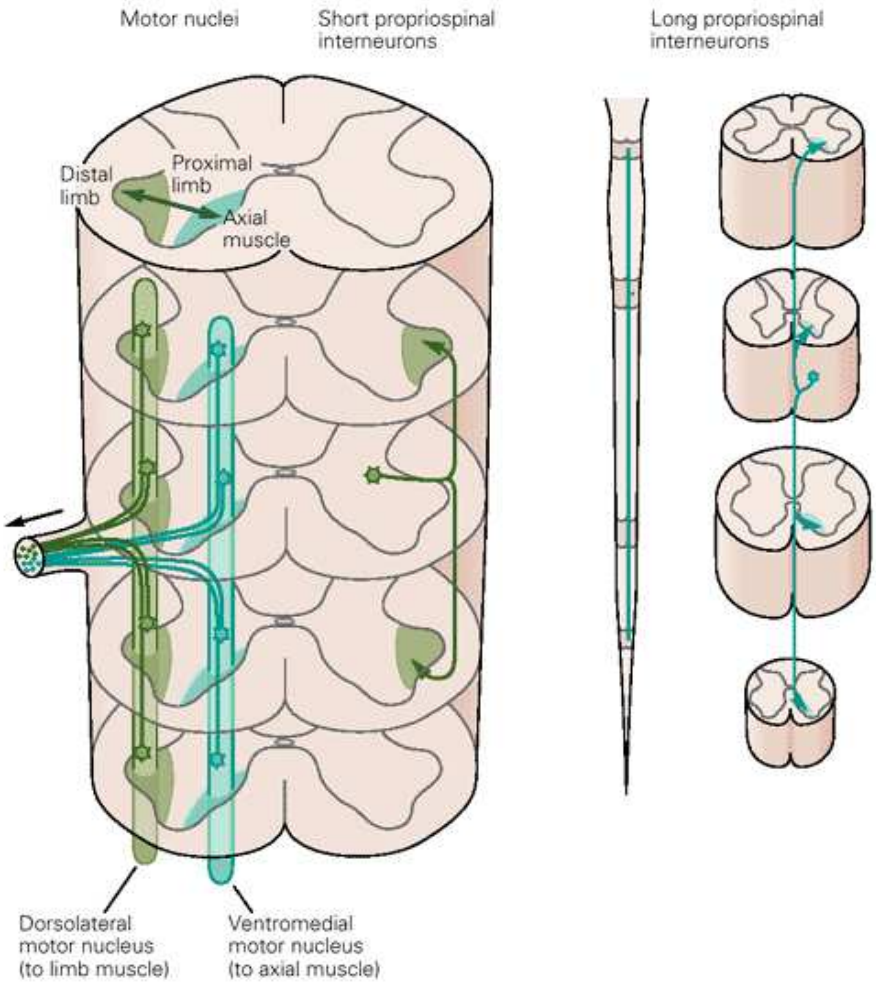

Figure 2.3: The medial nuclei are connected by longer interneurons, while the lateral nuclei are connected by shorter interneurons. Taken from [44].

rating modulated reflex circuits into descending control signals; simplifying the information sent from the cortex to the lower levels. This modulation of the reflex circuit defines what is called a 'functional set', which is the state of the reflex pathways for a specific task.

2.1.1 Reflexes

There are several different types of reflexes, defined by the structure of their neural circuit. The knee-jerk reflex circuit is shown in Figure 2.4. Reciprocal innervation reflexes are circuits that excite one group of muscles while inhibiting their antagonists,

useful for responses such as quickly pulling one's arm away from a painful stimulus. Crossed-extention reflexes cause the opposite effect on muscles in the contralateral limb, and are useful for load bearing and balance. Stretch reflexes act to resist the lengthening of a muscle, and can be used to reinforce descending motor commands.

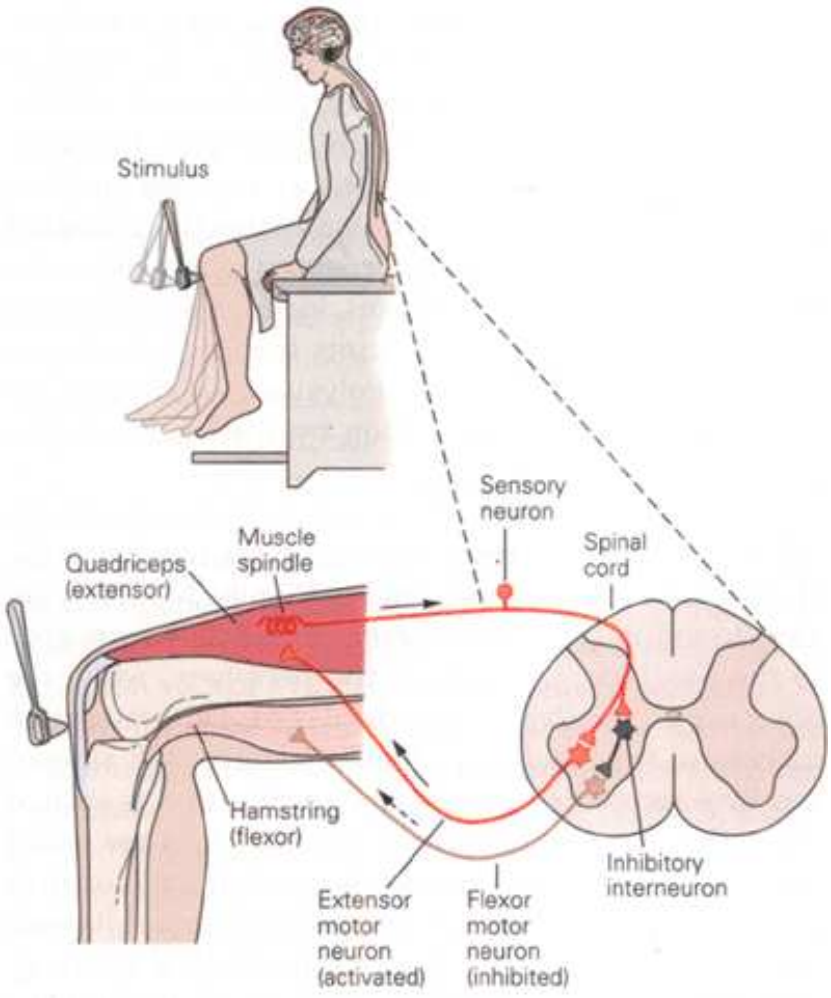

Figure 2.4: The knee-jerk reflex is controlled by a monosynaptic neural circuit in the spine. Taken from [93].

The importance of these reflexes, and the ability to regulate their actions, becomes clear when considered in the context of a control signal issued from higher control centers in the motor system, where the controlled muscles are subject to slight peterbations from unaccounted-for external forces. For example, controlling one's arm when there is a breeze. To have the sensory feedback information travel up to the higher control levels, calculate a corrective signal and send it back down would take far too long. In the amount time that this sort of process would take, the external

peturbations would have changed and the system would indefinitely be attempting to correct for the external forces sensed moments ago. The supraspinal system is in fact recognized as being incapable of generating rapid correction to undesired deviations from the motor plan [130]. Therefore it is necessary to have a system in place that can accept motor command signals, implement them, and immediately correct for deviations from the specified motor plan.

In the NOCH framework, the spinal system corrects for small deviations from descending motor commands caused by predicted external forces.

2.1.2 Motor Primitives

In addition to providing reflex circuits to keep motor actions on track that simplify the motor control signals, studies of spinalized frogs and rats suggest the presence of something akin to a basic building block of motor actions in the spine [84]. The study performed electrical stimulation of the interneuronal circuitry in the lumbar spinal cord of frogs and rats which resulted in muscle activations imposing a specific posture upon the innervated muscles, independent of the initial position at the time of stimulation. The experimenters attached a force transducer to the right ankle of a spinalized frog and measured the responses at various leg orientations. They found that the resulting movements were well-structured and predictable, forming a vector field, dubbed a 'spinal force field' [84], seen in Figure 2.5.

Interestingly, it was discovered that the fields induced by electrical stimulation of the lumbar spinal cord obey vector summation rules [85]. The resulting force field produced by stimulating two separate points is can be predicted by adding together the force fields obtained when the sites were stimulated independently. This vector summation of force fields implies that many of the complex nonlinearities of movement can be at least partially removed [85].

Indeed it has been shown that using the force fields as motor primitives allows for a wide repetoire of motor behaviours to be easily induced [83]. The force fields generated are nonlinear functions of velocity and limb position, and allow for the descending signals to command a weighted summation of the various motor primitives. This spinal characteristic further reduces the complexity of information required in a motor signal from higher control centers.

It is apparent that the systems and circuits of the spinal cord are necessary to consider when building a biologically plausible model of motor control. The existence

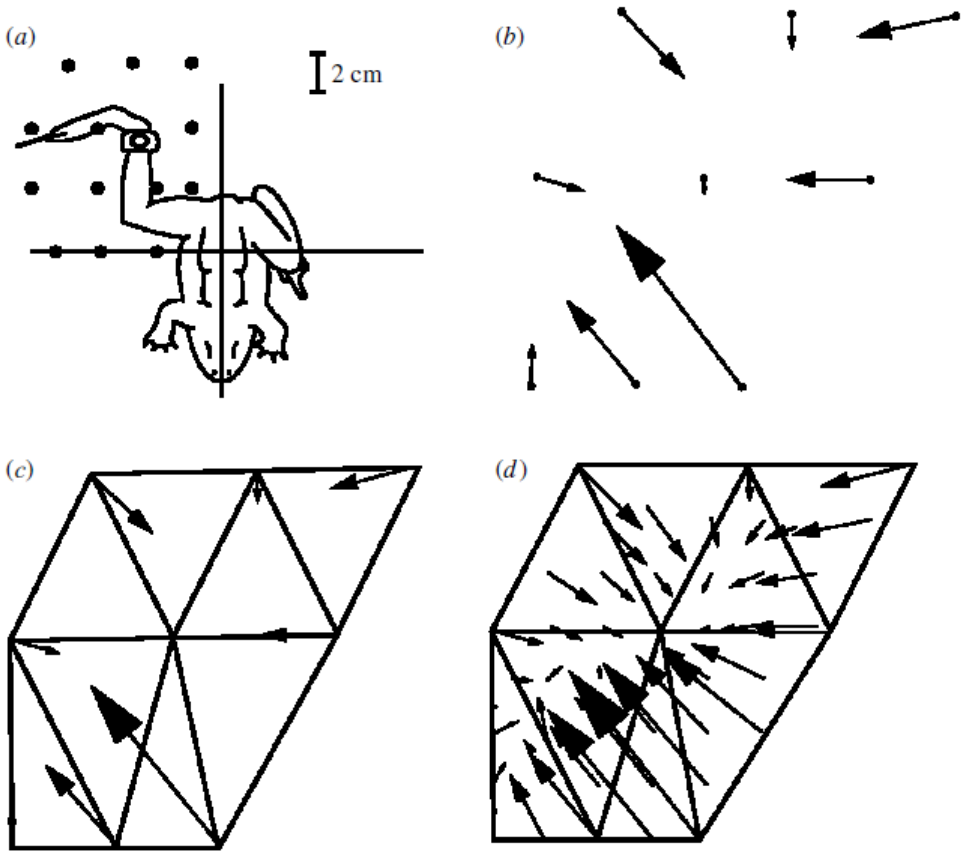

Figure 2.5: Force fields induced by the microstimulation of a frog's spinal cord, taken from [84].

and availability of motor primitives and modulated reflexes reduce the required complexity of the descending command signals and provide a critical foundation for the functional context of higher control centers. It is essential to take them into account when examining the potential biological implementations and plausibility of different motor control models.

2.2 The brain stem

The brain stem is located at the base of the cerebral hemispheres, and is composed of the midbrain, pons, and medulla, seen in Figure 2.6. Evolutionarily, it is one of the oldest parts of the brain, as it is found in even primitive vertebrates. The forebrain of reptiles, amphibians, and fish constitutes a very small portion of the brain, with the vast majority of behaviours being guided by the brain stem. Actions, such as

feeding, drinking, sleep, and emergency responses, are governed by relatively simple, stereotypic motor responses.

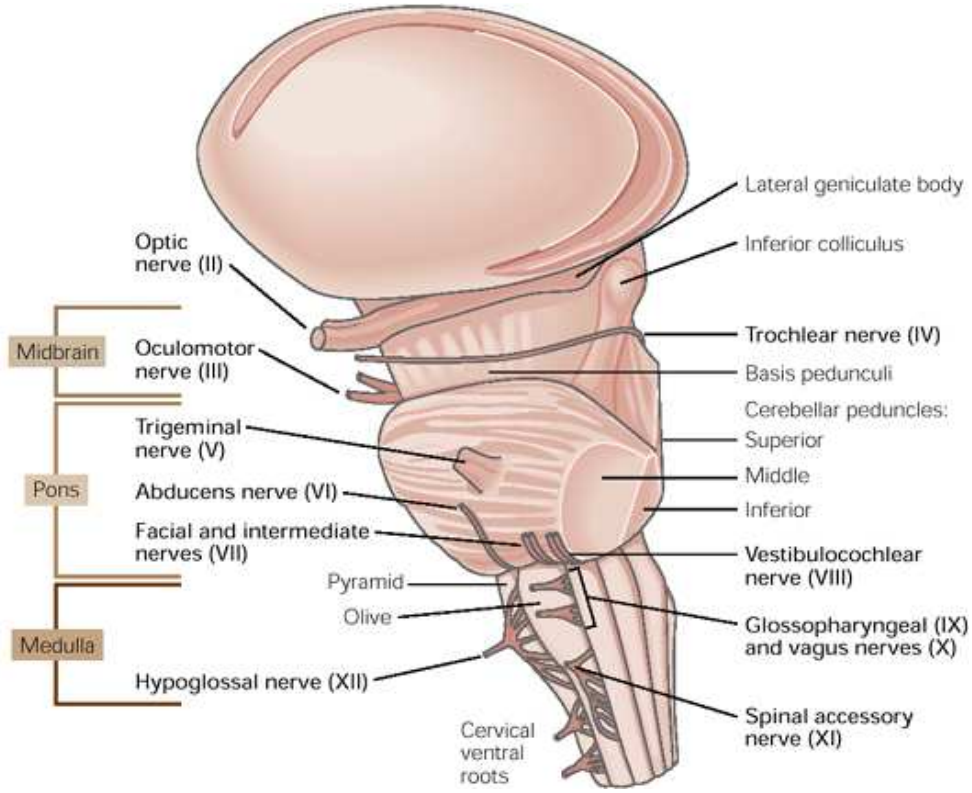

Figure 2.6: The brain stem is composed of three sections, the midbrain, pons, and medulla. Taken from [106].

The actions controlled by the brain stem persist in more complex vertebrates, and indeed in humans it has been found to still be responsible for a myriad of systems fundamental to basic life support. The structure of the brain stem is similar to that of the spinal cord. Many of the brain stem's neural circuits, however, are far more complex than those found in the spinal cord. The reason for this is that the brain stem deals with highly complicated regulatory systems that are fundamental to life, whereas the spinal circuits typically control systems that are not immediately necessary for survival.

The reticular formation is the core of the brain stem, containing interneuron circuits that generate motor patterns and control reflexes for the neck, chest, and most of the abdominal organs. Further, the brain stem is responsible for innervating the muscles of the face and controlling eye movement, among many other things [106].

The brain stem is the gateway between the body and the rest of the brain, carrying nearly all of the information exchanged between them [44]. Because of this, the brain

stem is able to perform a cursory filtering of the sensory feedback from the body's sensory receptors, and relay the information relevant to ongoing motor tasks.

Since the brain stem is mostly concerned with internal system regulation, motor control and sensory information processing of the head, neck, and face areas, it is clustered with the spine in the work presented in this thesis. In a large-scale biological implementation of motor control of the limbs, this abstraction may need to be revisited when the system incorporates multiple limbs. In that case, there is a large amount of sensory information being sent to the forebrain, and an explicit addressing of the brain stem and its filtering capabilities may be necessary. However, since the NOCH model deals with single-arm control in this thesis, this simplification is appropriate.

2.3 The primary motor cortex

The primary motor cortex (M1) is located in the posterior part of the frontal lobe, in Brodmann's area 4, Figure 2.7. It works with the premotor areas to plan and execute motor actions. It was named the 'primary motor cortex' because of the direct connections it has to the spinal cord's motor neurons. There are also secondary and tertiary motor areas, named for their proximity to the motor pathways.

2.3.1 M1 motor map

M1 is one of the most studied areas of the brain, with experiments dating back to 1870 by Gustav Fritsch and Edvard Hitzig, which showed that electrical stimulation of the cerebral cortex of a dog produced movement [34]. It was discovered that the location of stimulation in M1 determined which part of the body responded. In addition, lesions to a section caused the corresponding body part to become paralyzed.

Since these early experiments, a 'motor map' of M1 has been developed, for the purpose of detailing the muscles controlled by various locations in M1 [95]. Importantly, the size of the cortical area that is dedicated to a particular muscle or muscle group relates directly to the amount of fine grained control that M1 has over it. For example, the fingers and hands have a much larger section dedicated to their operation than the back or legs [44]. The initial motor map was refined in 1957, with the introduction of somatotopy, leading to the motor homunculus model of M1, Figure 2.8. Somatotopy is the idea that nearby sections of the body are represented by

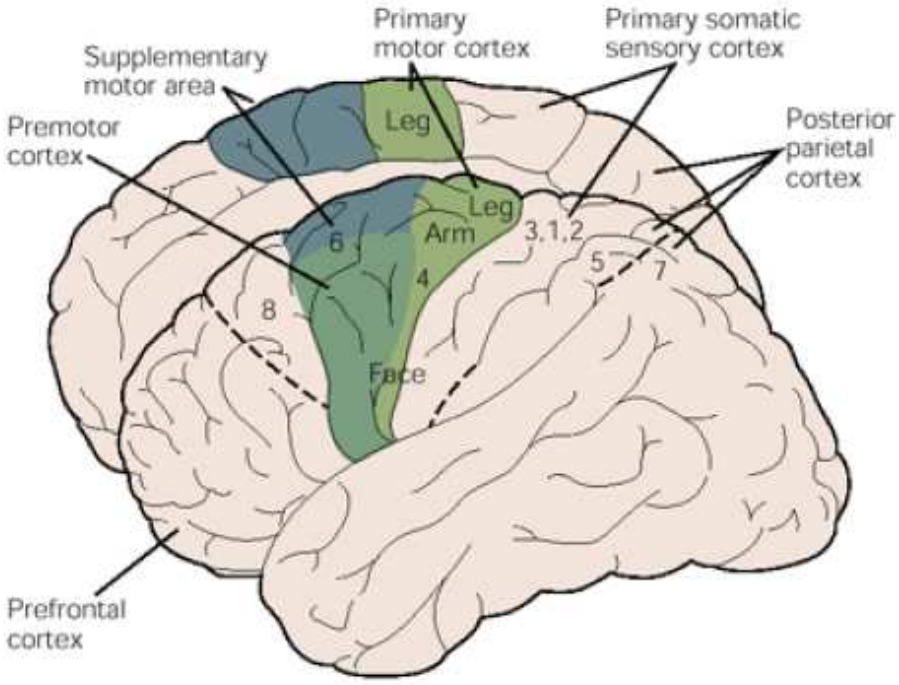

Figure 2.7: The motor cortex. Highlighted are the primary and premotor cortex areas. Taken from [126].

neurons nearby each other in the cortex [96].

The idea of M1 containing a somatotopic respresentation with specific, discrete cortical sites holds great appeal in its simplicity. However, there have been numerous studies in the last decade and a half that show that this is in fact not the case in M1 [11, 19, 20]. For example, it has been shown that finger movement in M1 of monkeys involves an overlapping mosaic of neuronal populations that control movement of each finger with varying degrees of specificity [111]. Noteably, it was also shown that these areas have plastic properties as well. For example, the amount of space in M1 dedicated to finger control in a string musician will be found to be larger than in non-musicians [31]. It has also been shown that the removal of sensory feedback to areas in M1 will trigger a reorganization of the motor map [107]. Together, this evidence shows that somatotopic maps in M1 will vary between individuals, and that specific sites cannot necessarily be pinpointed for the control of a muscle or group of muscles.

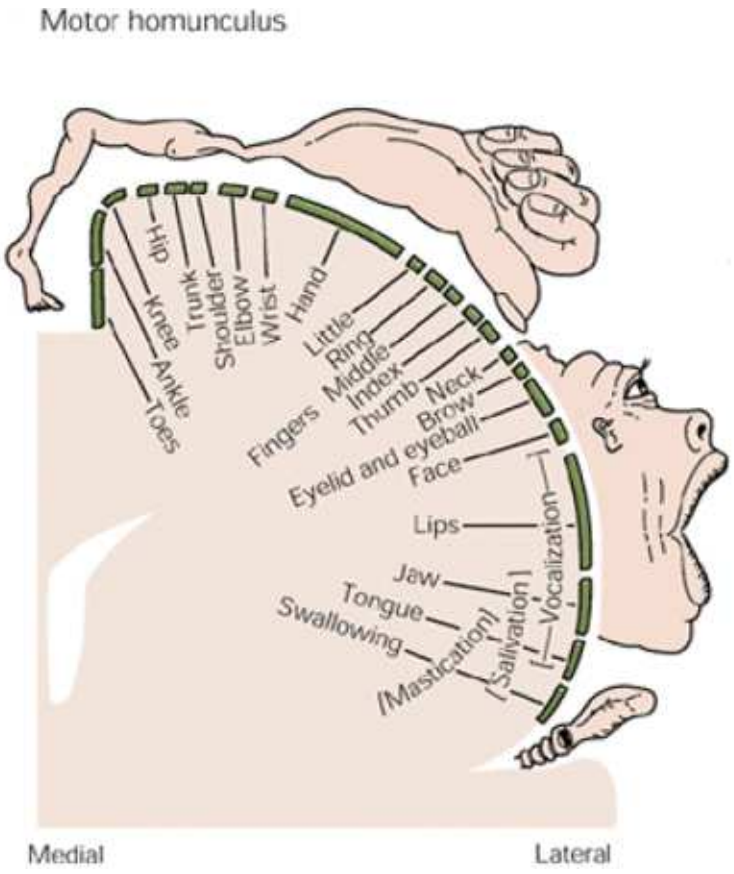

Figure 2.8: The motor homunculus of M1, taken from [44].

2.3.2 What does M1 output?

In 1982 Dr. Apostolos Georgopoulos performed single cell recording experiments on monkeys who were performing arm movements in a two-dimensional apparatus [42]. The monkeys performed arm movements in 8 directions at 45 degrees intervals starting at the same point and with the same amplitude. He discovered groups of neurons that varied in an ordered fashion with the direction of movement. The activity of these cells was the most intense for arm movement in a particular direction, reducing gradually as the arm movement became farther and farther away from the preferred direction. Plotting a neuron's response in terms of firing rate, with respect to movement direction resulted in a bell-shaped tuning curve, where the peak of the curve represented a cell's 'preferred direction', [41], seen in Figure 2.9. The preferred direction of a neuron is thus the movement direction to which it is most responsive.

In a later study of populations of cells in these areas, it was revealed that by taking a summation of the neural activity weighted by the cell's preferred directions, the

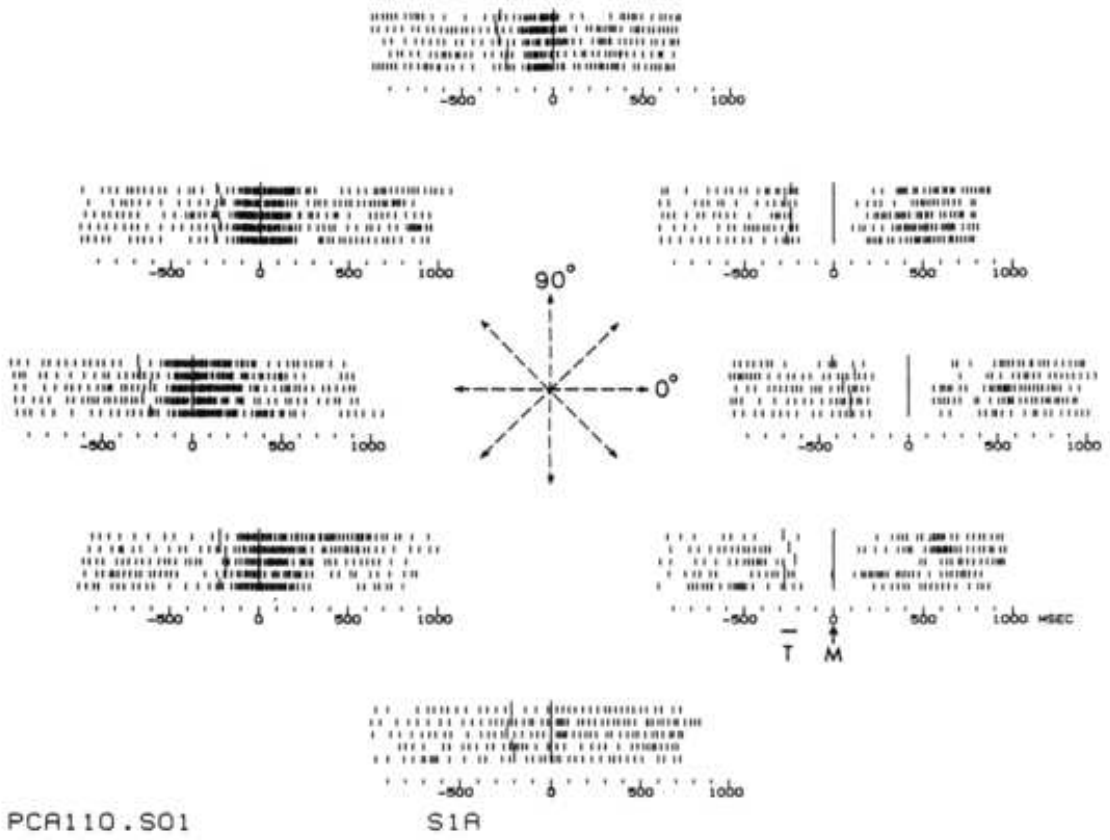

Figure 2.9: The preferred direction firing of a neuron in M1. Taken from [42]

direction of arm movement could be recovered [42]. This lead to theories suggesting the possibility of the motor cortex generating end effector movement command signals which would be sent to the spine, and there converted into muscle activations [43]. These results lead researchers to believe that it was not a low-level muscle activation control signal that was being generated by M1, but rather a more abstract parameter of arm movement.

However, further research also correlated the activity of M1 neurons with arm position, acceleration, movement preparation, target position, distance to target, overall trajectory, muscle coactivation, serial order, visual target position, and joint configuration [129]. Clearly the underlying function behind the output signal of M1 must, in some fashion, have these parameters incorporated.

In fact, if one accounts for the computations necesssary to effect an endpoint force, involving muscle length, rate of change of length, and multi-joint mechanics, these parameters arise as artifacts of the calculation [129]. Emmanuel Todorov (2003) presented a simplified model of M1 that was able to do exactly this [130].

There are several consideration that support this model. First, having the entire supra-spinal system encoding only for direction is an inefficient use of resources. Even with the neural circuitry of the spinal cord designed to simplify the required information in the descending control signal the work left to the archaic spinal circuits is tremendous if all that is given is a directional command. In addition, at several levels of the motor system, feedback is incorporated into the descending control signal, and the process becomes significantly more difficult if the feedback must be first converted into an abstract, high-level parameter of movement such as end effector force. Similarily, CM cells, which are cells that connect directly from the cortex to the motor neurons in the spine, would then require the motor neurons in the spine themselves to interpret the high-level command. This evidence all suggests that the primary motor cortex encodes for muscle activation.

This encoding strategy, however, does not take advantage of the motor primitives available in the spine, which have been shown to be capable of easily generating a vast repetoire of complex movements [84]. Incorporating the use of motor primitives into a model of M1 output is a natural next step in models of motor control, as it allows for the system to utilize the spinal force fields for larger, sweeping movements of the limbs and control muscle activation of specific areas when fine grained control is necessary. Indeed, this is the assumption of M1 output made by the model developed here.

2.3.3 Synergies and expert actions

As mentioned previously, the primary motor cortex has plastic properties and develops finer grained control of movements with practice. Also, as a set of motor actions are practiced, a repetoire of movements is built up that can be called upon quickly and without volitional control.

For example, the beginner piano player must carefully place each finger when first learning chords. However, a more experienced pianist will be able to effortlessly move their hand to the correct location and in the correct position. With further practice, chord progressions and trills will become part of the movements incorporated in the pianist's playing. Clearly there is a skill set being built up, and in the motor cortex elements of such skill sets are called synergies. Expert actions are complex actions, or a sequence of complex actions, that have become a synergy in the primary motor cortex, so they no longer require conscious control.

Motor synergies can be compared to the representations found in the visual cortex. Ascending the hierarchy of visual cortices, V1, V2, V3, V4, the activity of neurons represent increasingly more complex features of the visual world. V1 contains neurons that are sensitive to very basic image features, such as color and line orientation. V2 neurons then react to different combinations of the features encoded by V1, such as a red horizontal line. At the next stage in the hierarchy increasingly complicated shapes and features are represented, and in the inferior temporal cortex, which is one of the highest levels of visual processing, there are neurons selective to face, and other complex stimuli [132].

Similar to the representation of increasingly complicated features in the ascending levels of the visual hierarchy, lower levels of the motor hierarchy would represent basic muscle or muscle group activations, with succesive levels representing increasingly complex combinations of the features of the level below [132]. In this context it is not difficult to imagine the incorporation of motor primitives into the command signal sent out from M1, such that this high level synergy activation would in turn activate the synergies of the level below, until the level of motor primitives had been reached. Combining motor primitives with more specific muscle activation commands would allow for the immense flexibility of a system capable of developing expertise in a staggering amount of complex motor actions and carrying them out automatically.

While the primary motor cortex may be capable of developing synergies through the practice of motor actions, M1 itself is not sufficient to create novel sequences of complex movements. It recieves input from the premotor areas, which have been shown to be involved in the generation of such sequences, and are discussed in the next section.

2.4 Premotor Areas

The premotor areas of the cortex lie in Broadman's Area 6, just rostral to the primary motor cortex, in Area 4 [44]. The premotor areas are subdivided into several sections, the dorsal premotor cortex (PMd), the ventral premotor cortex (PMv), the cingulate motor area (CMA), the supplementary motor area (SMA), and the presupplementary motor area (pre-SMA), seen in Figure 2.10. Each of these areas is heavily interconnected with the rest.

In the 1930's it was discovered that electrical stimulation of the premotor area of

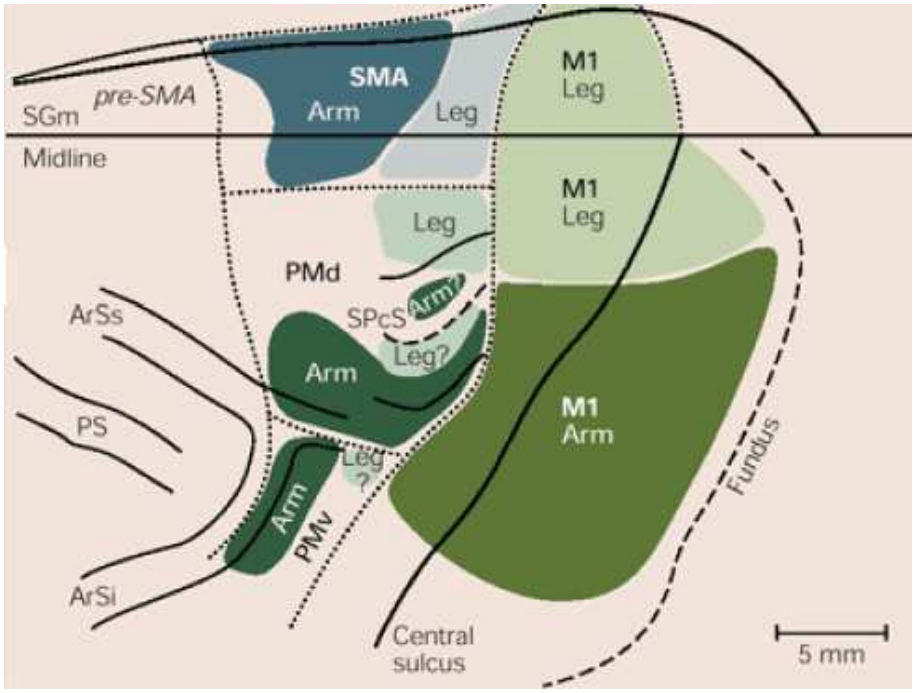

Figure 2.10: The premotor cortex, taken from [126].

the brain resulted in motor actions, similar to M1. However, the required stimulation was larger and the resulting motor actions more complex, such as head turning or torsion of the body [35, 92]. The premotor cortex has been associated with interesting phenomena that have led researchers to believe that this area is responsible for the generation of high level motor plans independent of muscle activation. For example, when a subject is instructed to write with a pen, their writing style is not specific to the hand. Regardless of whether the hand, foot, or mouth was used, an underlying similarity is clear. This is known as motor equivalence [144].

The premotor areas are involved in motor planning, integration of visual and proprioceptive feedback into motor actions, timing of movements, and learning new motor action sequences [44]. In this section the function of each of the premotor areas and how they relate to the overall motor system will be described.

2.4.1 Premotor Cortex

The premotor cortex occupies the more lateral spaces of Broadman's area 6. The premotor cortex areas are involved in motor action planning and the integration of sensory information into motor control. The PMv and PMd have strong connections to areas of the posterior parietal cortex (PPC), parieto-occiptal (PO) pathway, and

prefrontal cortex (PFC), which give the premotor cortex access to proprioceptive information, visual information, and working memory respectively.

Dorsal premotor area

It is known that the PMd is responsible for triggering motor responses based on spatial cues [145], both visual and auditory [142]. Studies of humans and monkey with lesions to the PMd showed that the patients were incapable of forming associations between arbitrary external cues and motor actions [97, 98]. Visual information in particular is associated with the PMd, as it has both direct and indirect connections to the PO areas thought to be involved in transforming visual information into an appropriate reference frame for assimilation with motor actions [38].

These connections to the visual processing areas and the heavy connections to the other premotor areas responsible for motor plan generation suggest that the PMd may serve to correct and refine motor actions based on visual feedback.

Additionally, the PMd has been suggested as playing an important role in the mental rehearsal of motor actions based on data from experiments performed on monkeys. In these experiments the PMd of the monkeys exhibited the same activity pattern both during movement observation, performance, and the instructed-delay period before any motion ocurred [24].

Ventral premotor area

The ventral premotor cortex has been shown to be heavily involved in controlling hand movements during object manipulation [23]. The PMv has strong connections to working memory in PPC area 46, and the anterior intraparietal (AIP) area which is involved in goal representation [139]. The AIP-PMv circuit has been shown to be fundamental to the appropriate control of hand shape for object grasping [103]. Also, the grip strength has been shown to be a function of the firing strength of the neural activity in PMv during precision grip [57].

Although the PM has substantial CM cells, control of the hand from PMv is mediated through the primary motor cortex. This has been shown in experiments where PMv activity has little effect on the CM cells projecting from the PM areas, but greatly affected the output of CM cells in M1 [120].

One of the more curious and interesting features of the premotor areas is also

housed in the ventral premotor cortex:'mirror neurons'. Mirror neurons were first encountered in the 1980's by Giacomo Rizzolatti et al [103]. In this study electrodes were placed in the PMv of a macque monkey and showed that these neurons were activated not only when the monkey was performing an action, but also when the monkey observed another monkey or human performing the action [94]. It wasn't until several years later that the same group performed an explicit experiment on the area looking for these neurons and discovered that approximately 10% of the neurons in the inferior frontal and inferior parietal cortex are mirror neurons as well [36]. Mirror neurons have since been confirmed and greatly studied, and corresponding areas have been identified in humans through neuroimaging studies [52]. The mirror neurons in the PMv have been shown to be most responsive when observing grasping or the manipulation of objects and, importantly, have been shown to be sensitive to a particular action [37].

Mirror neurons are thought to be critical to observational learning, and indeed would provide the ideal training signal for a system attempting to replicate an action or sequence of actions.

2.4.2 Cingulate motor area

The cingulate motor area (CMA) is located in the banks of the cingulate sulcus in the medial surface of the cerebral hemisphere. It is divided into two distinguishable sections, the rostral and caudal, CMAr and CMAc, respectively, and has dense connections the rest of the premotor areas, as well as the basal ganglia, thalamus, and primary motor cortex.

There are many hypotheses about the function of the CMA, but there has been much difficulty in finding convincing support for any one underlying function. Some of the presented theories of function involve reward-based motor selection task information processing [119], monitoring progress of required behavioural tasks [59], and guiding movements visually during execution [105], specifically from the CMAc.

Due to the lack of convincing evidence for any well defined function, however, explicit assumptions about the operation of the CMA are left out of the NOCH framework; it is instead clustered with other premotor areas and left for analysis in the future.

2.4.3 Supplementary and pre-supplementary motor areas

The supplementary motor area (SMA) and pre-supplementary (pre-SMA) are located in the medial aspect of the dorso-medial frontal cortex. The SMA and pre-SMA are involved in internally intitiating movement, motor plan switching, learning new movements and executing movements from memory, they are also involved in guiding the timing of separate features of a motor plan, priming motor actions, and importantly, the supression of unwarranted action execution [44].

Supplementary motor area

The SMA has been shown to be crucial to the spontaneous generation of movement [137, 113]. Monkeys trained to recieve a reward when lifting their arm above a certain height refrained from performing this action after lesions to the SMA. However, with prompting from external cues the action was once again carried out regularly [128]. Human patients with lesions to the area have also been shown to have a marked reduction in the number of spontaneous movements of the contralesional limb [68].

The SMA has also been shown to be important in the inhibition of actions. Patients with lesions to this area show tendencies to manipulate objects presented to them without reason, even when such actions are disruptive to an ongoing motor action [16]. In the most extreme cases, alien limb syndrome is developed, in which the affected limb occasionaly makes semi-purposeful movements, such as grasping nearby objects, seemingly outside of the will of the patient [16]. Notably, the lesions were not restricted solely to one area (as lesions rarely are); surrounding areas were also damaged and could be contributing to this effect. However, in experiments where specific ablation of the SMA was conducted on monkeys, involuntary grasping was also reported [137], supporting the idea that the SMA specifically is involved in inhibition of actions.

External stimuli also trigger priming in the SMA, thought to be the loading of potential manipulation motor plans, part of a phenomenom known as object affordance [53]. Object affordance is the set of possible actions available to an individual using an object, and having the object affordance ready available would facilitate behavioural responses to an object. A further external stimuli or an internally generated signal would then be used to determine which actions of the affordance to execute.

Clearly the SMA plays an integral part in motor action planning, as both external

and internal stimuli trigger activity in the area. The SMA, however, should not be viewed as responsible for storing motor programs. If such were the case, we would expect to see much more devestating effects to motor control in patients with lesions to the SMA. The activity of the SMA does correlate highly with the timing of specific actions in movement sequences. Indeed, in humans with lesions in the SMA show difficulties with sequential movements and sequential performance of multiple movements [125]. These results are consistent with the suggestion that the function of the SMA involves the loading of motor plans and their components.

Pre-supplementary motor area

The pre-SMA is thought to be involved in the learning of new motor sequences. PET scans during experiments where the subject is learning a motor program have shown increases in dopamine in the pre-SMA and basal ganglia [39], which is significant because of dopamine's known involvement to learning facilitation. Additionally, monkeys who had their pre-SMA temporarily paralyzed through muscimol injection were shown to be signifcantly less competent at learning new sequences, but when prompted to perform already learned motor programs, the execution was unaffected [86].

The pre-SMA is also highly correlated with the changing of motor plans [60, 79]. This was tested in an experiment with monkeys carrying out saccades, where the target saccade direction was based on the color of a dot in the center of the screen. The first set of runs established a constant motor pattern of moving in a particular direction, and then the target would deviate unpredictably. During the 'switch' trials the pre-SMA area became active when the motor plan correction occurred, and remained inactive when the monkey disregarded the switch signal and instead performed the previously established eye saccade [79].

In another experiment researchers applied microstimulation to the pre-SMA during the monkey's saccade trials. It was found that when the microstimulation was applied the frequency of incorrect, fast responses declined, and the frequency of slower, correct responses increased. Importantly, the switch related neurons were not 'switch specific', meaning that the neurons were not sensitive to a specific switch in motor plans, but rather to the fact that the motor plan needed to be switched [60].

The pre-SMA was also shown to be highly active during associate learning between a given external stimulus and a motor response, where it modulates the activation of

a motor plan loaded for a particular object [60].

Based on the experimental results described in this section, the premotor areas are taken to be responsible for the planning of sequences of movements at a high level, the coordination of the motor plan features and their timing, learning new movements, and the integration of visual feedback into motor plan correction and refinement.

2.5 The basal ganglia

The basal ganglia are a cluster of four nuclei located at the base of the forebrain. The nuclei of the basal ganglia are the striatum, globus pallidus (or pallidum), the substantia nigra, and the subthalamic nucleus (STN), seen in Figure 2.11. The striatum is further divided into the caudate nucleus, putamen, and ventral striatum. The basal ganglia is strongly connected with virtually the entire cerebral cortex, the hippocampus, amygdala, thalamus, and brain stem, Figure 2.12. Unlike most areas of the motor control system, the basal ganglia do not have direct projections to the spinal cord, instead projecting to the brain stem and thalamus. Through the thalamus, the basal ganglia has indirect connections with the prefrontal cortex and the precentral motor cortex (consisting of the SMA, PM, and M1) [26].

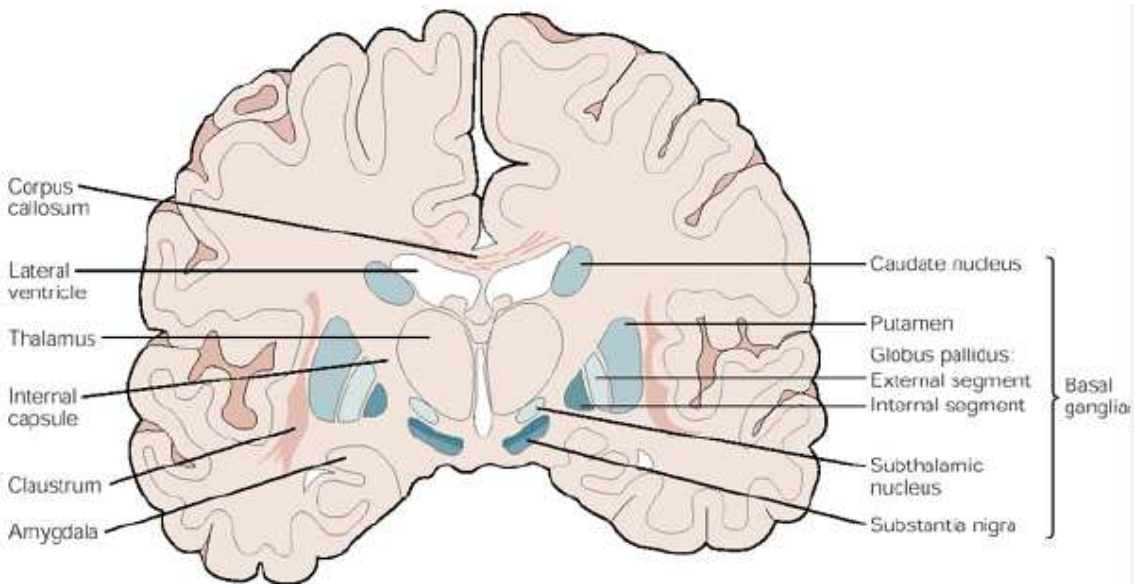

Figure 2.11: The areas of the basal ganglia, taken from [26].

The basal ganglia were identified as being involved in voluntary motor control through examination of the brains of patients who suffered from Parkinson's, and

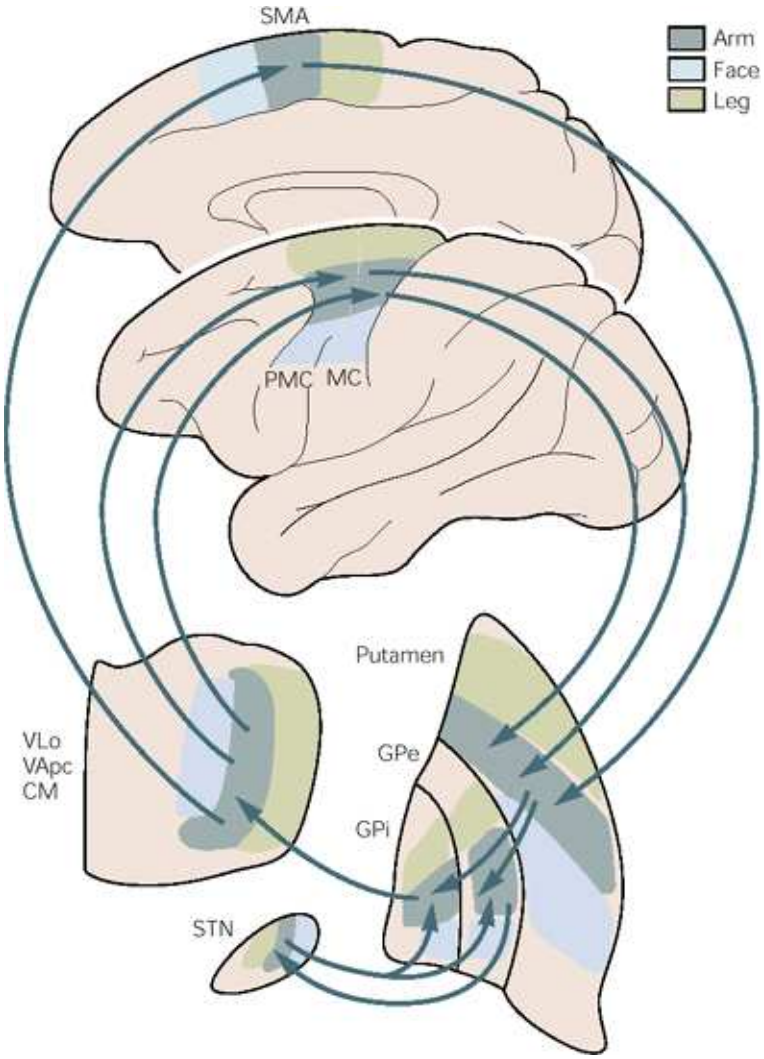

Figure 2.12: Circuits through the basal ganglia, taken from [26].

Huntington's disease [2]. In these patients, this area revealed degeneration of the four nuclei compromising basal ganglia. Since these initial findings, their role as a major component in the control of volitional motor actions has been studied and confirmed [21, 22]. In recent models, the basal ganglia have been assigned tasks such as enabling actions selected by an evaluation of the environmental states [28] and the learning and execution of goal-directed, sequential behaviour, and dimensionality reduction [51, 81, 138].

The striatum is the major input location for the basal ganglia, recieving virtually all of the input to the area. It projects to the globus pallidus and substantia nigra, which form the major output of the basal ganglia. The projections within the basal ganglia are nearly all inhibitory, with the only source of excitatory projections from

the STN. Notably, the external projections to the thalamus and the brain stem are also inhibitory.

There are two pathways for the activity of the striatum to affect the output nuclei, the direct and indirect pathways, seen in Figure 2.13. The direct pathway inhibits the activity of neurons in the output nuclei, which have a very high tonic discharge rate, whereas the indirect pathway increases the activity of these output neurons. Because the output of the basal ganglia inhibits the activity of their targets in the thalamus and brain stem the direct pathway from the striatrum provides a means to uninhibit the neurons in the thalamus and ultimately cause cortical activation, facilitating movement. Conversely, the indirect pathway causes further suppression of the neurons targeted by basal ganglia output, thereby inhibiting movement [143].

The basal ganglia are integral to the functon of a family of circuits linking the thalamus and cerebral cortex. Largely segregated both structurely and functionally, each circuit has a unique origin in the cerebral cortex, traverses a distinct path through the basal ganglia and thalamus, and then returns to the same area of cerebral cortex. These circuits are the skeletomotor, oculomotor, prefrontal, and limbic circuits [89]. Importantly, the circuits have essentially no communication with each other, aside from a point in the substantia nigra pars reticula where there is physiological evidence suggesting that crosstalk may occur [3]. There is also evidence that inside the skeletomotor circuit the PM, SMA, and M1 operate with their own independent subcircuits [3].

The circuit of most relevance to this thesis is the skeletomotor circuit, which begins and ends in the precentral motor fields. Importantly though, this does not mean that the other circuits are to be ignored. It is clear from the structure of these circuits that a specific processing function is occuring in the basal ganglia-thalamus loop, and information about the underlying computations for any of the circuits will reveal insight into the function of the others.

The input to the basal ganglia for the skeletomotor circuit is by and large the putamen. It recieves information from both the precentral motor fields and postcentral somatosensory areas and is therefore thought to be an important site for the integration of movement and movement related sensory feedback information [26].

Through recordings of single-cell activity of neurons in the skeletomotor circuit of monkeys, it has been shown that during the execution of a specific motor action the neural activity of the output nuclei in the basal ganglia increases in the majority of cells, but is decreased in a select few [58]. If the selected motor plan is sent from

the putamen along both the direct and indirect pathways, then the indirect pathway could be used to brake or smooth out motor actions, while the direct pathway would be used to simultaneously facilitate the movement, creating a circuit useful for scaling. If however, the putamen were to send information relating to the selected motor plan along the direct pathway, and the competing motor plans along the indirect pathway for suppression then a type of selection would be implemented [26].

There is also evidence from the damage done by Parkinson's or Huntington disease to the indirect pathway from the striatum to the output nuclei suggesting that this pathway is critical to the suppression of unwanted motor plans. Heightened activity here leads to a poverty of motor action, as seen in Parkinson's, and lessened activity results in unwanted actions, as seen in Huntington disease [2].

Given the information described above and considered in the context of the motor system as a whole, it seems likely that both scaling and selection are implemented through the direct and indirect pathways. In general, the direct pathway from the striatum to the output nuclei would be utilized for the selection and scaling of the chosen motor plan, while the indirect pathway supresses the competing motor plans. Organization along these lines would then also be useful for learning and fine-grained control. The indirect pathway could reduce the inhibition of specific competing motor plans, implementing a kind of selection in the indirect pathway as well; those neurons that were suppressed to a lesser degree would send out a signal that could be amplified later on. This would allow the incorporation of those additional motor plans into the control information at later stages of processing. In this way the system could experiment with combinations of plans and actions, and be used to create a large number of novel movements.

It has been experimentally shown that inside the skeletomotor circuit of a monkey, the activity of a substantial proportion of the movement related neurons depend on the limb movement direction, independent of the muscle activation pattern. Furthermore, in trained primates, the activity of arm related neurons in the pallidum is highly correlated with the amplitude and veloctiy of a movement [26]. There has also been a distinction between subcircuits active during preparation for and execution of motor actions discovered in experiments showing neurons responding strongly during one stage or the other [82, 116]. The experiments showing results of neural activity pertaining to movement direction is important because it lends credibility to the idea of a high level controller through the skeletomotor circuit working in a state space more abstract than muscle activation, such as joint torque or end-effector force. Sep-

aration of the tracts is also a key property because it allows the separate areas to use the basal ganglia in a similar fashion for different processes.

In the NOCH framework, the skeletomotor circuit will be separated functionally into subcircuits. The circuit through the SMA will not be explicitly represented, because of its seemingly close ties with manipulating objects using the hand, which is not represented in any of the simulations performed here. The circuits through M1 will be used during motor action execution and control of expert actions. The circuit through the PM will be used in optimal action selection at an abstract, higher level of body representation, and for the scaling of movement amplitude and velocity.

2.6 The cerebellum

The cerebellum is located just above the brain stem, in the lower dorsal section of the brain. The neural structure of the cerebellum is unique, as it is arranged in an unusually highly organized, repeating sequence [45]. The cerebellum is also incredibly dense. Although it constitutes only 10% of the total volume of the brain, more than half of the brain's neurons are located here. In the cerebellum there are three distinct sections: the central vermis and the intermediate and lateral zones of the hemispheres on either side of the vermis, seen in Figure 2.14. Each of these regions recieves input from separate areas of the brain and spinal cord and projects to different motor system areas.

The cerebellum is composed of an outer layer of gray matter, known as the cerebellar cortex, internal white matter, and three pairs of deep nuclei: the fastigial, interposed, and dentate [45]. The cerebellum has an overwhelming number of input connections, on the order of 40 times more than the number of output projections. These projections are mainly to the premotor and motor systems in the cerebral cortex and brain stem, and originate almost entirely in the deep nuclei. It is thought that the cerebellum serves to store internal models used by the motor system for generating motor actions appropriate for a given task, and to measures the discrepancy between the intended and executed motor actions, from which it generates a corrective signal [45].

In experiments performed on animals with lascerated cerebellums, neither sensory thresholds nor the strength of muscle contractions are affected, implying that the cerebellum is not fundamental to perception or movement [17, 18]. Damage to the

cerebellum does however impair balance, reduce muscle tone, and disrupt the spatial accuracy and temporal coordination of movements [80]. It also significantly affects motor learning and certain cognitive functions [147]. If the cerebellum is indeed a site for storing internal models it would be expected that motor learning would be devestated by damage to the area. While errors in movement could be corrected through the slower visual feedback loop, it would be expected that large errors would be found in movements necessitating feedforward control, such as bracing one's self to catch a heavy object. Indeed this is the case, as found in experiments involving patients with damage to the cerebellum [13]. This research also indicates that the cerebellum is responsible for much of learning through trial and error, and involved in correction and refinement of skills through repetitive practice [48, 99].

As previously mentioned, the sections of the cerebellum are connected to different areas and involved in distinct processes. The input to the vermis consists of visual, auditory, and vestibular information in addition to somatic sensory input from the head and proximal parts of the body [45]. It is involved in the control of posture, locomotion, and gaze by projecting through the fastigial nucleus to the cortical and brain stem regions that generate the descending motor commands that control the proximal muscle of the body and limbs.

The intermediate zones of the hemispheres recieve somatosensory input from the limbs and project through the interposed nucleus to the lateral corticospinal and rubrospinal systems, which control the more distal muscles of the limbs and digits. Together, the vermis and intermediate hemisphere areas are referred to as the spinocerebellum because they recieve somatosensory input from the spinal cord.

The lateral portions of the hemispheres make up the cerebrocerebellum, named for their heavy connections with the cerebral cortex. These areas project through the dentate nucleus to the motor, premotor, and prefrontal cortices.

In the spinocerebellum there are two inverted somatosensory maps (where the location of body part representation is mirrored in the opposite section), and the deep cerebellar nuclei have also been found to be organized somatotopically. These deep nuclei are tonically active, and it has been found that when damaged, the timing and precision of movements are compromised [45]. It has been proposed that this results from defective anticipatory control, where the information accounting for body intertia or appropriate actions for task accomplishment has been jeopardized [45].

The intermediate and lateral zones are of particular interest to this thesis. More specifically the function of the lateral zones is a focus, as imaging data has shown

that the cerebrocerebellum is highly involved in the planning and mental rehearsal of complex motor actions and the conscious assesment of movement errors [45]. The ventral and dorsal spinocerebellar tracts provide information about the descending motor commands from other parts of the motor control system [76]. These tracts set up a feedback loop for the cerebellum to monitor the results of signals sent to the motor and premotor areas, ideal for trial and error refinement of internal models [63]. The discrepancy between the intended and the sensory feedback from resulting motor actions is also thought to be used to generate corrective motor commands which are sent from the cerebellum to the brain stem for incorporation into ongoing movements, resolving any undesesirable deviations caused by erroneous internal models [45].

The plausibility of a biological implementation of such a learning system has been addressed by a number of models based on the neural structure of the cerebellum [29, 63, 146], and there is significant evidence for neurophysiological based internal models [47, 62, 66, 67, 118, 127].

When considering the storage of internal models, it is obviously necessary to take into account the structure of the cerebellum. As previously mentioned, the cerebellum is unique in that it has a highly uniform anatomical organization [73]. Additionally the basal ganglia is known to be involved in the scaling of movements. It seems possible then that the internal models in the cerebellum are stored in a normalized fashion, to be given scale by sensory input regarding the operating space and amplitude by determined movement strength and velocity at a later stage.

This type of storage clearly has several notable advantages over explicit storage of every internal model used. First and foremost is its adaptability, which is a principle that is observed throughout the brain. Having internal models that are transformable significantly increases the number of situations that one model can be used for, and also reduces the neuronal resources required to store this information. Scalable internal models also leads to the idea that multiple models could be combined linearly, which again vastly increases the representational power afforded by a small number of internal models.

Of course, this type of manipulation would not be appropriate for all internal models. This work makes the assumption that there are various levels of representation of movement accounted for in the cerebellar internal models. The manipulative operations are more applicable to higher level representations of movement, such as end-effector position in three-dimensional space and force control than to internal models of muscle activation.

This evidence all provides a convincing case for the role of the cerebellum as a locus for storing internal models, as well as its involvement in motor learning and monitoring of descending motor commands. In the NOCH framework, the cerebellum is assumed to perform exactly these functions for the motor control system.

2.7 The thalamus

The thalamus is located in the center of the brain, situated between the cerebral cortex and midbrain. The thalamus acts as a gateway for nearly all cortical input, and is often thought of as a relay station for information from the rest of the brain and spinal cord.

Little is known about the functions performed by the thalamus, due to its highly complex structure and the substantial and diverse connectivity with other areas of the brain. However, it is clear from single cell recordings and pathway tracing studies that the different nuclei of the thalamus deal with information from different areas of the brain. Pertaining to the motor control system is a group of nuclei in the ventral thalamus known as the thalamic motor nuclei: the ventrolateral (VL) nucleus, ventral anterior (VA) nucleus, and ventral medial (VM) nucleus [44]. These nuclei recieve projections from deep cerebellar nuclei, the vestibular nuclei, and extrapyramidal system (striopallidum and substantia nigra) [26].

The VL in humans is further divided into the ventral oral anterior (VOa) and posterior (VOp) nuclei, and the ventral intermediate nucleus (Vim) [56]. The inputs to these areas are formed by special pathways that are topographically organized. The rostral regions of VL predominantly pallidal projections, where the caudal areas receive input from the deep cerebellar nuclei. Additionally, there are nigrothalamic (from the substantia nigra) projections that terminate in the ventral portions of rostral and caudal VL.

There is much support for the hypothesis that inside the thalamus there is very little interaction between the information from the basal ganglia and the cerebellum [8, 7, 61, 75, 87, 109]. Recently there has been an increasing amount of single cell recording and pathway tracing research examining the timing of neural activity and the flow of information to and from the thalamus in hopes of teasing apart the function of the thalamus from those with which it connects [10, 27, 122]. However, the possible functions of the thalamus, independent of information routing, is still very

much a mystery. For this reason the thalamus is presumed to be a relay station in the work presented in this thesis, left open for future development when more conclusive deductions may be drawn.

2.8 Conclusions

This chapter has discussed the neuroanatomy of the areas of the forebrain involved in the motor control system, discussing current models and new ideas of function, and reviewed the structure of the brain stem and neural circuits in the spinal cord. These considerations are summarized in Figure 4.1.

The next chapter will detail the mathematics behind hierarchical optimal motor control models, and the following chapter will bring together the two subjects by presenting a biologically plausible model of motor control based on hierarchical optimal control theory. The assigned function of each of the areas discussed in this section will be explained and the information flow through the system giving rise to hierarchical optimal control of the motor system will be presented.

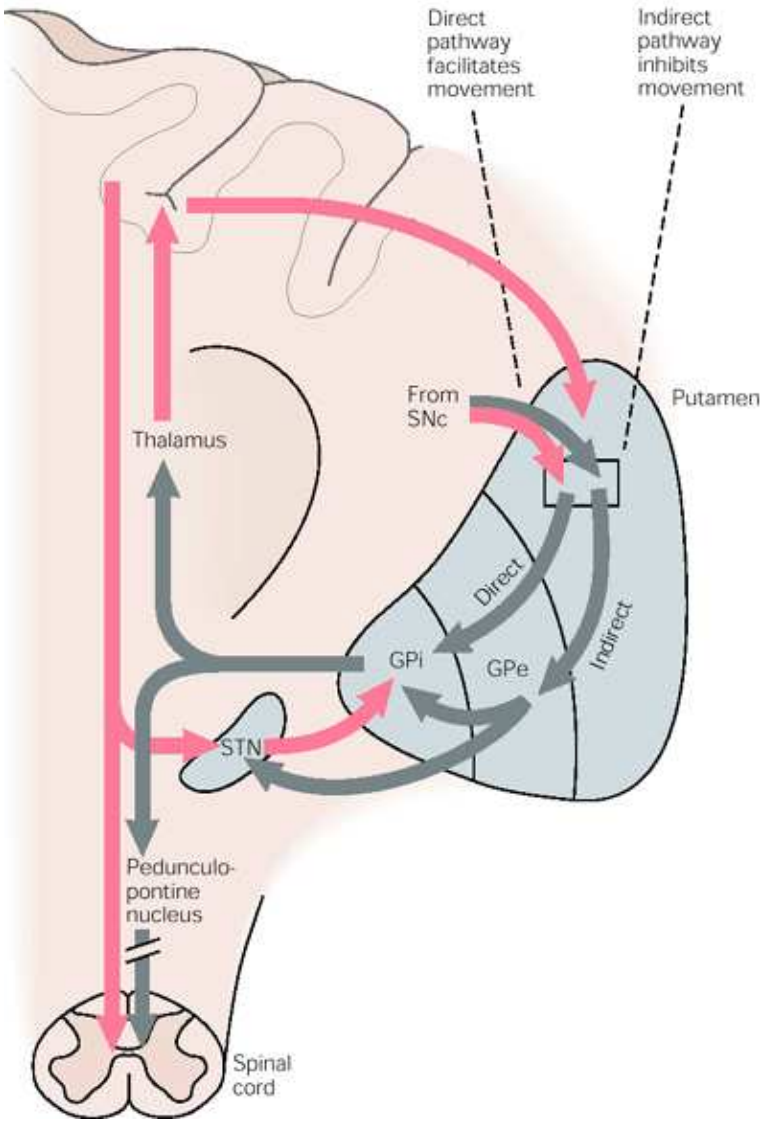

Figure 2.13: The direct and indirect pathways through the basal ganglia, taken from [26].

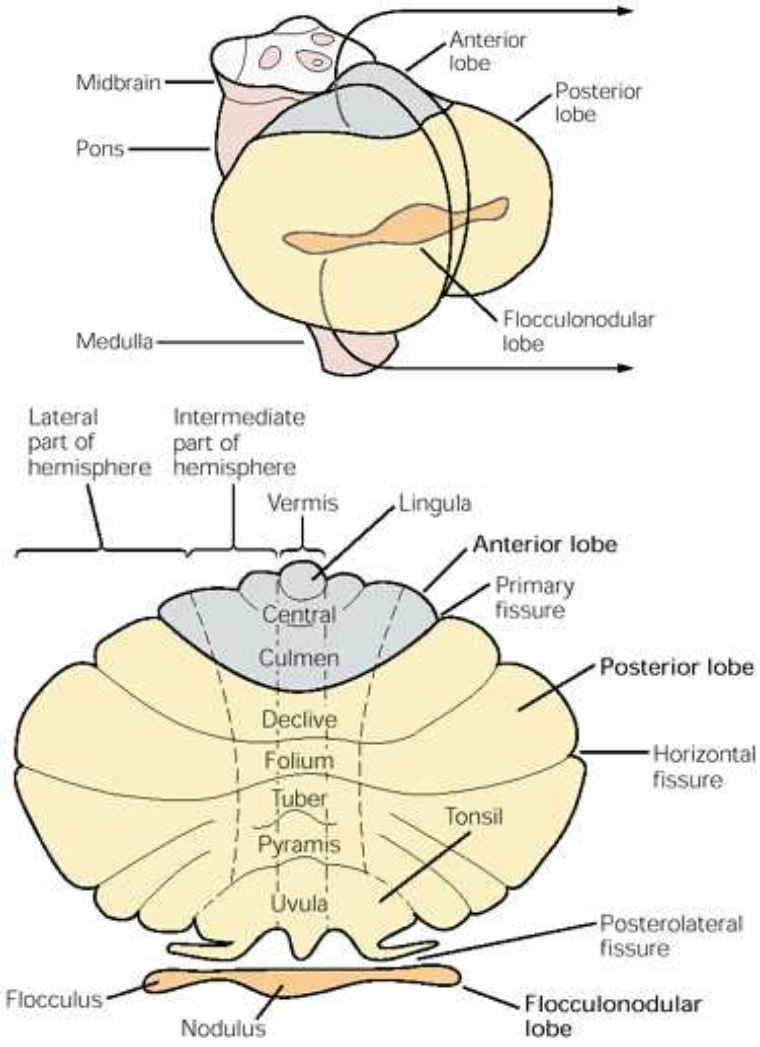

Figure 2.14: The cerebellum, taken from [126].

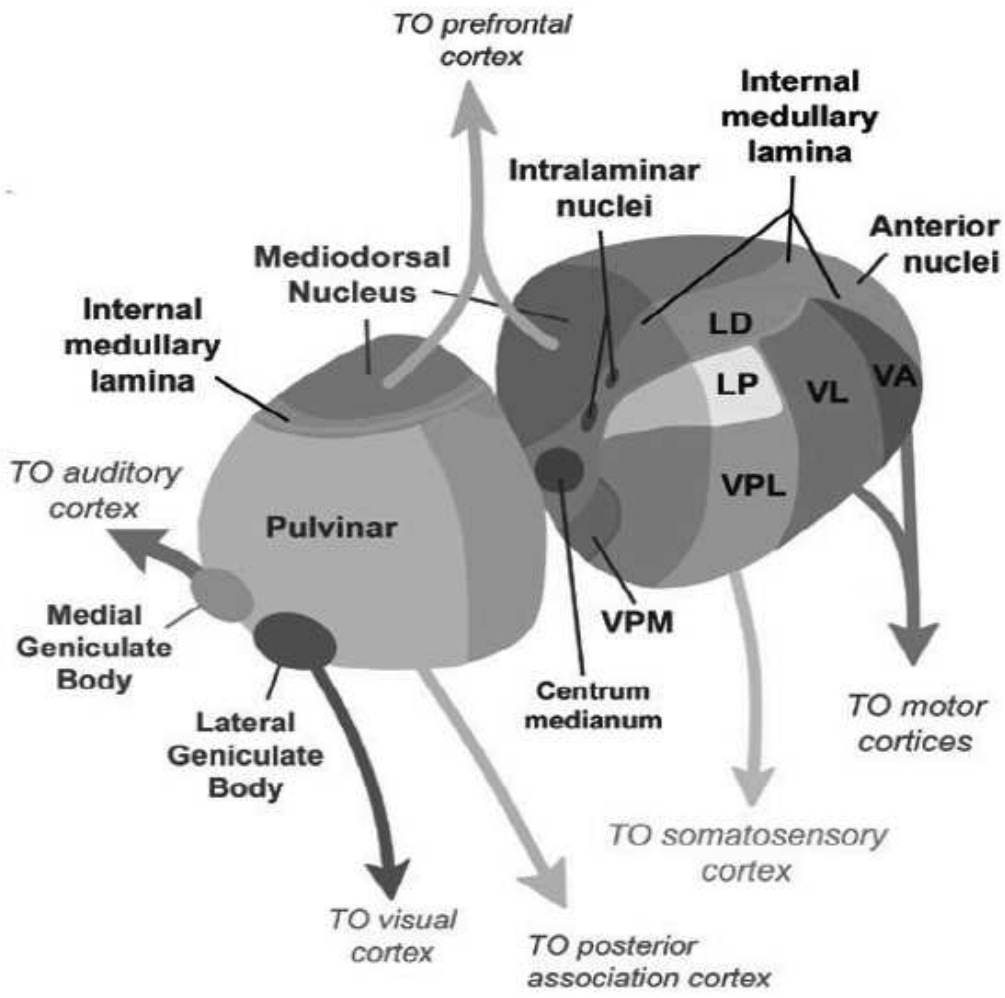

Figure 2.15: The thalamus, taken from [90].

Chapter 3

Control theory

3.1 Introduction

In this chapter different optimal controllers for a two-link arm in a horizontal plane will be examined. The goal of these controllers is to move the end effector from its starting point to the target location while minimizing required time and control energy, as specified by a cost function. However, determining the optimal path to the target will prove to be a difficult task; greedy optimization of the immediate action does not work, because the path with a cheap immediate action cost may incur very large costs later, so all future costs must be taken into account at the current step as well. This can become an expensive computation because the number of possible future states grows exponentially with time.

This problem is summarized in the Bellman equation, which is fundamental to optimal control theory. The Bellman equation defines an optimal 'cost to go' function, which gives the cost to move from any given state to the target optimally. The cost-to-go function is critical because it allows the greedy algorithm to compute the optimal action from any state. However, the Bellman equation only defines the cost-to-go function as the solution to an unsolved optimization problem. There is no generic, all-inclusive definition of the optimal cost-to-go function for every system. In this chapter different approaches to defining the cost-to-go function will be discussed.

Before system optimization is discussed, however, a brief review of control theory will be given to define some basic notation and ideas.

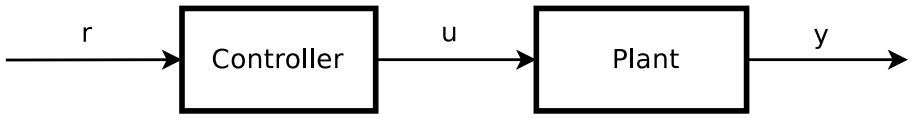

Figure 3.1: Open loop system block diagram.

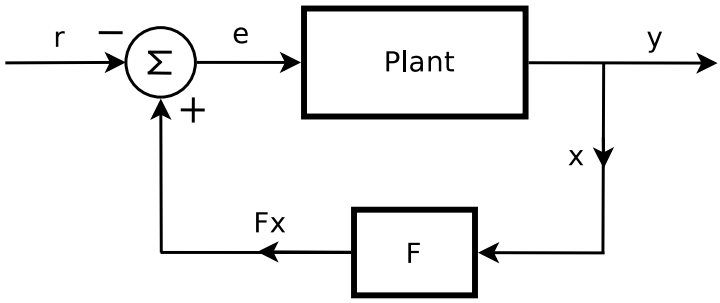

Figure 3.2: Closed loop system block diagram.

3.2 Background

There are two basic categories that control systems can be divided into: open loop (feedforward) and closed loop (feedback), shown in Figures 3.1 and 3.2. Open loop, or feedforward, systems are those that work by predicting the necessary control signal and then executing it without any monitoring of the output. An example of a feedforward controller would be a person who gets a glimpse of a dartboard 10 feet away from him and then must throw all of his darts in the dark. The motor plan must be generated before throwing, and then no sensory information available for adjusting the motor plan.

In the open loop block diagram there is the input signal **r**, the controller, the control signal **u**, the plant, and the output signal **y**. The box labelled 'plant' represents the simulated system that is being controlled. For example, in the cruise control system of a car, **r** is the desired speed, **u** is the amount of gas or brake to apply, the plant is the car travelling along the road, and **y** is the speed reading on the odometer.

A closed loop, or feedback, system, on the other hand is a system that has sensors that can convey the system output back to the controller, which can then correct its signal appropriately. An example of a feedback system would be the same person in a lighted room throwing darts. After generating an initial motor plan and throwing the first dart, they are able to update their arm movement accordingly based on where the dart hit.

In the closed loop system block diagram there is now a feedback gain matrix **F** in

lieu of a controller, the system state vector $\mathbf{x}$, a summation point where the feedback $\mathbf{Fx}$ is subtracted from the input signal $\mathbf{r}$, and the resulting error term $\mathbf{e}$ which is fed into the controller.

The motor control system is typically a feedback, closed loop system. When sensory feedback is available it is used to adjust our movements appropriately and correct any deviations from the specified motor plan. It does also, however, have an accurate feedforward signal, controlling our motor actions before any sensory feedback has reached the brain. The rest of this chapter is focused on building up a controller that accurately models the motor control system in the brain.

Begining with very simplistic controllers based on linear models of the system dynamics, the first section discusses the process of controller design. Given the evidence for a hierarchical system in the brain (see Chapter 2 and [65, 74]), the following section focuses on building up a hierarchical representation of the system. This section will demonstrate the framework used to build hiearchical control systems, and also briefly discuss the nonlinear systems control optimization technique, iLQG [71], which is used to optimize the control of the highest level of the hiearchy. The iLQG technique is also presented to be used as a point of comparison for the novel optimal control algorithm presented in Todorov's (2009) 'Efficient computation of optimal actions', whose implementation in the hierarchy is detailed in the third section. This optimization technique is used as a basis for the high level optimization of the model developed in this thesis.

3.3 Optimal control of linear systems

To begin designing an optimal controller, two things will be necessary: a system model, and a cost function. The system model that will be used in this section is a linear model, the accompanying cost function will be quadratic. The section will go through the derivation of each, how to design a control signal to stabilize the system for any input, and then create a controller that tracks step input to the system.

3.3.1 The state model

A state model is defined by the system state vector $\mathbf{x}$, the system output vector $\mathbf{y}$ and the next step dynamics $\dot{\mathbf{x}}$. For the model in this section they will be defined as

follows:

$$\mathbf{x} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix},$$

where $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ represents the shoulder and elbow joint angles, and $\dot{\theta} = \begin{bmatrix} \dot{\theta_1} \\ \dot{\theta_2} \end{bmatrix}$ represents the velocities at those joints. The next state dynamics $\dot{\mathbf{x}}$ and output $\mathbf{y}$ are defined by the state matrix $\mathbf{A}$ and control matrix $\mathbf{B}$, in the following equations:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{3.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{3.2}$$

Breaking down the right hand side of equation (3.1), the first term represents the passive dynamics of the system, which is the behaviour of the system at the next time step without any control signal. The second term on the right hand side is the control-dependent dynamics, which is the contribution of the control signal $\mathbf{u}$ to the next state. Let the input signal $\mathbf{u}$ be the torque applied to each joint, $\mathbf{u} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$. The change in angular position θ across one timestep is the angular velocity $\dot{\theta}$, which is found in the current state $\mathbf{x}$, so all that remains to be defined is the angular acceleration and our system model is complete. Using the rotational equivalent of force equals mass times acceleration, it is possible to define the state change of $\dot{\theta}$. The angular acceleration of the system is $\ddot{\theta} = \frac{\tau}{L^2 m}$ where m and L are the arm segment mass and length. Writing out the equations for each of the next-state vector variables and putting them into a linear system of equations gives

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta_1} \\ \dot{\theta_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L^2 m} & 0 \\ 0 & \frac{1}{L^2 m} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}. \tag{3.3}$$

The observable output of the system is going to be the joint angles θ, so $\mathbf{y}$ is defined

explicitly as

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \mathbf{C}\mathbf{x}, \tag{3.4}$$

which completes the definition of this system model for the planar two-link arm.

3.3.2 The cost function

The cost function that for this model is in quadratic form. Having the cost function defined as a quadratic gives us an explicit minimum, allowing the optimal solution to be found analytically. The cost function designed here will be an infinite-horizon cost function, which means that no time limit is specified.

Infinite-horizon cost functions consist of two weighted terms, the state cost over time, and the control cost over time. The significance of these terms can be well demonstrated by considering them in the context of an arm controller. When a target has been identified for the arm to reach to, there are an infinite number of potential paths to take to reach it. The cost function is used to value certain paths over the others. If the timer to a nuclear bomb has just reached 1 second, and there is a big red button within reach to disarm it, then the important issue in the arm control is speed. How much energy it requires is irrelevant, so the most direct path will be taken at top speed to hit the button in time and save the world. This is an example of state cost being weighted relatively much heavier than control cost. However, imagining the situation where there is ample time and hundreds of buttons that must be pressed, now the arm movements will be much slower and energy efficient. This is an example of the control cost being relatively heavier than the state cost. For any situation the cost function's weightings determine the compromise between state and control costs that must be realized to achieve optimal performance.

Formally, the infinite-horizon cost function is written

$$\mathcal{L} = \frac{1}{2} \int (\mathbf{x}^\intercal(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\intercal(t)\mathbf{R}\mathbf{u}(t))\, dt, \tag{3.5}$$

where $\mathcal{L}$ is the cost function, and $\mathbf{Q}$, and $\mathbf{R}$ are weighting matrices. The reason

for using weighting matrices instead of scalars is that more precise control of cost assignment is possible with matrices. Here $\mathbf{Q}$ and $\mathbf{R}$ will be diagonal matrices and act as scalar weights, but it should be noted that they are capable of individually weighting each state and control parameter.

This leads to a natural question: how are $\mathbf{Q}$ and $\mathbf{R}$ chosen? It is one of the unfortunate truths of control theory that there exists no standard procedure for choosing weight matrices that achieve specific results. Rather, determining an effective cost function is heavily based on trial-and-error, and experience. However a typical default value for the weight matrices is the identity matrix, and so they are defined as such here.

3.3.3 Stabilizing linear systems

On the path to designing a controller that can take our system wherever our heart desires (or wherever the hand is reaching, as it were) we will need to stabilize the system. A stable system is one whose output stays bounded for any given bounded input.

The physically unconstrained arm model developed in the previous section is a good example of an unstable system. Examining the system model defined in equation (3.3), it is easy to see that this system is unstable. If a constant control signal is applied, the velocity of the end effector will continue to increase, moving it faster and faster in the specified direction. This is apparent in Figure 3.3.3, which is the system output from simulating 10 seconds of operation with constant input using MATLAB's `lsim` function.

Mathematically, this system is unstable because its poles, which are the eigenvalues of the state matrix $\mathbf{A}$, are not exclusively negative (see [140] for details). To fix this problem the system structure must be amended. Currently the system is feedforward, and only the control signal $\mathbf{u}$ is manipulable. To make the system stable, feedback must be added, as seen in Figure 3.2. When feedback is added to the system, the next state dynamics are changed to incorporate the new information signal, $\mathbf{Fx}$. The next state dynamics are now defined as

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B}(\mathbf{Fx} + \mathbf{u}),$$

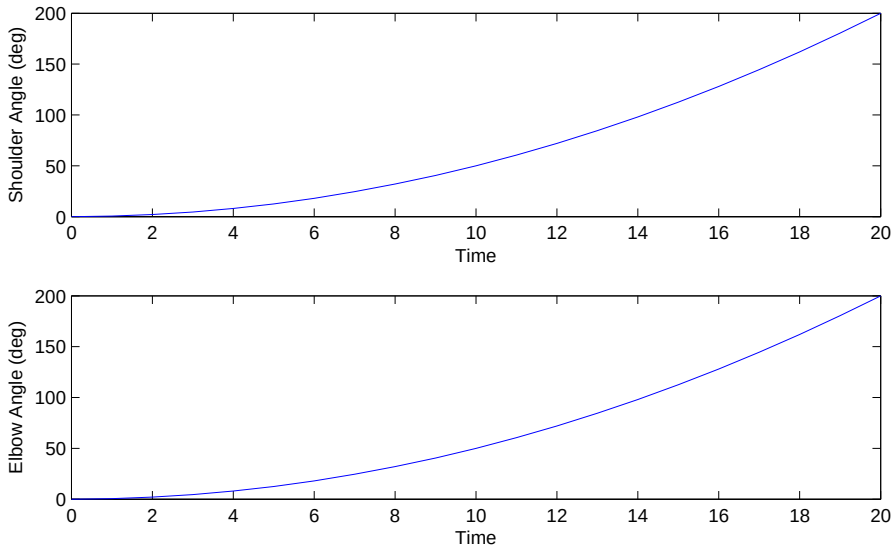

Figure 3.3: Output from an unstable system.

which can be rewritten

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{BF})\mathbf{x} + \mathbf{Bu}.$$

It can now be seen that by adding feedback the designer can affect the state matrix, ($\mathbf{A}$ + $\mathbf{BF}$) through the gain matrix $\mathbf{F}$. With careful selection of $\mathbf{F}$ the placement of the system poles can be controlled so that all the eigenvalues have negative real values (see [140] for details). Because the state matrix is directly responsible for how the system moves through time, when designing an optimal system the poles must be determined using the cost function's $\mathbf{Q}$ and $\mathbf{R}$ matrices. In MATLAB function, the `lqr` function performs this design step by taking the system model and cost function as input and provides the optimal feedback gain.

As mentioned before, exactly how the system is brought to equilibrium is determined by the cost function's weight matrices, $\mathbf{Q}$ and $\mathbf{R}$. Figure 3.4 shows the performance of the system under different cost functions. As expected, when the state matrix $\mathbf{Q}$ is weighted more the system arrives at equilibrium faster, and when the control matrix $\mathbf{R}$ is weighted more the system is driven to equilibrium with much less energy.

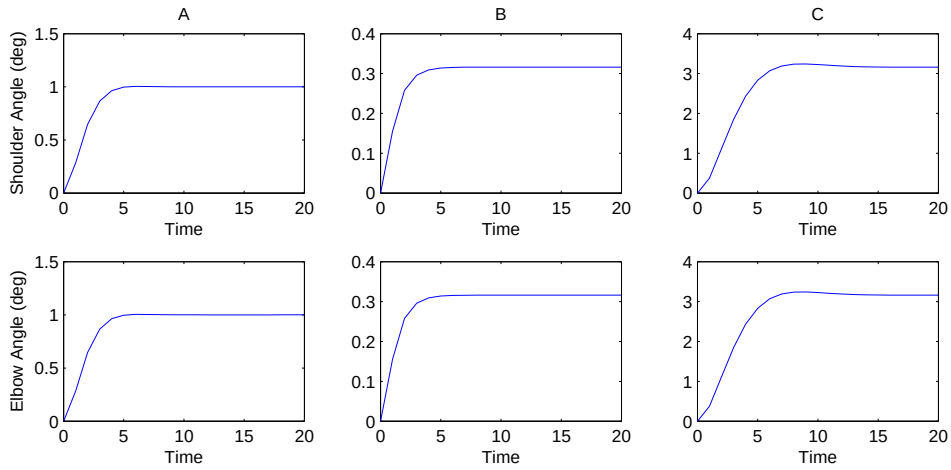

Figure 3.4: In these graphs the x axis is time and the y axis is position, the graphs on top are for θ_1 and the graphs on bottom are for θ_2. A)The performance of the system with **Q** and **R** set to the identity matrix. B)Here **Q** has been weighted 10 times more than **R**. The system quickly converges to a stable state. C)Here **R** is 10 times heavier than **Q**. The system moves with much less energy to equilibrium, which takes a longer time.

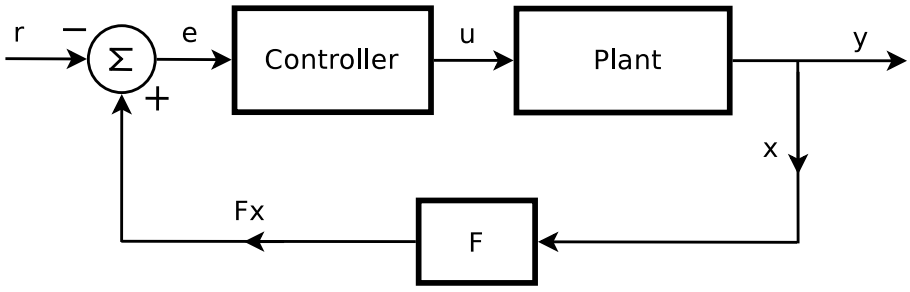

Figure 3.5: Closed loop system with gain and a controller

3.3.4 Tracking reference input in linear systems

During the stabilization of the system the input signal was not of particular concern. The goal was only to ensure that for any bounded input the system would be driven to an equilibrium point in an optimal fashion. However, of interest to the work done here is a system that can track the input to the system, such that the system output matches the input. This requires a reformulation of the system structure once again, the new system displayed in Figure 3.3.4.

Now the system input is the reference signal **r**, which gives the error term **e** when summed with the feedback signal. The error signal is then given to the controller, which generates the control signal **u** there is then sent to the plant.

The technique used in this section for tracking step input uses matrix algebra to define a control signal that moves the stable point of the system to the reference input signal's value. The original, intuitive proof for this technique is highly involved and can be referred to in [30]. As a full walk-through of the proof falls outside of the scope of this thesis, however, the following simplified, unintuitive process will be used. It is set up in the following way: First, create a new matrix $\mathbf{T}$, and define it such that:

$$\mathbf{T} = -\mathbf{C}(\mathbf{A} + \mathbf{BF})^{-1}\mathbf{B}. \tag{3.6}$$

Now, inside the controller this $\mathbf{T}$ matrix is combined with the feedback $\mathbf{Fx}$ and reference signal $\mathbf{r}$ to define the control signal $\mathbf{u}$ in the following way:

$$\mathbf{u} = \mathbf{Fx} + \mathbf{T}^{\intercal}(\mathbf{TT}^{\intercal})^{-1}\mathbf{r}. \tag{3.7}$$

The next state dynamics of the system are now defined as

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{BF})\mathbf{x} + \mathbf{BT}^{\intercal}(\mathbf{TT}^{\intercal})^{-1}\mathbf{r}, \tag{3.8}$$

which leads to $\mathbf{x}$ being defined

$$\mathbf{x} = -(\mathbf{A} + \mathbf{BF})^{-1}\mathbf{BT}^{\intercal}(\mathbf{TT}^{\intercal})^{-1}\mathbf{r}, \tag{3.9}$$

and the system output $\mathbf{y}$ then becomes

$$\begin{aligned} \mathbf{y} &= -\mathbf{C}(\mathbf{A} + \mathbf{BF})^{-1}\mathbf{BT}^{\intercal}(\mathbf{TT}^{\intercal})^{-1}\mathbf{r} \\ &= \mathbf{TT}^{\intercal}(\mathbf{TT}^{\intercal})^{-1}\mathbf{r} \\ &= \mathbf{r}, \end{aligned} \tag{3.10}$$

which is our desired result (see [140] for more details). Now the system will attempt to stabilize at the value specified by the reference signal in an optimal fashion, as specified by the cost function.

The output from this step tracking system is shown in Figure 3.3.4. Again there are three different cost functions implemented to show the effects of the state and control cost weighting matrices. When the weight matrix for the state cost $\mathbf{Q}$ was weighted heavier, the system implemented very large control signals to move as quickly as possible to the target, which ended up causing ineffective and energy wasteful actions. Placing more weight in $\mathbf{R}$ caused the system to use smaller control signals

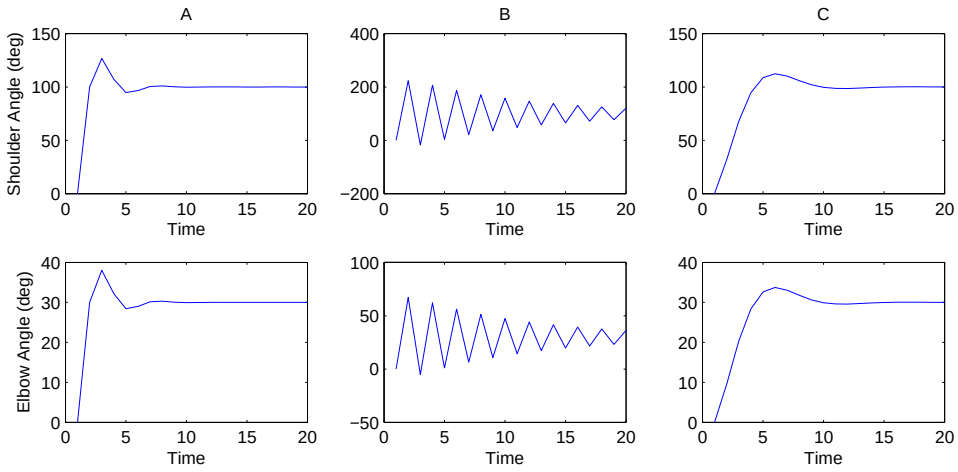

Figure 3.6: In these graphs the x axis is time and the y axis is position, the graphs on top are for θ_1 and the graphs on bottom are for θ_2. A)The performance of the system with $\mathbf{Q}$ and $\mathbf{R}$ set to the identity matrix. B)Here $\mathbf{Q}$ has been weighted 5 times more than $\mathbf{R}$. The system quickly converges to a stable state. C)Here $\mathbf{R}$ is 10 times heavier than $\mathbf{Q}$. The system moves with a lower energy control signal to equilibrium, resulting in a more stable output.

and actually converge to the desired target state faster. These are things that must be taken into account when designing a cost function, and shows the importance and difficulties of choosing a cost function.

Although useful for illustrating the effect of a cost function on control of a system in a simulated environment, this method of control is not robust to noise or incorrect system models, as seen in Figure 3.3.4B. As it is often the case in reality that there will be noise between the controller and the plant, affecting the control signal, as well as between the system sensors and the controller, affecting the feedback. Additionally, system models are abbreviations of the actual physics of a plant. It is necessary to have a control system in place that can account for these undesirable effects and bring the system back under control.

3.3.5 Making the controller robust to noise

The way that these errors in the system model and control and feedback signals will be accounted for is by adding components to the system that have a model of the noise and ideal input/output and attempt to correct the current input/output appropriately. A Kalman filter is added to process the feedback signal, and a linear-

quadratic regulator is used for the control signal.

The ideal performance of the system for a given initial condition is simulated offline, without actually engaging the plant, by using the system model. The input commands $\mathbf{u}$ and output signals $\mathbf{y}$ are then recorded and stored as the optimal input and output signals, denoted $\mathbf{u}_0$ and $\mathbf{y}_0$. During system operation, system signals are measured in terms of their deviations from these ideal input and output signals. The Kalman filter provides an estimate of the actual system output, which allows the deviation from ideal output, denoted $\delta\mathbf{z}$, to be calculated and used in the generation of $\delta\mathbf{u}$. The corrective signal from the linear-quadratic regulator, $\delta\mathbf{u}$, is then summed with the ideal control signal and sent to the plant.

A system set up with this structure is reffered to as Linear-Quadratic-Gaussian (LQG) controller, for a linear system model with a quadratic cost function disturbed by additive white Gaussian noise. The derivation and application of LQG systems is reviewed in the tutorial style article by Michael Athans, [9]. An depth examination of such a system is excluded here, but it is important to note that methods of making a system robust to noise exist.

Even when made robust to noise, however, the controller developed so far only operates optimally in joint space. The target is specified as the joint angles that the arm must have to put the end effector at the desired location in 2D space. Because it is working through joint torque, it moves to the target optimally in *joint space*, not in 2D space. The implications of this can be seen in Figure 3.3.5 which compares the arm moving to its target position in joint space and in 2D space.

As the end effector is of primary concern, it is desirable to control the joint space model in a way such that it operates optimally in 2D space. To do this a system hierarchy will be introduced.

3.4 Hierarchical control theory

A hierarchical system of controllers allows for the low dimensional control of high dimensional systems by defining transformations to more and more abstract representations of the system and operating at the highest level.

As mentioned previously, hierarchical control has long been thought to be an underlying characteristic of motor control in the brain, however it has only been very recently that an effective framework has been developed [131]. It is based on applying

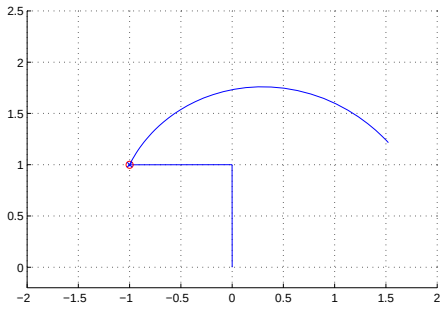

Figure 3.7: Optimized in joint space.

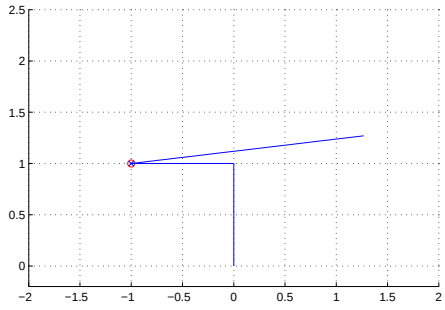

Figure 3.8: Optimized in end effector 2D space.

Figure 3.9: The trajectories taken by the end effector when the arm moves optimally to the target position in joint space and 2D space.

optimal control theory to a multi-layer structure and can be used to analytically define a hierarchical system. Using this framework and a 7 degrees-of-freedom (DOF) arm model operating in 3D space motor, control systems that can reproduce human arm movement have been developed recently [72].

One of the important features of this new approach is that it provides an analytical approach to the design of complicated hierarchical systems, which allows an explicit understanding of the functions performed at all points in the system and can provide valuable insight into the dynamics of such systems. This approach could be contrasted with a machine learning approach where a multi-level system is trained to some unknown function that can replicate training set data within some range of tolerable error. Although the end result may be similar, it is difficult to gain insight into the operation of the underlying systems through machine learning. As the goal of this research is to understand how motor control operates in the brain the analytical approach will be taken here.

In this section the simple linear model developed in the last section will be modified and extended structurally to become a hiearchical two level system. The top level will be a representation of the end effector dynamics in 2D space, and the bottom level will be a joint space representation of the arm dynamics. The control system will optimize the more abstract 2D space end effector representation, and send a descending signal to the lower level to guide the joint torque control.

First, both levels of representation will be defined; this will involve a change in

notation to stay consistent with the hierarchical control theory in [131]. A simple high level controller will then be implemented to judge the effectiveness of a high level controller operating with only consideration to the 2D movement. After this a model of the low level system dynamics and control cost will be incorporated into the high level representation and a comparison between the effectiveness and energy consumption of both models will be compared.

3.4.1 System design

Restating the low level dynamics

The controllers designed up until this section have been explicity for linear system models. This was reflected in the formulation of the dynamics

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

by the matrices $\mathbf{A}$ and $\mathbf{B}$. At this point the notation of the dynamics will be modified for the sake of generality. The dynamics will now be formulated as

$$\dot{\mathbf{x}} = \mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}, \tag{3.11}$$

where $\mathbf{a}(\mathbf{x})$ can be any function, and the control matrix $\mathbf{B}$ can be dependent on on the state vector $\mathbf{x}$, but still must interact with the control signal $\mathbf{u}$ linearly. The reason for creating a dependence on $\mathbf{x}$ for $\mathbf{B}$ is to allow the control matrix to compensate for body inertia in sufficiently complex models.

The low level model for this system will continue to be modeled linearly for simplicity. However, the techniques used here can also be applied to nonlinear system models at the lower level as well. Using the same system model developed previously for the low level model, attention is now turned to defining the high level model.

The high level system

Previously, the $\mathbf{y}$ variable referred to the output vector of the system. Now, however, $\mathbf{y}$ will be used to denote the high level state vector, which is a static function of the plant state:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}). \tag{3.12}$$

How the tranformation function $\mathbf{h}$ is defined will be very important. In [72] it is stated that $\mathbf{h}$ must be defined such that enough information is preserved in the transformation to recover the state cost of the lower level, and that the dimensionality of the high level state vector must be lower than the low level state vector. The first part of this statement is true, to perform approximate optimization there must be enough information preserved by $\mathbf{h}$ such that the low level state cost can be recovered. The reason for this being that if the high level controller is confused about the actual state of the plant, proper control of the system is unlikely.

Reducing the dimensionality, however, is more of a guideline than a requirement. Ideally the dimensionality will be reduced by $\mathbf{h}$, but in this toy system it is not feasible to reduce the system any more, since we are already in a 2-dimensional space. Rather, the hierarchy will be used to transform the system into a more intuitively controlled state space. Specifically, while the shortest trajectory for the end effector to take in 2D space may not be clear in joint space on the low level, on the high level it is simply a straight line to the target.

The high level system dynamics will be denoted

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}) + \mathbf{G}(\mathbf{y})\mathbf{v} \tag{3.13}$$

where $\mathbf{f}(\mathbf{y})$ is the system's passive dynamics, $\mathbf{G}(\mathbf{y})$ is the control-dependent dynamics, and $\mathbf{v}$ is the high level control signal. Together, $\mathbf{f}$ and $\mathbf{G}$ should allow the high level controller to choose control signal $\mathbf{v}$ in the preferred (and potentially lower-dimensional) high level representation space. In the case of systems where the transformation between levels also reduces dimensionality the high level controller will then also be able to operate on a lower dimensional space, controlling the system without all of the plant details.

To relate Eq. (3.12) to Eq. (3.13), we differentiate Eq. (3.12) with respect to time and substitute in for $\dot{\mathbf{x}}$, which yields

$$\dot{\mathbf{y}} = H(\mathbf{x})(\mathbf{a}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}) \tag{3.14}$$

where $H(\mathbf{x}) = \partial\mathbf{h}(\mathbf{x})/\partial\mathbf{x}$ is the Jacobian of the function $\mathbf{h}$. Now the low level controller is acting to choose a control signal $\mathbf{u}$ such that the performance of the low level system (Eq.(3.14)) matches the desired system performance described by the

high level system (Eq. (3.13)). Formally, this is written

$$H(\mathbf{x})\mathbf{a}(\mathbf{x}) + H(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} = \mathbf{f}(\mathbf{y}) + \mathbf{G}(\mathbf{y})\mathbf{v}. \tag{3.15}$$

Once the system dynamics of the high level are modeled, by defining $\mathbf{f}(\mathbf{y})$ and $\mathbf{G}(\mathbf{y})\mathbf{v}$, the high level controller will send its optimal control signal $\mathbf{v}$ to the lower level, whose task it is then to find a minimal cost low level control signal $\mathbf{u}$ such that Eq. (3.15) is satisfied.

Explicit and implicit modeling

There are two ways that the feedback loop for the optimal controller on the high level can be set up, through explict and implicit modeling. In explicit modeling, the dynamics of the high level are completely defined at the high level, the effect of the high level command $\mathbf{v}$ on the low level plant is simulated through $\mathbf{f}(\mathbf{y})$ and $\mathbf{G}(\mathbf{y})$ and the actual results of the command on the plant are restricted to the low level. The benifit of this method is that modularity arises, and standard optimization techniques can be used on the high level to find the optimal high level control signal. Modularity also allows for offline simulations of control signals at the high level, as most of the techniques for finding optimal control signals involve iterations. Being able to carry out trials without actually using the plant allows for very quick testing, since the entire system does not need to be engaged.

The second method, implicit modeling, gives the high level online access to $H(\mathbf{x})\mathbf{a}(\mathbf{x})$, so it does not need to model the low level passive dynamics of the system at all. Instead it knows their transformation exactly and uses that to update $\mathbf{y}$. As a result of $\mathbf{f}(\mathbf{y}) = H(\mathbf{x})\mathbf{a}(\mathbf{x})$, the necessary conditions for the low level control signal $\mathbf{u}$ change from Eq. (3.15) to

$$H(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} = \mathbf{G}(\mathbf{y})\mathbf{v}. \tag{3.16}$$

Now it is possible for the high level controller to incorporate the low level state cost into its trajectory planning and exploit the low level passive dynamics. In addition, Eq. (3.16) now isolates the control-dependent dynamics on both levels so the high level can also employ some function $\tilde{r}(\mathbf{v}, \mathbf{h})$ to match the low level cost function $r(\mathbf{u}, \mathbf{x})$ exactly. The high level controller can now work to solve the exact optimal control problem using both the true plant dynamics and true cost function. Since the high

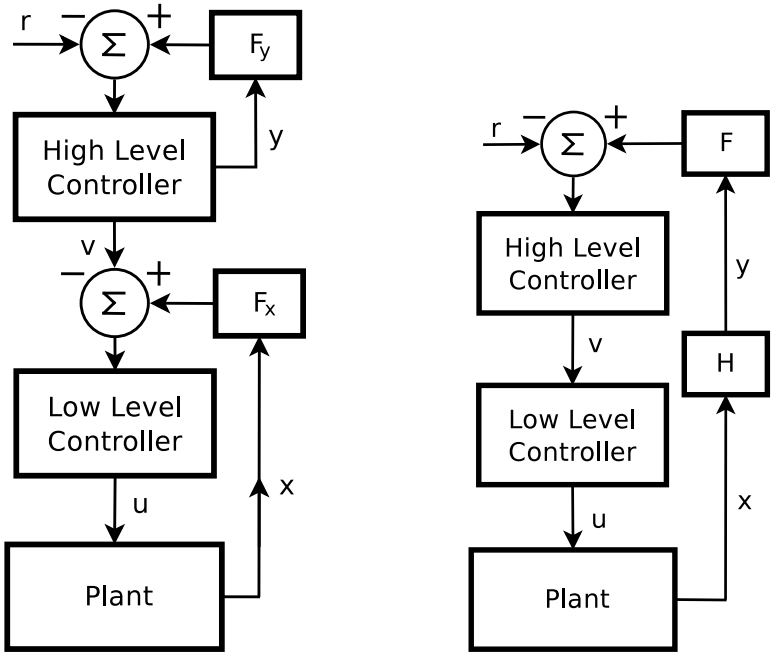

Figure 3.10: Block diagrams of hierarchical systems employing the explicit and implicit modeling methods.

level system is no longer autonomous, however, it is not guaranteed that standard optimization techniques will provide optimal solutions.

Both the explicit and implicit methods will be developed in this section, and their performance will be compared at the end. As stated above, the dynamics are the difference between these two methods, so we simply set the state vector of the high level

$$\mathbf{y} = \begin{bmatrix} \mathbf{p} \\ \dot{\mathbf{p}} \end{bmatrix}, \tag{3.17}$$

where $\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$ and $\dot{\mathbf{p}} = \begin{bmatrix} \dot{p_x} \\ \dot{p_y} \end{bmatrix}$, and let the high level control signal $\mathbf{v}$ be end effector force in 2D space.

The explicit model dynamics will be based on a simple linear approximation of the low level dynamics, such that

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} \dot{\mathbf{p}} \\ 0 \end{bmatrix}, \quad \mathbf{G}(\mathbf{y}) = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \tag{3.18}$$

where $\mathbf{I}$ is the identify matrix. The transformation matrix that will take the low level to the high level can then be defined

$$H(\mathbf{x}) = \begin{bmatrix} \mathbf{J}(\theta) & 0 \\ \dot{\mathbf{J}}(\theta) & \mathbf{J}(\theta) \end{bmatrix}. \tag{3.19}$$

where $\mathbf{J}(\theta) = \partial\mathbf{p}/\partial\theta$, and $\dot{\mathbf{J}}(\theta) = \frac{d}{dt}(\mathbf{J}(\theta))$. The lower rows of $H(\mathbf{x})$ are garnered through rearranging the equation for $\mathbf{J}(\theta)$to solve for $\partial\mathbf{p}$ and taking the derivative with respect to time.

For the implicit model the high level will be updated using the transformed low level passive dynamics, so instead let

$$\mathbf{f}(\mathbf{y}) = \begin{bmatrix} \dot{\mathbf{p}} \\ \mathbf{e} \end{bmatrix}, \quad \mathbf{G}(\mathbf{y}) = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \tag{3.20}$$

where $\mathbf{e}$ is the 4^{th} to 6^{th} rows of $H(\mathbf{x})\mathbf{a}(\mathbf{x})$.

At this point it is possible to start designing the optimal controller for the explicit model of the high level using the previously described optimal control methods and cost function for linear systems. For the implicit model method, though, there is still work to be done.

Much like the low level system model was restated for generality, the cost function used by the implicit model will also be redefined here. The state and control cost weight matrices $\mathbf{Q}$ and $\mathbf{R}$ will be replaced by the instaneous state and control dependent costs $q(t, \mathbf{x})$ and $r(t, \mathbf{u})$,

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \int (q(t, \mathbf{x}) + r(t, \mathbf{u}))dt. \tag{3.21}$$

This cost function represents the true cost function of the plant, as opposed to the cost function used by the explicit model, which is based solely on the high level dynamics. As discussed previously, it is possible for the system to recover the current low level state cost $q(t, \mathbf{x})$ through $\mathbf{h}(\mathbf{x})$ by definition, so an accurate model of the control-dependent cost on the low level would allow the high level controller to fully account for the actual cost function while operating in the preferred representation space.

Modeling the low level cost function

To complete the high level system model definition there must be a control-dependent cost created for the high level cost function $\mathcal{L}$. As mentioned in the previous section, it is desirable to have the high level cost function accurately represent the low level cost function so the plant dynamics can be exploited by the high level controller.

Define $\tilde{r}(t, \mathbf{v})$ to be the high level representation of the low level $r(t, \mathbf{u})$. Although it is guaranteed that the low level state cost $q(t, \mathbf{x})$ can be accurately reproduced on the high level, there is no such assurance that the low level $r(t, \mathbf{u})$ cost can be recovered on the high level. However, control dependent costs are usually modeled using a quadratic:

$$r(t, \mathbf{u}) = \frac{\gamma}{2}\mathbf{u}^\intercal\mathbf{u} \tag{3.22}$$

If it is possible to represent the low level control signal $\mathbf{u}$ using the high level control $\mathbf{v}$ as

$$\mathbf{u} = K(\mathbf{x})\mathbf{v} \tag{3.23}$$

for some $K(\mathbf{x})$, then the high level control dependent cost can be formulated as

$$\tilde{r}(t, \mathbf{v}) = \frac{\gamma}{2}\mathbf{v}^\intercal K(\mathbf{x})^\intercal K(\mathbf{x})\mathbf{v}, \tag{3.24}$$

accurately recovering the low level $r(t, \mathbf{u})$.

Taking the low and high level control dependent dynamics, as defined by equations (3.18) and (3.20), and applying them to the equality defined by Eq. (3.16), gives

$$\mathbf{J}(\theta)\mathbf{u} = \mathbf{v}. \tag{3.25}$$

We can now perform SVD on $\mathbf{J}(\theta)$ such that $\mathbf{J}(\theta) = U\Lambda V^\intercal$. The matrix $V^\intercal$ can then be seen as projecting the control signal $\mathbf{u}$ into task relevant (Ω) and irrelevant ($\bar{\Omega}$) spaces. In other words $V^\intercal$ sets the low level controls that do not affect elements of $\mathbf{x}$ that contribute directly to the high level state to 0. Because the transformation $H(\mathbf{x})$ that defines the high level state preserves enough information to recover the low level state cost $q(t, \mathbf{x})$ this can be seen as weeding out the unimportant controls from the low level.

Now the high level control $\mathbf{v}$ need only be concerned about matching $V^\intercal \mathbf{u}$, so define

$$K(\mathbf{x}) = (U\tilde{\Lambda})^{-1} \tag{3.26}$$

and the high level cost $\tilde{r}(t, \mathbf{v})$ can recover the low level control dependent cost through Eq. (3.24). The implicit model has now captured the passive dynamics and true cost of the system from the low level.

3.4.2 Optimal control of the high level

Optimal control of the high level, in the explicit model case, can be done using the techniques developed in the linear systems section.

To create an optimal control signal for the implicit model case, an iterative algorithm for optimal control design for nonlinear stochastic systems (presented in [136]) will be employed. The idea behind the algorithm is analagous to second order methods for numerical optimization. The goal is to find the minimum of a complex function. The function, even though not likely quadratic, can be approximated locally using a quadratic. The process finds the minimum of the quadratic, constructs a new local quadratic approximation of the complex function at that point and repeats until convergence.

Quadratics are used in numerical optimization because they represent the most complex function with an analytical solution. Similarly, the LQG problem is the most complex optimal control problem with an analytical solution, so a local LQG approximation of the system dynamics is made, the optimal LQG controller is found, the solution is used to construct a new local LQG approximation, and the process iterates until convergence [71]. Quite aptly this algorithm is titled the iterative LQG (iLQG) method.

Much like the second order numerical optimization methods are only guaranteed to find local minima, the iLQG technique will only converge to a local optimal solution. Therefore, it will be necessary to seed the initial iteration with an appropriate approximation of the optimal trajectory. The seed used here will be the trajectory generated by a simple linear estimation of the system dynamics, ie the trajectory planned by the explicit model's high level controller, which is an optimal path in end effector space without consideration for the low level dynamics.

3.4.3 Designing the low level controller

The low level controller is responsible for generating the minimal cost control signal $\mathbf{u}$ that satisfies Eq. (3.15). This is a constrained optimization problem and can be attacked using quadratic programming, with the MATLAB function `quadprog`. This function will minimize

$$\frac{\gamma}{2}\mathbf{u}^{\intercal}\mathbf{u} \tag{3.27}$$

constrained by

$$H(\mathbf{x})\mathbf{a}(\mathbf{x}) + H(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} = \mathbf{f}(\mathbf{y}) + \mathbf{G}(\mathbf{y})\mathbf{v},$$

which, in the case of implicit modeling, reduces to Eq. (3.16).

It is also possible to introduce physical constraints at this level by introducing a cost term that punishes movements approaching the joint limits, $[\theta^{\mathtt{min}}, \theta^{\mathtt{max}}]$. Detailed in [72], the process involves adding the term $\nabla_t \sum_i^n g(\theta_i)$ where g is a polynomial function of the form

$$g(\theta_i) = \frac{(\alpha_i\theta_i + \beta_i)^6}{100^6} \tag{3.28}$$

for $i = 1, ..., n$, where n is the number of degrees of freedom of the system model, and

$$\alpha_i = \frac{200}{\theta_i^{\mathtt{max}} - \theta_i^{\mathtt{min}}}, \quad \beta_i = 100 - \frac{200}{\theta_i^{\mathtt{max}} - \theta_i^{\mathtt{min}}}.\theta_i^{\mathtt{max}}. \tag{3.29}$$

The $\alpha_i\theta_i + \beta_i$ normalizes the joint angle $\theta_i \in [\theta_i^{\mathtt{min}}\theta_i^{\mathtt{max}}]$ into a range of $[-100, 100]$, and 100^6 then normalizes $g(\theta_i)$ into $[0, 1]$.

When joint constraints are added into the low level cost function, the constrained optimization problem becomes minimizing

$$\frac{\gamma}{2}\mathbf{u}^{\intercal}\mathbf{u} + \eta\nabla_t \sum_i^n g(\theta_i) \tag{3.30}$$

constrained by eq (3.15) under explicit modeling and eq (3.16) under implicit modeling.

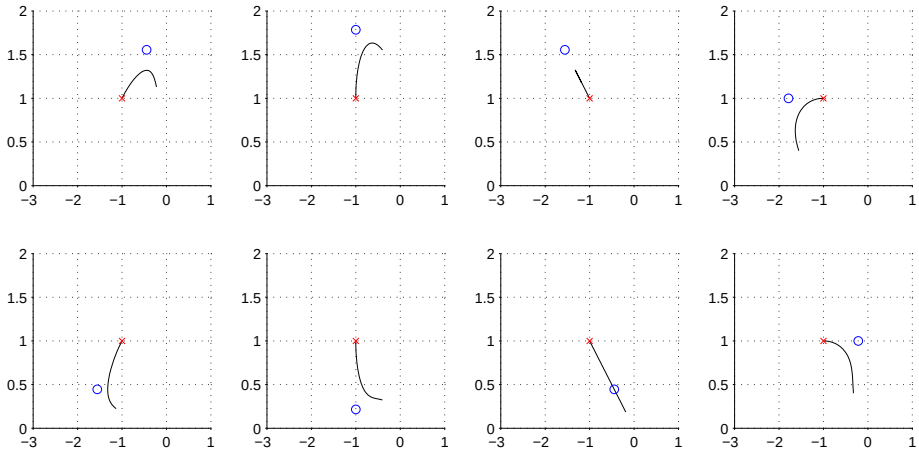

Figure 3.11: Explicit arm model controlled arm trajectories. Here the red x represents the start point and the blue circle is the target. In all cases the system fails to end at the target and is highly expensive. From the top left and going across and to the next line: Cost: 2340.303307, Cost: 4232.832443, Cost: 2965.638202, Cost: 4199.426291, Cost: 2333.154190, Cost: 3770.481721, Cost: 1365.204799, Cost: 3731.474542.

3.4.4 Results of explicit and implicit control

In this section the results of controllers designed through explicit and implicit modeling methods will be compared.

First, the performance of the explicit model will be examined. This will be an autonomous linear controller on the top level that bases its control signal off of a simple low level system model. The arm model will have two links that are each of length and mass 1, will start at an initial end effector position of (-1,1) in (x,y) coordinates (equivalent to $(\frac{\pi}{2},\frac{\pi}{2})$ in joint space, where the elbow and the shoulder are at 90°), and move to 8 points on a circle with a 1.5 diametre centered at the initial end effector position. The system is simulated over five seconds with a half-second timestep.

The results are displayed in Figure 3.4.4, and it is plain that they are far from optimal. In all cases the arm fails to reach the endpoint and the cost is exorbitant (which becomes clear when comparing to the implicit controller below). The total cost of each movement, μ, is calculated as

$$\mu = \int u(t)^2 dt + \omega_p e_{pos} + \omega_v e_{vel}$$

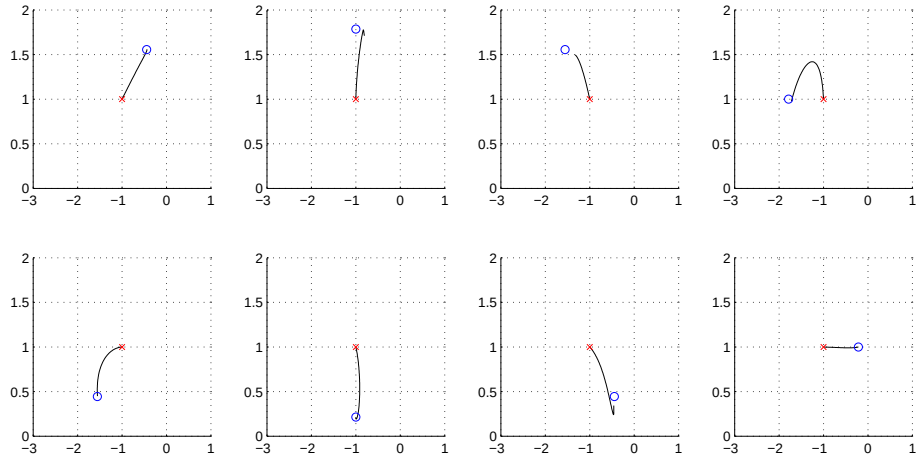

Figure 3.12: Implicit arm model controlled arm trajectories. From the top right, following the trajectory counterclockwise: Iterations = 100; Cost = 0.0668, Iterations = 60; Cost = 1052.6089, Iterations = 38; Cost = 1052.5572, Iterations = 64; Cost = 86.9450, Iterations = 31; Cost = 44.4467, Iterations = 82; Cost = 9.3194, Iterations = 38; Cost = 254.3198, Iterations = 93; Cost = 0.1023.

where ω_p is the weighting on the error between the endpoint hand and target locations, e_{pos}, and ω_v is the weighting of the velocity at the last timestep, e_{vel}.

The results from the implicit model simulation over five seconds with a half-second timestep are shown in Figure 3.4.4. The simulated implicit model operates in the same environment with the same initial conditions and low level dynamics as the explicit model, but has online access to the current system state. This allows the system to be optimized in regard to the true cost function of the system plant, as opposed to a simplified model. These results show that the implicit model achieves much better control of the model by accounting for the low level dynamics when creating a high level optimal control signal.

The work in this thesis will be based on using implicit modeling in the hiearchical system. Based on the ready availability of low level system state feedback (body position) from the highly interconnected sensory cortex and the benifits of high level optimization that accounts for the low level dynamics it seems unlikely that the motor control system in the brain would not have a similar principle of operation. However, the current system control is still not ideal. The high level iLQG optimization technique discussed in this section has several critical shortcomings. The process requires an approximately optimal initial trajectory, as it can only converge to a local optimal

solution, iteration, and the generalization of the control signals to novel actions is unclear.

In the next section a technique that guarantees the global optimal control signal without iteration will be introduced which will form the basis for the high level control system presented in the next chapter.

3.5 Efficient computation of optimal actions

In 'Efficient computation of optimal actions' (2009), Todorov presents a new method for optimal control of systems. This technique is based on recognizing the duality between the optimal control problem and Bayesian inference.

In optimal control problems, the controller takes into account the current state and goal of the system to determine an optimal action at that point in time which is then executed, the clock is updated, and the process is repeated until the goal is reached. To reformulate this as an inference problem, we pretend that the system has already reached the goal state and is trying to infer the muscle activation that moved the system from its current state to the goal. The clock then updates and the process repeats until the goal state is reached.

In arm models with more than 2 degrees-of-freedom there is no unique solution to this problem, because there are infinitely many ways to accomplish a task; this is known as motor redundancy. When the goal of the system is to point to a dot on the wall, there are infinite possible trajectories to take to bring the end effector to the wall, in addition to infinitely many possible arm positions to take at a final state.

While a lack of complete system state goals at each timestep make control problems more difficult, and likewise a lack of complete sensory measurements at each timestep inference problems more difficult, motor redundancy gives the system greater flexibility. The system is allowed to be controlled inexactly and still achieve the goal, because deviations can be corrected for at a later point in time if they prove adverse to task completion. Analogously in inference, long term predictions are difficult, while short term predictions (inferring motor control commands near the target) are easier.

It is also possible to reformulate inference problems as control problems. Given the current sensory measurement and the previous world state estimate, let the inference problem be correctly estimating the current state of the world. The control problem is to take the state estimate and correct it such that it agrees with the current sensory

measurement. The cost function will be concerned with balancing the size of the correction with the accuracy of the estimate. The control cost is zero when there is no correction to the estimate and the accuracy cost is zero when there is no deviation from the measurement.

In large environments this is a very difficult problem, because the controller is trying to correct for all of the environment's elements. However, if perception is geared towards the task-relevant observations then then problem becomes significantly more tractable. This problem highlights the importance of sensory filtering in the motor control system and will be brought up again in the next chapter.

These examples demonstrate that optimal control and Bayesian inference can be related when task goals are associated with sensory mesurements and control signals are associated with state estimate corrections [135]. Stating this in another way, Bayesian inference is the sensory domain equivalent to optimal control, so Bayesian inference solutions can be used on optimal control problems if they can be reformulated appropriately.

Key to the formalization of this duality is that the probabilities used in inference and the costs used in control are related by an exponential transformation. Intuitively this can be seen when one considers that accumulating costs add while probabilities multiply, and the expontial transformation turns sums into products. This idea is worked through in further detail in [135], but at this point we can turn to exploiting the exponential transformation to explicitly rework optimal control problems into the probability domain, resulting in a greatly simplified Bellman's equation.

3.5.1 Simplifying Bellman's equation

As stated previously, the Bellman equation expresses the optimal cost-to-go function, $v(\mathbf{x})$, as the minimum over actions $\mathbf{u}$, of the immediate cost, $l(\mathbf{x}, \mathbf{u})$ plus the expected cost-to-go $E[v(\mathbf{x}')]$ at the next state $\mathbf{x}'$, formally this is written

$$v(\mathbf{x}) = \min_{u} \{\, l(\mathbf{x}, \mathbf{u}) + E_{x' \sim p(\cdot|\mathbf{x},\mathbf{u})}[\, v(\mathbf{x}')\,]\,\}. \tag{3.31}$$

Here optimal control in a discretized space will be examined. Often it is the case that the set of actions our controller has available are thought of as discrete 'go left' or 'go right' commands. To move to the probability domain however, it will be necessary for our actions to be continous. How to do this becomes clear when one considers

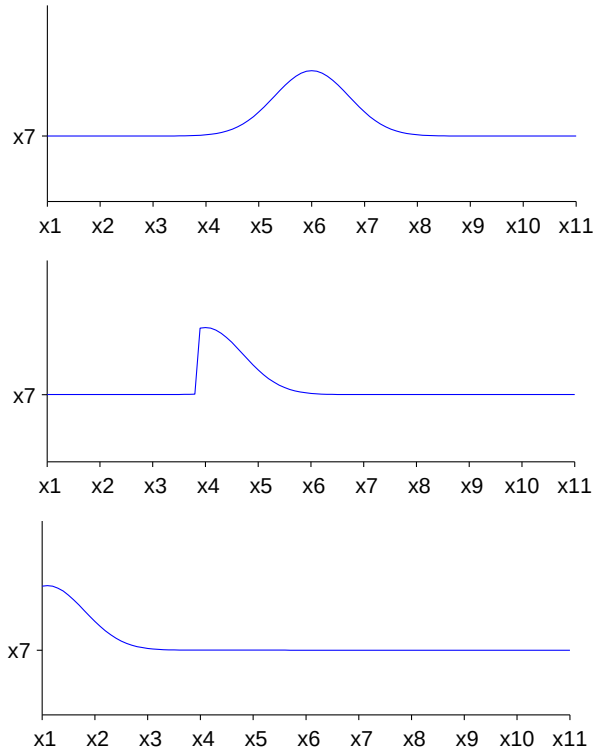

Figure 3.13: If the system's transition probabilities for a state x7 are specified as in the top figure, and the controller is attempting to move the system to x1, it can adjust the probabilities under passive dynamics so as to reach the target as quickly as possible, as in the middle figure. It cannot, however, move the system to a point outside of the reach of movement under passive dynamics, as in the lower figure.

that although the actions are typically thought of as discrete, what the controller is doing in specifying an action is adjusting the transition probabilities of the system dynamics, which are continous.

In this framework then, instead of asking the controller for an action which is then translated into an effect on the system's transition probabilities, the controller is allowed to directly specify the system's transition probabilities. Where the probability of the next state $\mathbf{x}'$ given the current state $\mathbf{x}$ under passive dynamics is denoted $p(\mathbf{x}'|\mathbf{x})$, the probability of the next state given the state and the control signal $\mathbf{u}$ is denoted $p(\mathbf{x}'|\mathbf{x},\mathbf{u}) = u(\mathbf{x}'|\mathbf{x})$.

The controller is restricted, however, in that it can only move the system to points reachable under passive dynamics. An example of this is illustrated in Figure 3.5.1. This keeps the controller from making unreasonable movements when specifying the transition probabilities.

The control cost of an action is specified by the Kullback-Leibler (KL) divergence function, which is a standard measurement of difference between two probability distributions, in this case between the state distribution under passive dynamics $p(\cdot|\mathbf{x}')$ and under control $u(\cdot|\mathbf{x}')$. Specifically, the KL-divergence measures the expected logarithmic ratio between two distributions, and is denoted $KL(a(\cdot|x) \;||\; b(\cdot|x)\;) = E_{x' \sim a(\cdot|x)}\left[\log\left(\frac{a(\mathbf{x}'|\mathbf{x})}{b(\mathbf{x}'|\mathbf{x})}\right)\right]$. To keep the KL-divergence well defined, $u(\mathbf{x}|\mathbf{x}')$ will be set to zero whenever $p(\mathbf{x}'|\mathbf{x}) = 0$.

Thus, under this system the immediate cost, $l(\mathbf{x}, \mathbf{u})$ of the Bellman equation can be specified as

$$l(\mathbf{x}, \mathbf{u}) = q(\mathbf{x}) + KL(u(\cdot|\mathbf{x}) \;||\; p(\cdot|\mathbf{x})\;) = E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})}\left[\log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})}\right)\right] \tag{3.32}$$

where $q(\mathbf{x})$ is a state cost function specifying the penalty for being in a state. An example of a state cost function would be to encode the euclidean distance from the target.

At this point a new function will be introduced to ease the notation and facilitate intuitive understanding of the following equations. Let $z(\mathbf{x}) = \mathrm{e}^{-v(\mathbf{x})}$ be known as the desirability function, equal to the exponentiation of the negative cost-to-go of the state $\mathbf{x}$. It is termed the 'desirability' function because it reflects the fact that the function is large wherever the cost-to-go is small, and small wherever the cost-to-go is large.

Rewriting the Bellman equation in terms of z and substituting in Eq (3.32) gives

$$-\log(z(\mathbf{x})) = q(\mathbf{x}) + \min_u \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})}\left[\log \frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x}) z(\mathbf{x}')}\right]\right\}. \tag{3.33}$$

The term being minimized is almost a proper KL-divergence, except that the distribution $p(\cdot|\mathbf{x})z(\cdot)$ does not sum to one. To correct this, a normalization term $\mathcal{G}[z](\mathbf{x})$ will be introduced. Let

$$\mathcal{G}[z](\mathbf{x}) = \sum_{\mathbf{x}'} p(\mathbf{x}'|\mathbf{x}) z(\mathbf{x}') = E_{\mathbf{x}' \sim p(\cdot|\mathbf{x})}[z(\mathbf{x}')], \tag{3.34}$$

and multiply and divide the denominator to give

$$\begin{aligned}
-\log(z(\mathbf{x})) &= q(\mathbf{x}) + \min_u \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[\log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')\mathcal{G}[z](\mathbf{x})/\mathcal{G}[z](\mathbf{x})} \right) \right] \right\} \\
&= q(\mathbf{x}) + \min_u \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[-\log(\mathcal{G}[z](\mathbf{x})) + \log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')/\mathcal{G}[z](\mathbf{x})} \right) \right] \right\} \\
&= q(\mathbf{x}) + \min_u \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[-\log(\mathcal{G}[z](\mathbf{x})) + KL\left(u(\mathbf{x}'|\mathbf{x}) \,\middle\|\, \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})} \right) \right] \right\}.
\end{aligned} \tag{3.35}$$

The global minimum of a KL-divergence occurs when the two distributions being compared are equal. Then with $u(\cdot|\mathbf{x})$ part of a KL-divergence the optimal control signal, u^*, is therefore

$$u^*(\mathbf{x}'|\mathbf{x}) = \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})}. \tag{3.36}$$

Now it is possible to simplify the Bellman equation. First exponentiate and take into account the normalization term

$$z(\mathbf{x}) = \exp\left(-q(\mathbf{x}) - \min_u \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[-\log\left(\mathcal{G}[z](\mathbf{x}) + KL\left(u(\mathbf{x}'|\mathbf{x}) \,\middle\|\, \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})} \right) \right) \right] \right\} \right), \tag{3.37}$$

now because $-\log(\mathcal{G}[z](\mathbf{x}))$ is not dependent on $\mathbf{u}$ it can be moved outside the minimization, and assuming optimal control u^* set the KL-divergence to zero giving

$$\begin{aligned}
z(\mathbf{x}) &= \exp(-q(\mathbf{x}) + \log(\mathcal{G}[z](\mathbf{x})) + 0) \\
z(\mathbf{x}) &= \mathrm{e}^{-q(\mathbf{x})}\mathcal{G}[z](\mathbf{x}).
\end{aligned} \tag{3.38}$$

Since the expectation $\mathcal{G}[z](\mathbf{x})$ is a linear operator then Eq. (3.38) is linear in z! Rewriting this in vector notation will also reveal information about z. Enumerate the states 1 to n and let $z(\mathbf{x})$ and $q(\mathbf{x})$ be represented by the n-dimentional column vectors $\mathbf{z}$ and $\mathbf{q}$, respectively. Let $p(\mathbf{x}'|\mathbf{x})$ be represented by the n-by-n matrix $\mathbf{P}$, where the row indices correspond to $\mathbf{x}$ and the column indices correspond to $\mathbf{x}'$. Now it is possible to rewrite Eq. (3.38) as

$$\mathbf{z} = \mathbf{QPz} \tag{3.39}$$

where $\mathbf{Q} = \mathrm{e}^{-q(\mathbf{x})}$. In this form it is clear to see that $\mathbf{z}$ is an eigenvector of the matrix

$\mathbf{QP}$, and in fact z can be solved for using power iteration once the state cost matrix $\mathbf{Q}$ and passive dynamics matrix $\mathbf{P}$ have been defined. Using z the controller can recover the optimal action at each defined state in the environment.

This gives a very clean solution for the optimal control of systems operating in a discrete environment. Because they optimize a linear version of the Bellman equation, the controllers generated through this process will hereafter be referred to as 'linear Bellman controllers' (LBCs). The application of LBCs to continuous state environments is discussed in [134], which shows that the continous state environment is a specialized case of the discrete environment. As the discrete environment provides such a clean solution to optimal control, however, it will be the focus of the work done here.

Operating inside continuous environments using discrete representations will be discussed further later in this section, but it is important to note that the 'continuous' representation is essentially a discrete representation with a very small step between represented points in the environment. This allows the discrete framework to be used for modeling without decreasing the power of LBCs, and in fact is used as the basis for a highly efficient approximately optimal controller in large environments requiring fine-grained control where detailed representation of the entire operating space is implausible.

At this point the implementation process for this algorithm will be worked through as a high level controller for the hierarchical system of the last section is developed. First a controller that operates optimally with concern strictly for end effector position in 2D space will be created, then an efficient method for fine-grained control in large environments will be examined. As well, consideration will be given to techniques for object avoidance, and finally the addition of low level dynamics considerations will be considered.

3.5.2 Implementing the algorithm

One of the most computationally expensive components of this algorithm is the generation of the passive dynamics matrix $\mathbf{P}$. Every possible state is enumerated along the columns and rows of $\mathbf{P}$, the rows corresponding to $\mathbf{x}$ and the columns to $\mathbf{x}'$. A probability distribution is then generated based on the passive dynamics of state $\mathbf{x}$ over all of the potential next states.

Using a simple linear model of end effector movement in 2D space the modeled

system's state and control vectors will be defined

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ \dot{p_x} \\ \dot{p_y} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

with system dynamics reminiscent of Eq. 3.3,

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}.$$

The variance of the probability distribution over $\mathbf{x}'$ for a state $\mathbf{x}$ will be a function of the velocity. Importantly, the controller can move the system anywhere within the range of the distribution of $\mathbf{x}'$. This means that how the distribution over the possible next states is specified is critical to proper formation of the controller. In this system, because acceleration is controlled through the forces applied to p_x and p_y, the distribution must vary only over the range of velocities, $\dot{p_x}$ and $\dot{p_y}$. If the distributions were to vary over p_x and p_y the controller would expect to be able to directly affect position, which it cannot.

The size of the $\mathbf{P}$ matrix depends on the range of represented state variables. For this implementation, each of the variables will have a range of 10 units. This will result in 10^4 possible states, requiring the $\mathbf{P}$ matrix to have at least 10^8 elements. However, most of these elements will be zero, so it is possible to significantly reduce the memory required to represent $\mathbf{P}$ with a sparse matrix.

Once the $\mathbf{P}$ matrix has been specified, the state cost matrix, $\mathbf{Q}$, must be defined. This matrix is significantly less expensive to compute, as it is a diagonal matrix encoding the (un)desirability of each state. For this controller, the state cost function will simply be the euclidean distance of the state's p_x and p_y elements to the target, plus the absolute value of the velocities $\dot{p_x}$, $\dot{p_y}$ to ensure that the ideal state is a standstill at the target location. Again, $\mathbf{Q}$ will be represented by a sparse matrix as it only has diagonal values.

Once the $\mathbf{P}$ and $\mathbf{Q}$ matrices are generated, the principle eigenvector of $\mathbf{QP}$ can be found with MATLAB's `eigs` command for low dimensional systems. This setup

minimizes the infinite horizon average cost, where the principle eigenvalue can be written $\lambda = \mathrm{e}^{-c}$ with c being the average cost of an action.

$$\mathrm{e}^{-c} z(\mathbf{x}) = \mathrm{e}^{-q(\mathbf{x})} \mathcal{G}[z](\mathbf{x}) \quad \Longrightarrow \quad \mathrm{e}^{-c} \mathbf{z} = \mathbf{QPz} \tag{3.40}$$

For larger systems it becomes necessary to use eigenfunction approximation techniques such as those described in [132]. Here eigenfunction approximation is unnecessary due to the simple system being modeled.

With $\mathbf{z}$ determined, it is possible to define the u^* at every state, using Eq. (3.36). Recall that a command $u^*(\mathbf{x}'|\mathbf{x})$ defines the optimal transition probability over $\mathbf{x}'$ for a state $\mathbf{x}$ by modifying the distribution over next states determined by the passive dynamics. To recover the specific action the controller must apply to bring about the desired transition probability, the state corresponding to the maximum distribution value is taken from $p(\mathbf{x}'|\mathbf{x})$ and $u^*(\mathbf{x}'|\mathbf{x})$, and the passive dynamics maximum state is subtracted from the optimal control maximum state, leaving a vector $\begin{bmatrix} 0 & 0 & f_x & f_y \end{bmatrix}^\intercal$. The optimal control signal is then defined $\mathbf{u} = \begin{bmatrix} f_x & f_y \end{bmatrix}^\intercal$.

After these offline calculations the optimal control signal for a given target has been determined for every possible state, and the controller essentially becomes a lookup table: the controller must simply find the current state and output the corresponding optimal control signal. Importantly, when the target changes the $\mathbf{P}$ matrix does not need to be recalculated, all that is needed is a new $\mathbf{Q}$ matrix weighting the states appropriately to generate $\mathbf{z}$ and then $\mathbf{u}^*$.

The operation of the implemented controller is shown in Figure 3.5.2. The environment has been divided up into a 10×10 grid that the end point moves through with a maximum velocity of 3 units. It can be seen upon careful examination of these trials that this controller is actually approximately optimal, because a straight line is not always possible when moving through the grid points. The more represented points in the operating space, the closer to a straight line the controller can achieve for all targets.

When examining this system for fine-grained control, however, it is clear that a simple 10x10 representation of the environment will not be appropriate, even for modest sized environments. There will simply be too many unrepresented locations and desired precision will not be achieved. It is also clear that detailed representation for even modest sized environments is not plausible. Assuming the arm is restrained to a metre square operating space, if even only centemetre accuracy and ten velocity

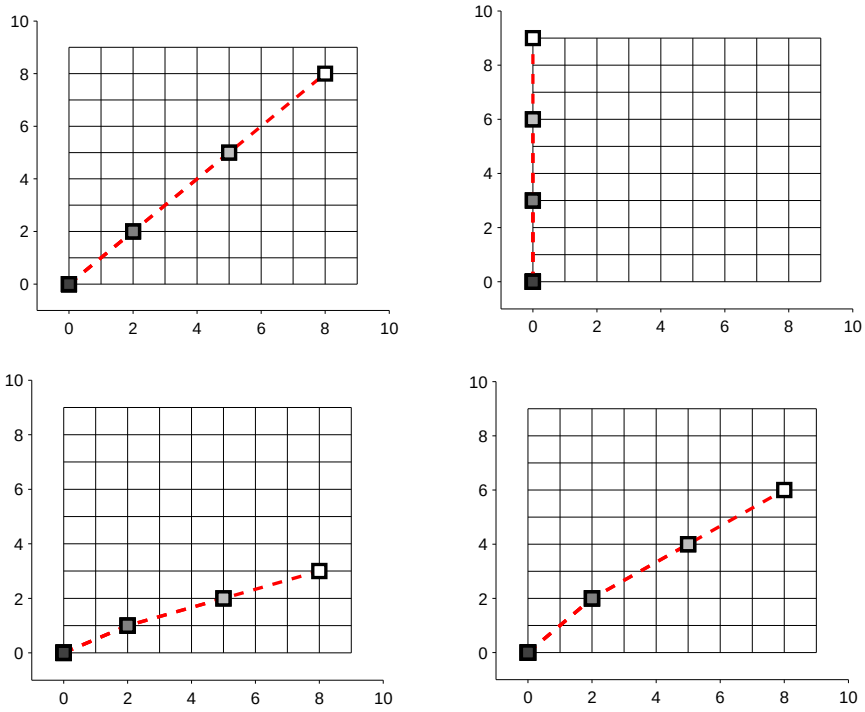

Figure 3.14: Optimal control of end effector position using a linear Bellman controller in 2D space in a 10×10 environment representation starting from $(0, 0)$ and moving to {(8,8),(0,9),(8,3),(8,6)}

values are required the $\mathbf{P}$ matrix would need to be at least size $10^6 \times 10^6$, with 10^{12} elements. It is apparent then that neither method will be appropriate for fine-grained control in large environments. This problem will be approached using dynamic scaling.

The rest of this chapter presents the practical contributions of this thesis.

3.5.3 Dynamic scaling

It is evident that the motor control system in the brain is capable of fine-grained motor control in large environments. The work presented in this section provides a method to account for exactly such control. This is achieved by using a low dimensional

representation of the entire operating space initially, and then updating the system with more detailed motor control plans as the end-effector nears the target.

These calculations can be performed quickly online for two reasons. First, the system being modeled is low dimensional so the eigenvector can be directly computed, and the operating space representation is minimal, keeping the size of the sparse **QP** matrix reasonable. The second reason is based on the observation that the passive dynamics of end-effector movement in a small operating space are a scaled version of the passive dynamics in a large operating space. This allows the system to reuse the same passive dynamics matrix to represent both, avoiding the costly recalculation of the **P** matrix. To compute the motor plan of a smaller area then all that is needed is a new state cost matrix **Q**, which is calculated based on the location of the target relative to the points in space represented.

In Figure 3.5.3 the system is shown performing in a 100 unit square 2D environment, using an eight by eight resolution representation of the operating space, with eight possible x and y velocity values. The system creates a fine-grained motor plan for the end-effector to move to the target from its initial state with three refining updates, bringing the end-effector within .05 units of the target. Using a standard fine-grained representation of the entire environment would require $2000 \times 2000 \times 8 \times 8 = 256 \times 10^6$ states to achieve this. Our system used $8^4 \times 4 = 16384$ represented states, providing a 99.9936% reduction in the number of represented states required, making this a significantly more computationally plausible method for basic fine-grained motor control in large environments.

3.5.4 Object avoidance

It is also possible, through manipulation of the **Q** matrix to encourage object avoidance. By making the costs of states in the environment blocked by objects high, the controller will tend to choose paths around the objects. It is important not to put the costs of the objects too high though, because the values of **Q** will become approximately 0, since $\mathbf{Q} = \mathrm{e}^{-\mathbf{q}}$. When this happens the random element used in MATLAB's `eigs` command during singular value decomposition becomes a determining factor in $\mathbf{z}$, causing optimal control and even consistency between trials to not be guaranteed.

To perform true object avoidance, however, there must be changes made to the passive dynamics matrix, **P**, to reflect hitting the object and not being able to move forward. All states capable of reaching any of the locations blocked by the object in

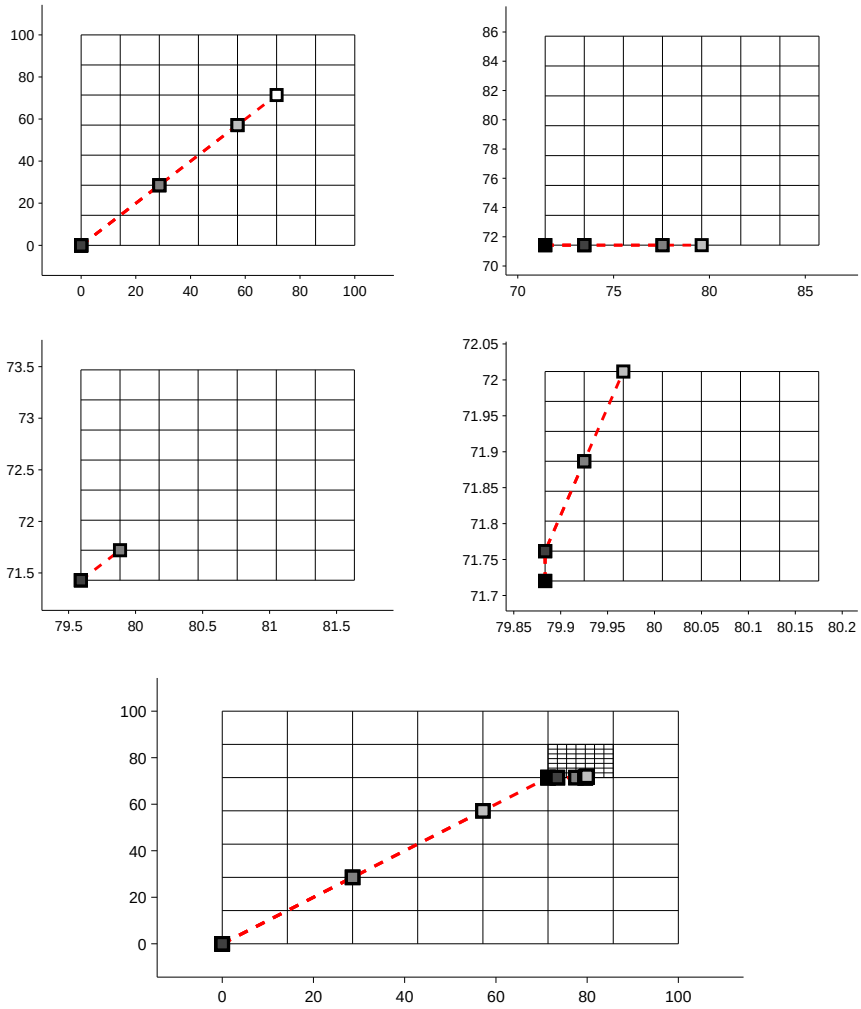

Figure 3.15: The trajectory of the end effector travelling from (0,0) to (80,72) in an environment 100 by 100 units, arriving withing a .5 unit threshold.

one timestep must be recalculated and then a **z** reflecting absolute object avoidance can be calculated from the new **QP**. If only the **Q** matrix is altered, the passive dynamics still allow movement through an object, which can be readily seen when the distance an object covers can be traversed in a single timestep. In these cases the end effector can be seen moving through, or 'jumping', an object.

An instance when this method for object avoidance would be useful would be for obstacles that will affect the overall trajectory of the system, but do not need to be directly negotiated. For example, if there are two pots of boiling water two feet apart from each other and one is reaching for a glass of not-boiling water in between, the pots of boiling water can be modeled using increased state cost values because it is

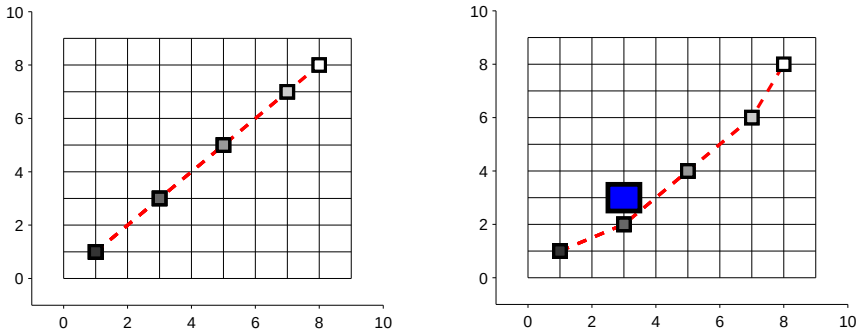

Figure 3.16: Encouraging object avoidance in the linear Bellman controller. The first figure shows the trajectory designed by the optimal controller to reach the point (8,8) from (1,1) with no objects in the environment. The second figure shows that by increasing the cost of represented points where there is

not necessary to operate closely to them. If however, the cup is directly behind one of the pots, it is then important to ensure that the system does not try to go through the pot.

3.5.5 Accounting for low level dynamics

Up to this point, the high level controller has been designed without concern for the low level dynamics. The previous work on hierarchical systems [72, 131] and the optimal control method solving the linear Bellman equation [132, 134, 135] implemented in this section have not been brought together. The work here will extend the optimal controller that solves the linear Bellman equation to account for simplified low level dynamics, laying the framework for a hierarchical system under linear Bellman control to be designed.

Recall that in the hierarchical system, when modeling the low level dynamics from the high level with an implicit controller an accurate low level state cost is guaranteed, but the low level control cost can't be explicitly known at the high level. An approximation of the low level control signal based on the high level control signal, as in Eq. (3.24), thus becomes necessary.

Previously, the matrix K was used to account for the difference between the control cost functions at the high and low level, defined in Eq. (3.26). Now, however,

the control cost functions are no longer of a form

$$r(t, \mathbf{u}) = \mathbf{u}^\intercal \mathbf{u},$$

as the control costs are defined as the logarithmic ratio between the controlled and passive dynamics transition probabilities

$$r(u(\mathbf{x}'|\mathbf{x})) = \log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}\right), \tag{3.41}$$

where time dependencies have been omitted.

Previously, the transformation K was determined by specifying the transformation between the cost functions on the two levels, here the high and low level costs will be set equal through a transformation and rewritten to solve for $u(\mathbf{x}'|\mathbf{x})$. Starting with

$$\log\left(\frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')}\right) = \log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')}\right) \tag{3.42}$$

where $p_y(\cdot)$ and $p_x(\cdot)$ represent the passive dynamics of the high and low levels, respectively, and $z_y(\cdot)$ and $z_x(\cdot)$ represent the desirability functions of the high and low levels.

Recall that in the implicit modeling method, $\mathbf{y} = h(\mathbf{x})$ is updated online, so Eq. (3.42) can be rewritten as

$$\log\left(\frac{v(h(\mathbf{x}')|h(\mathbf{x}))}{p(h(\mathbf{x}')|h(\mathbf{x}))z_y(h(\mathbf{x}'))}\right) = \log\left(\frac{u(\mathbf{x}'|\mathbf{x})}{p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')}\right) \tag{3.43}$$

which highlights the transformation between the two cost functions. For readability however, $\mathbf{y}$ will continue to be used instead of $h(\mathbf{x})$. At this point the logarithms will be discarded and $u(\mathbf{x}'|\mathbf{x})$ will be solved for

$$\frac{u(\mathbf{x}'|\mathbf{x})}{p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')} = \frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')},$$

$$u(\mathbf{x}'|\mathbf{x}) = \frac{v(\mathbf{y}'|\mathbf{y})p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')}{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')}, \tag{3.44}$$

$$u(\mathbf{x}'|\mathbf{x}) = \frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')/p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')}.$$

Formulating the right-hand side with $v(\mathbf{y}'|\mathbf{y})$ alone on top makes it so that only the lower term requires normalization and helpful for clarity later on.

Now that a function to account for the low level control cost using the high level control signal has been generated it will be plugged back into an early version of the linear Bellman equation, Eq. (3.33), resulting in

$$-\log(z_y(\mathbf{y})) = q_y(\mathbf{y}) + \min_v \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[\log \left(\frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y}) z_y(\mathbf{y}') / p_x(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}')} \right) \right] \right\}. \tag{3.45}$$

Now the denominator of the minimization needs to be normalized by

$$\mathcal{G}[z](\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x},} \frac{p(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}')}{p(h(\mathbf{x}')|h(\mathbf{x})) z_y(h(\mathbf{x}'))}, \tag{3.46}$$

and so the multiplying and dividing by $\mathcal{G}[z](\mathbf{x}, \mathbf{y})$ gives

$$\begin{aligned}
-\log(z_y(\mathbf{y})) &= q_y(\mathbf{y}) + \\
&\quad \min_v \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[\log \left(\frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y}) z_y(\mathbf{y}') / p_x(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}') \mathcal{G}[z](\mathbf{x}, \mathbf{y}) / \mathcal{G}[z](\mathbf{x}, \mathbf{y})} \right) \right] \right\}, \\
&= q_y(\mathbf{y}) + \\
&\quad \min_v \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[-\log\left(\mathcal{G}[z](\mathbf{x}, \mathbf{y})\right) + \right. \right. \\
&\qquad \left. \left. \log \left(\frac{v(\mathbf{y}'|\mathbf{y})}{p_y(\mathbf{y}'|\mathbf{y}) z_y(\mathbf{y}') / p_x(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}') / \mathcal{G}[z](\mathbf{x}, \mathbf{y})} \right) \right] \right\}, \\
&= q_y(\mathbf{y}) + \\
&\quad \min_v \left\{ E_{\mathbf{x}'\sim u(\cdot|\mathbf{x})} \left[-\log\left(\mathcal{G}[z](\mathbf{x}, \mathbf{y})\right) + KL \left(v(\mathbf{y}'|\mathbf{y}) \,\middle|\middle|\, \frac{p_y(\mathbf{y}'|\mathbf{y}) z_y(\mathbf{y}') / \mathcal{G}[z](\mathbf{y})}{p_x(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}') / \mathcal{G}[z](\mathbf{x})} \right) \right] \right\}, \\
&= q_y(\mathbf{y}) + -\log\left(\mathcal{G}[z](\mathbf{x}, \mathbf{y})\right) + KL \left(v(\mathbf{y}'|\mathbf{y}) \,\middle|\middle|\, \frac{p_y(\mathbf{y}'|\mathbf{y}) z_y(\mathbf{y}') / \mathcal{G}[z](\mathbf{y})}{p_x(\mathbf{x}'|\mathbf{x}) z_x(\mathbf{x}') / \mathcal{G}[z](\mathbf{x})} \right)
\end{aligned} \tag{3.47}$$

Now it is possible to define the high level optimal control signal that accounts for the low level dynamics. Because $v(\mathbf{y}'|\mathbf{y})$ is one side of a KL-divergence, it must be

set equal to the otherside to produce the global minimum,

$$v(\mathbf{y}'|\mathbf{y}) = \frac{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')/\mathcal{G}[z](\mathbf{y})}{p_x(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')/\mathcal{G}[z](\mathbf{x})}. \tag{3.48}$$

With the KL-divergence equal to zero, Eq. (3.47) becomes

$$-\log(z_y(\mathbf{y})) = q_y(\mathbf{y}) - \log\left(\mathcal{G}[z](\mathbf{x},\mathbf{y})\right), \tag{3.49}$$

which rewritten in terms of $z_y(\mathbf{y})$ becomes

$$z_y(\mathbf{y}) = \mathrm{e}^{-q_y(\mathbf{y})}\mathcal{G}[z](\mathbf{x},\mathbf{y}) = \mathrm{e}^{-q_y(\mathbf{y})}\left(\frac{p_y(\mathbf{y}'|\mathbf{y})z_y(\mathbf{y}')}{p(\mathbf{x}'|\mathbf{x})z_x(\mathbf{x}')}\right). \tag{3.50}$$

Or in vector notation,

$$\mathbf{z}_y = \mathbf{Q}_y\left(\frac{\mathbf{P}_y\mathbf{z}_y}{\mathbf{P}_x\mathbf{z}_x}\right) = \mathbf{Q}_y\left(\frac{\mathbf{P}_y}{\mathbf{P}_x\mathbf{z}_x}\right)\mathbf{z}_y, \tag{3.51}$$

which can again be solved for using MATLAB's `eigs` command for low dimensional systems, and otherwise approximated using techniques in [132], as previously discussed.

Now to determine the optimal linear Bellman controller on a high level that accounts for low level dynamics, the low level linear Bellman equation must be solved, using the implementation method discussed above. The $\mathbf{P}_x$ and $\mathbf{z}_x$ terms are then used in the computation of the high level's desirability function $\mathbf{z}_y$, which can then be used to determine the high level optimal control signal $\mathbf{v}$.

In this arm model, no simplification of the lower level passive dynamics were necessary, because the low level is equally easy to solve for using a linear Bellman controller. In hierarchical systems with dimension reduction between low and high levels, however, it is necessary to use an approximation of the low level dynamics and a reduced system state representation. While this does not exactly capture the dynamics of the low level, it allows an approximation to be incorporated, increasing the efficiency of the high level controller at the low level. This approximation of low level dynamics becomes important even in arm models with 7 degrees-of-freedom in the joint space representation. The reason that a linear Bellman controller can be used for arm control is because it is possible to opearate on a simple abstraction of the arm model. Requiring the joint space to be solved for with a high dimensional representation would render the benefits of a hierarchy useless.

Of note in this method of accounting for low level dynamics is that the low level's state cost is embedded in the low level desirability function $\mathbf{z}_x$. This leaves the high level state cost function the freedom to be determined soley with respect to high level optimization. This is in contrast with the hierarchical system described in [131] where the high level state cost must explicitly replicate the low level state cost function.

3.6 Conclusions

This chapter has presented the control theory relevant to the model of neural motor control being developed in this thesis. Importantly, the main focus of this chapter has been on hierarchical control models and the linear Bellman controller, which were brought together in the last section of this chapter.

In the next chapter, a biologically plausible model of motor control will be presented which combines the results of this and the previous chapter.

Chapter 4

NOCH: A framework for biologically plausible models of neural motor control

This chapter brings together the previous result to present the main theoretical contributions of this thesis: A framework for a biologically plausible model of motor control based on hierarchical optimal control theory, the Neural Optimal Control Hierarchy (NOCH).

First, the overall system model diagram is presented and a system walk-through detailing the flow of information will be given. The explicit implementation of the mathematical model developed in the last chapter within this framework is then discussed. Next, a more detailed system model of volitional control of movement and motor learning will be discussed. Finally, outstanding issues yet to be addressed by this model and oversimplificantions that require reworking are then be discussed, and a comparison to past models of neural motor control is made.

4.1 Overall system diagram

In Figure 4.1 a block diagram of the NOCH model is presented. The relationships between the various areas are represented numerically, roughly corresponding to the order of the flow of information throughout the system. Importantly, this chronology should not be considered a constraint because of the massive parallellism found in the brain.

Briefly, the system model works as follows. The PM and SMA act as a high level controller, determining the optimal actions for the system to carry out during the task. The passive dynamics models used by the PM and SMA are retrieved from the cerebellum, while at the same time information relevant to the immediate task (such as target and sensory information from the environment) are sent to the basal ganglia for state cost definition. The passive dynamics and state cost information are integrated in the PM and SMA and a high level optimal motor command for the system is generated. The signal is then sent to the cerebellum and M1. In the cerebellum the high level command is converted into a low level control signal using the internal models kept in the cerebellum. The low level control signal is passed to the primary motor cortex which sends out execution commands to the descending systems. At the same time, M1 has recieved the high level command from the PM and SMA, which allows a relationship to be built up in M1 connecting the high and low level control signals. Eventually, this allows for the immediate execution of practiced actions without waiting for the calculations performed in the cerebellum.

System feedback is monitored at both the low and high level through the cerebellum and the PM and SMA, respectively. From the brain stem, the sensory feedback information is relayed to S1 where it is incorporated with motor command signals from M1 to produce a system state signal. The cerebellum monitors the descending motor command and system state signal in S1 to ensure both that the motor action executes properly and that its internal models are correct. When a deviation is detected the cerebellum attempts to correct the error by incorporating a corrective low level signal into the command signals sent to M1. If the correction is consistently required, the internal model in the cerebellum is adapted to include it, or a variation of the current model is created. The feedback loop through the PM and SMA uses sensory information transformed by S1 into a high level state representation. In the PM and SMA, large deviations can be accounted for and environments that are poorly modeled can be negotiated by incorporating visual feedback.

The functions of and relationships between the components of this system will now be discussed in more detail.

1 - System initiation

It has been shown that the PMd is directly involved in actions which are triggered by external stimuli [139, 143], and that the SMA is necessary for the spontaneous generation of movement [112, 135]. Here they are modeled as part of the same high

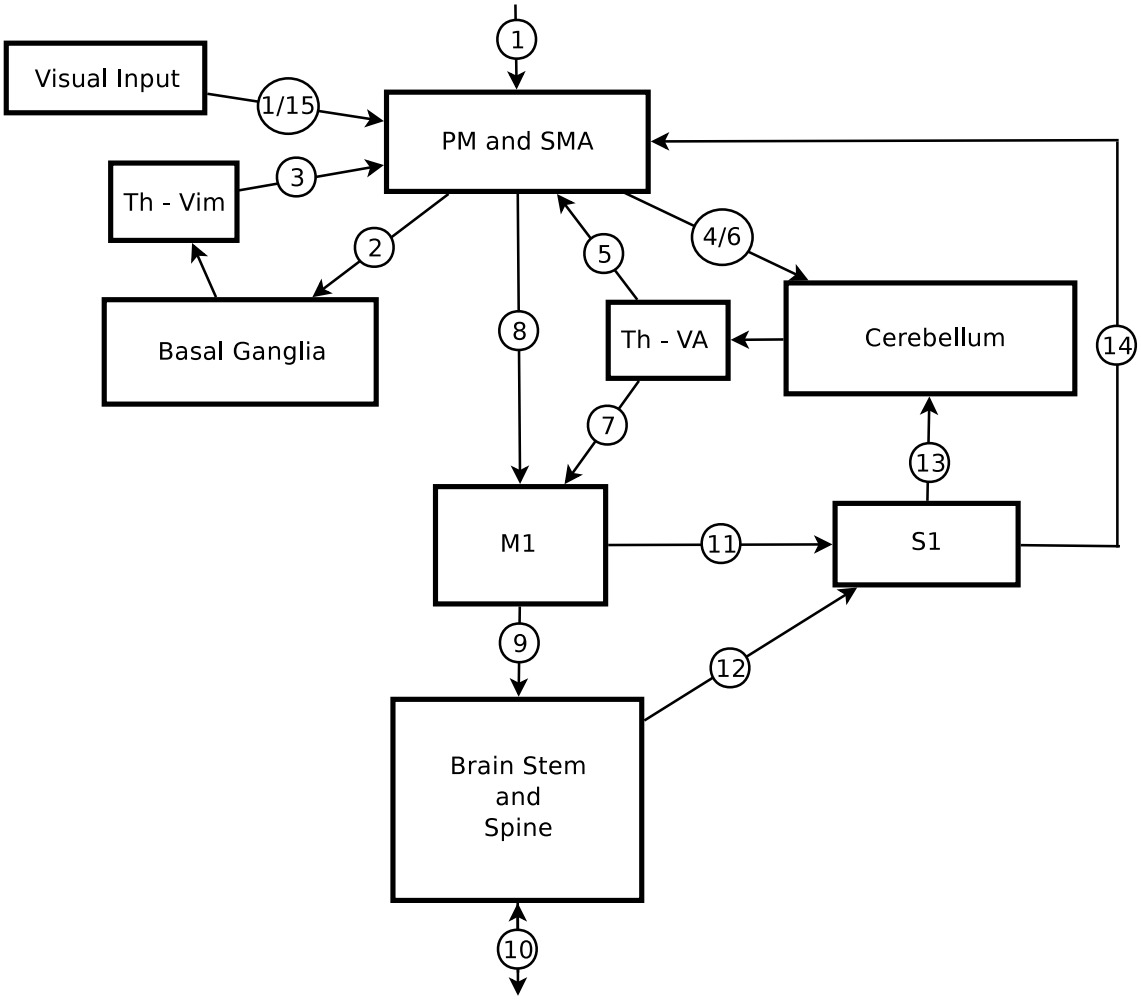

Figure 4.1: The overall system diagram. PM: Pre-motor areas. SMA: Supplementary motor areas. Th-Vim: Ventral intermediate thalamus. Th-VA:Ventral anterior thalamus. M1: Primary motor cortex. S1: Primary sensory areas.

level control component for the sake of simplicity. The PM and SMA integrate sensory information from the environment and current task goals to determine an appropriate high level system model. If, for example, the task involves kicking a soccerball, the high level system model used could be expected to be different than those used while driving. In the cases of arm reaching, potential high level system models would be end effector movement in 3D space or torque at the shoulder, elbow, and wrist in joint space.

2 & 3 - State cost information

The basal ganglia receives task relevant information from the PM and SMA, and are responsible for state cost calculations. This involves examining the task completion timeline and appropriately scaling motor actions as well as determining the acceptable amount of execution error. After the computation is completed, the basal ganglia relay the state cost information back to the PM and SMA through the ventral intermediate thalamus. While definitely not a complete representation of the basal ganglia's function in the motor system, interacting with the PM and SMA exclusively is not unrealistic. It has been shown through neuroanatomy tracing experiments that the basal ganglia projects preferentially to the SMA [122].

A disruption of the function of the basal ganglia could then lead to system errors such as an excess of unwanted movements or supression of voluntary movement. If an equality among different states near the target were assumed, then a continuous fluctuation between them would be expected, as is seen in Parkinson's disease. Conversely, if states are judged to be too expensive to justify movement there would be a marked decrease in motor actions, as is seen in Huntington's disease.

4 & 5 - Passive dynamics information

At the same time as the state cost information is being obtained from the basal ganglia, the PM and SMA are sending system model information to the cerebellum to retrieve an appropriate passive dynamics model. The cerebellum is widely regarded as a site for the storage of internal models [13, 47, 79, 107, 145] and so it would be expected that most any system dynamics model required by the PM and SMA would be acquired from this area. However, for heavily used high level models, such as end effector movement in 3D space under the effect of gravity, it would not be unreasonable to expect the passive dynamics to be stored inside the PM and SMA themselves. As a model of end effector movement can be used to represent a body part, the point of a tool, or even things unrelated to motor control such as a baseball flying through air it would be efficient for the system to keep such a model readily accessible.

At a slightly lower level, having models of the system dynamics for commonly used systems stored in the PM and SMA would also be expected of systems that give rise to phenomena such as mirror neurons. In the conversion of visual information to motor actions direct access to appropriate system models would seem necessary, allowing a particular technique to be retained in memory at the high level to work

through the low level implentation later.

The passive dynamics information in this system then is actually obtainable in two places, depending on the system being controlled. Well practiced systems would be quickly loaded from local storage in the PM and SMA, while less commonly used systems would require communication with the cerebellum, which would then send the information back through the ventral anterior thalamus. In the above system diagram, however, only the cerebellar loop is displayed for simplicity.

6 - High level control signal

After the PM and SMA have obtained the state cost and passive dynamics information, the high level control signal is computed. The process of generating the high level control signal can be repeated indefinitely before, or even without, sending information to the low level controllers. Inside this loop, the system models can be increased in complexity with every iteration, developing more and more specific motor plans. This sort of high level offline iteration can account for the priming of actions, where it has been seen that specific aspects of upcoming movements are calculated before the movement actually begins [53]. In these situations there would be two types of initialization commands, the 'get ready' and 'go' commands.

This repetition without execution would also be expected of a system capable of mental rehearsal of actions. In this way the PM and SMA can act as an explicitly modeled, autonomous, high level controller operating independently of the actual state of the system. Computing solutions offline has long been a staple of optimal control theory [9], and indeed mental rehearsal has been shown to have beneficial effects on human motor performance [100].

Once a high level signal is generated that is to be executed by the system, the PM and SMA send it to the cerebellum.

7 - Low level control signal

In addition to providing high level passive dynamics models to the PM and SMA, the cerebellum is also responsible for converting a given high level control signal to a low level control signal for M1 to execute. The cerebellum must take into account the body's inertia and external forces such as gravity to match the high level command. The cerebellum essentially uses the high level command as a map detailing where to

move when the system is in a given state, and is responsible for computing how to move the body where it is supposed to go. For example, if the arm is being held out in front of the body and the high level command is to move left one inch, different muscle activation forces are required than if the arm is swinging to the right and recieves a command to move left. The internal models in the cerebellum are critical to proper function and coordination of the body during movement, and are constantly being corrected for increased accuracy.

The ability for the system to perform optimally is explicitly dependent on the accuracy of these models. If the high level model is innacurate, the generated high level commands will be implausible for execution on the low level. Similarly, if the low level model is incorrect it will be unable to accurately implement the specified high level command.

Once a low level motor plan matching the high level control signal has been created, the cerebellum sends it to M1, through the ventral anterior thalamus, for execution. Recent neuroanatomy tracing studies have shown that this pathway projects preferentially to M1 [122], supporting this explicit communication with M1.

8 - Associating M1 and high level commands

At the same time as the PM and SMA send the high level motor plan to the cerebellum to create a low level motor plan, the high level signal is also sent to M1. Although initially this signal from the PM and SMA may be of no use to M1, as it continues to recieve the corresponding low level commands from the cerebellum, with practice, it will be able to associate high and low level control signals. In this way, for practiced movements in normal conditions, the processing performed by the cerebellum can be skipped and the PM and SMA can send high level commands directly to M1 for immediate execution.

As associations are also formed in the PM and SMA between desired motor actions and the corresponding high level motor plan, an incoming initialization signal will end up being essentially relayed to M1 for execution without any processing required. These movements are referred to as expert actions and are performed automatically by the system. Expert actions and volitional control of movements and motor learning will be discussed further later in this chapter.

9 - Movement through synergies

The structure of M1 has been compared to that of the visual processing areas; as a hierarchy of layers where the information represented is increasingly complex as the hierarchy is ascended. In the context of a motor system, this suggests that increasingly complicated motor actions across space-time are represented at each layer. These representations are referred to as synergies. Synergies allow M1 to call upon a single high level area representing a complicated action instead of seeking out the individual components and coordinating them appropriately, greatly reducing the required processing. An example of a synergy would be the hand positions for a pianist playing the A major chord, or the sequence of movements required to play a trill. Activating different synergies in M1 simultaneously would then give rise to a mix of the two actions, defined by the synergy activation weightings. Additionally, if there is a motor synergy that approximates the solution, it can be activated with M1 applying the appropraite corrections at a lower level such that the total M1 activation achieves the desired movement.

10 - Motor command execution and feedback

The descending motor command signal is sent down corticomotor neurons in M1 and through the brain stem to the motor neurons in the spinal cord where muscles are activated directly. Sensory receptors then relay information back up the spinal cord to the brain stem, which identifies the task-relevant information to be sent to higher processing areas.

11 & 12 - Process feedback information

Both feedback loops in this system are based off of sensory information processed in S1. Task-relevant sensory feedback information from the brain stem is relayed from the brain stem to S1, where it is incorporated with current muscle activation information obtained from M1 projections. This information is used to generate an accurate estimate of the system state at the low level, which is used by the cerebellum. After the low level system state has been defined, S1 then transforms it into a high level system state representation to be sent to the PM and SMA.

13 - Cerebellar feedback loop

Here the low level system state information from S1 is used to calculate the divergence from the desired system state. If it is determined that external forces are hindering performance or that an internal model is incorrect, a corrective signal is generated and incorporated into the command signal sent to M1. In the case that external forces are being applied, such as wind when shooting a basketball, an internal model will be developed to account for these circumstances. If there are no interfering external forces and it is determined that an internal model is incorrect, the model will be adapted to include the corrective signal. When successfully integrated, corrective signals will no longer be required through this loop because low level motor commands sent to M1 from the cerebellum in the future will properly account for these errors.

14 & 15 - PM and SMA feedback loop

The second feedback loop is directed through the PM and SMA, and is responsible for the larger, and slower, corrections to the motor plan. In this loop the high level system state feedback from S1 is incorporated with non-proprioceptive sensory information, such as visual or auditory feedback, into the motor plan corrections. The corrections implemented at this level affect the system the most, because an abstraction of the actual system is being manipulated. While the PM and SMA might be altering the end effector position only slightly, the resulting changes to the muscle activations required to implement this change could be quite large.

In this loop, the PM and SMA acts as an implicitly modeled controller, with direct access to the current system state. Under the control of the PM and SMA feedback loop, the motor system operates much slower, but can incorporate non-proprioceptive sensory feedback into the ongoing motor commands. This type of control can be thought of as when a beginner piano player must carefully look at their hand and place each finger in the correct position one at a time to form a chord.

Here, the PM and SMA explicitly calculates the system error and sends the cerebellum corrective signals at every time step. The cerebellum converts the high level command to a low level command, passes it on to M1, which executes the action. Feedback is then generated and sent to S1, which transforms it to a high level signal, and the process is then repeated. As this course of actions occurs, new synergies are developed in M1 until the system is quickly and accurately able to execute the high level command. Once this happens, the PM and SMA return from acting as an im-

plicitly modeled high level controller to only performing their duties as an explicitly modeled high level controller.

4.1.1 Mapping onto a hierarchical system of controllers

Explicity mapping this system model onto a hierarchal system with a high level linear Bellman controller, as described in the previous chapter, proceeds as follows. The PM and SMA send the information necessary to calculate the high level state cost matrix, $\mathbf{Q}$, to basal ganglia, which relays $\mathbf{Q}$ back to the PM and SMA through the ventral intermediate thalamus. The PM and SMA then retrive the high level passive dynamics matrix, $\mathbf{P}$, from the cerebellum or a local storage location. The two matrices are combined to create the $\mathbf{QP}$ matrix, and the desirabililty function, $\mathbf{z}$, is calculated by solving for the primary eigenvector of $\mathbf{QP}$. Once $\mathbf{z}$ is defined, the high level optimal control signal $\mathbf{v}^*$ can be calculated using Eq. (3.36), and the two are sent to the cerebellum.

The cerebellum is responsible for converting the high level motor command, $\mathbf{v}^*$, into a low level motor command, $\mathbf{u}$ by using quadratic programming as before, solving Eq. (3.16). This low level command is then sent to M1, which realizes the most efficient implementation of $\mathbf{u}$ through the available synergies. The resulting signal is then sent to interneurons in the spine through the corticomotor neurons in M1 and the brain stem.

The sensory feedback is filtered appropriately by the brain stem and sent to S1, where it is incorporated with the current low level command $\mathbf{u}$. This information is sent to the cerebellum, which uses the difference between the desired and actual states to create a low level error signal and corrective command which is incorporated into the current movement commands sent to M1.

Here, there is no difference between the sensory information available to the cerebellum and PM and SMA, because the exact system state is relayed, there is no proprioceptive or visual sensory feedback difference at the levels. As a result the high level controller does not switch to an implicit controller and take over, because there is no additional information to add to the control of the system. If the error is consistent and large, however, it is possible for the cerebellum to provide the PM and SMA with a modified high level model, which can then be used to regenerate the high level control and desirability function and hopefully improve system performance.

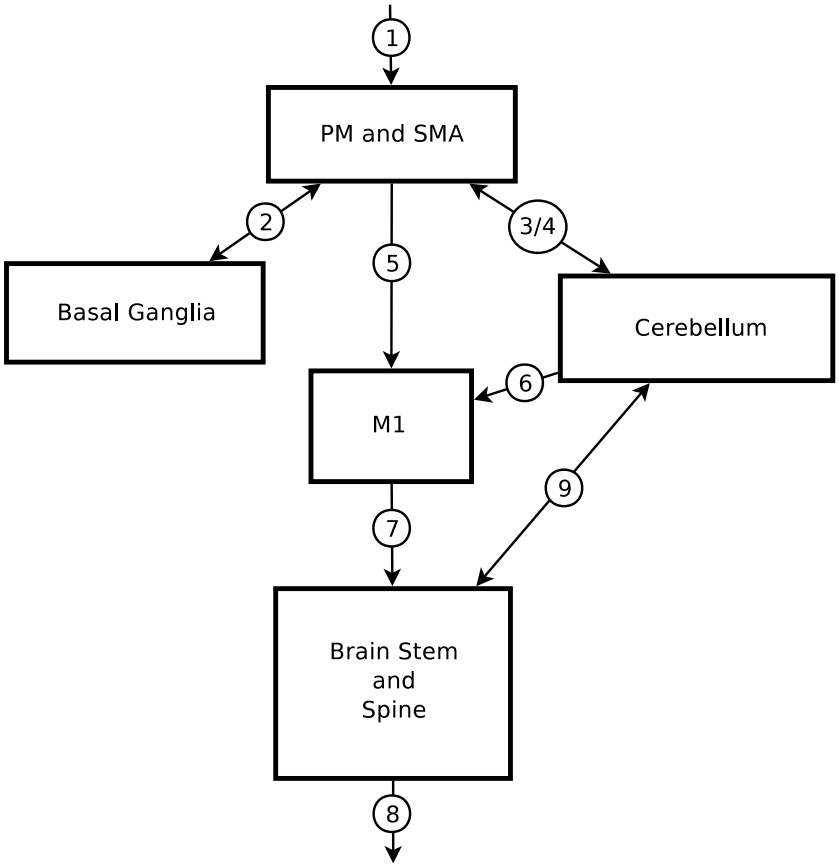

Figure 4.2: The conscious control of movement model, pathways to M1 and the PM and SMA from basal ganglia and cerebellum through thalamus areas suppressed for simplicity. PM: Premotor areas. SMA: Supplementary motor areas. M1: Primary motor cortex.

4.1.2 Volitional control of movement

During any volitional movement, there are many body control commands being issued outside of our awareness. These commands perform functions such as maintaining our balance, gait, or any other of a myriad of actions that require no conscious effort. It is important that these functions remain outside of our conscious experience, or our overall motor performance would suffer greatly from trying to control too many facets of movement at once.

Figure 4.1.2 depicts a system block diagram that focuses on a section of the framework not depicted in Figure 4.1. Here it is presented as the structure and flow of information through the motor control system when a person is intentionally controlling an aspect of their movement and the rest of the body is being subconsciously

controlled. As the pathways through the VA and Vim sections of the thalamus from the basal ganglia and cerebellum to the PM and SMA and the cerebellum to M1 have already been specified by the previously diagram, they are left out of Figure 4.1.2. It is important to note, however, that the projections from the cerebellum to the brain stem and back are direct, and not relayed through the thalamus.

Briefly, the system diagram above can be understood. The system is initialized and the PM and SMA interact with the basal ganglia and cerebellum to obtain the state cost and passive dynamics information for a high level model. At this point, the task-related movements that do not require conscious control are sent directly to M1 from the PM and SMA, and the cerebellum receives a high level command detailing the movement of the volitional action. The cerebellum then generates a low level command which is sent to M1, where it is incorporated into the synergies activated to implement the high level signal from the PM and SMA. This descending signal is then executed through the motor neurons in the spine and system feedback is gathered. The cerebellum additionally recieves another high level, abstract command from the PM and SMA detailing the desired state of the body overall, such as 'maintain walk', or 'stand still'.

The sensory signals from the body are projected up the spinal cord to the brain stem where they are filtered into task-relevant and task-irrelevant feedback. The task-relevant feedback is sent to S1, to be processed and incorporated into the system as previously discussed. The task-irrelevant information is not discarded, however. Instead it is sent directly to the cerebellum. The cerebellum, then, is responsible for maintaining the specified body position or state while the motor actions being controlled by the PM and SMA are carried out. It accomplishes this by modulating the descending systems through its direct connections with the brain stem, where the appropriate signals are generated using the internal models stored in the cerebellum.

1 to 5 - Generate the high level control signal

Similar to the system diagram in the last section, the first steps are the generation of the high level command signal. Here, however, the PM and SMA generates two extra set of signals. The first set of signals details actions to be implemented by M1, which have been learned by M1 associating the high level signals from the PM and SMA with the low level signals from the cerebellum. The second set of signals detail the desired state of the body during movement, and are sent to the cerebellum.

For example, a set of commands from the PM and SMA while playing (bad) piano would say to the cerebellum: "Put the index finger of the right hand on middle C, while maintaining this upright posture." To M1 the cerebellum would say: "This is called 'putting the index finger of the right hand on middle C' *and* in the left hand play this calypso bass pattern that was previously learned."

6,7 & 8 - Execute motor commands and recieve sensory feedback

Once the high level command projected to the cerebellum has been converted to a low level command ready for implementation in M1, the signal is incorporated with the ongoing actions specified by the PM and SMA and sent directly to M1. The commands need not be directed to unrelated areas of the body, either, such as the right and left hands. For example, the PM and SMA could issue a command to play an A major chord in the right hand, which would be sent directly to M1. At the same time, however, a signal to the cerebellum would be sent saying: "Once the A major chord is played, slip the middle finger off of the C# key to the C key to make the chord minor." In this way new synergies can be built off of already existing synergies, and after practice, this action will be able to be executed without any processing in the cerebellum.

Once the muscle activation command has been created in M1, it is sent down to the motor neurons through corticomotor neurons and the brain stem to be executed. The sensory feedback then travels up the spine to the brain stem. Here the information relevant to the task being carried out is separated out and sent to S1. The remaining information is then passed along to another section of the cerebellum, separate from that implementing the high level motor command from the PM and SMA.

9 - Unconscious control of the body

In the cerebellum the state of the entire body is considered while motor actions are being carried out, and the appropriate commands to maintain the desired system state are issued as necessary. The state to maintain is specified by a command from the PM and SMA. Using the system sensory feedback and internal models an error signal is generated, and the cerebellum sends out corrective signals back to the brain stem directly, where they are incorporated with the descending motor command and carried out by motor neurons.

4.2 The framework as a whole

Importantly, the systems described are not independent. They have been separated in this presentation to aid understanding, but are a single, unified system. Together, they provide a framework which may form a foundation for future research into the motor control system in the brain.

The remainder of this chapter presents empirical support for this framework, novel predictions about the function of the system components and expected impairments with damage to the system, and the research directions of future work.

4.3 Empirical support for the NOCH framework

As this model is based off of neurobiological research on the motor control system in the brain, there is already a standing body of empirical support for this framework. Here some of the key points from the neurobiology presented before will be briefly reviewed:

1. The M1 motor map is more than a straight mapping to controlled muscles. Instead it is thought to form a hierarchy of synergies akin to the visual processing representations which increase in complexity as the hierarchy is ascended [12, 19, 20, 120].

2. Models of M1 output that assume muscle activation signals are generated in M1 are able to account for M1 neural activity in detail [119].

3. When the PM and SMA are damaged, high level control of movement is jeopardized to the point that unwanted complex movements are conducted [16]. Additionally it has been observed that voluntary control of sequential movements and sequential performance of multiple movements is severely impaired [109], all consistent with this area acting as a high level controller.

4. The output activity of the basal ganglia, which reduces or increases the inhibition of thalamic areas projecting to the PM and SMA, could be used to identify target states.

5. Parkinson's and Huntington's diseases of the basal ganglia are explainable by assuming state cost assignment is performed in the basal ganglia.

6. The basal ganglia activity correlates with amplitude, velocity of high level movement independent of muscle activaiton [24], as would be expected of a component of the high level control system.

7. The basal ganglia and the cerebellum have both been found to be critical to the timing of movements [69]. The NOCH provides a very straightforward explanation for these observations in intentionally controlled movements. The basal ganglia determines the state cost function used to determine the high level optimal controller; this directly controls the scale of movement and also affects the timing of movements strung together in sequence.

8. Somatosensory maps have been found in the spinocerebellum and deep nuclei, necessary for issuing low level commands directly to the brain stem [45].

9. The vermis of the cerebellum is involved in the control of posture, balance, and locomotion of the body [45].

10. Cerebrocerebellum has shown to be highly active during movement planning and mental rehearsal of actions [41], also supporting a dual role in high level movement control.

11. The cerebellum transforms the high level signals from the PM and SMA into low level motor commands to be carried out by M1, and issues them appropriately based off of the high level information, thus directly controlling the timing of movements.

12. The cerebellum's involvement in the execution of practices action is supported by research performed on monkeys with cerebellar damage revealing that online modification to motor plans was severely impaired [4].

In sum, the NOCH model is consistent with a wide variety of empiracal evidence from the motor system.

4.4 Predictions

The structure and flow of information in the motor control system as hypothesized by this framework generates testable predictions about the motor behaviour to be

found in patients with lesions or damage to a particular area of the motor control loop. A number of the predictions correspond to observed phenomena already found in the literature. Here, however, several novel, testable predictions generated by this framework are briefly presented.

High level internal models stored in the PM and SMA

It might then be expected that high level internal models, or perhaps more likely, frequently used high level internal models, are not kept in the cerebellum. Rather, one of the predictions of this framework is that there is an area in the PM and SMA dedicated to storing high level internal models. This would reduce the amount of information passed from the cerebellum during execution, as well as reduce its responsibilities. Supporting this idea is low amount of cerebellothalamic projections recieved by the PM and SMA, which would have to transmit the passive dynamics information. The implications of the PM and SMA storing the passive dynamics information include that the PM and SMA must house internal models. This area might also be expected to overlap with the site where mirror neuron activity occurs.

Pathways corresponding to M1 and BS circuits through CB

According to the volitional control of movement model discussed above, a prediction of the structure of the projections to and from the cerebellum is that the PM and SMA project to distinct areas of the cerebellum, one which recieves input from S1 and projects out to M1, and another with reciprocal connections to the brain stem.

Damage to PM and SMA

Due to the fact that visual information enters the model through the PM and SMA, it would be expected that damage to this area would significantly impair the patient's ability to incorporate visual information into their motor actions. While their complex movements would also be severely affected, patients who retained the ability to voluntarily walk would be expected to find it difficult to avoid obstacles.

Damage to BG

Based on the basal ganglia's assumed role in high level model state cost assignment, it would be predicted that the functions the PM and SMA would require such infor-

mation for would experience marked deficiencies. For example, it would be expected that mentally navigating through an obstacle laden area, or new environments, would be significantly more difficult after damage to this area.

Damage to CB

It would be expected that learned movements would be executed with minimal error under normal operating conditions, however performance would be lost as soon as changes were applied. For example, a subject sitting in a still chair might be able to touch his nose on command, but if the chair is being spun it would be expected that they would not be able to complete the action because they would not be able to account for the external effects.

Additionally, the ability to unconsciously control body states or patterns during movement while focusing on another action would be severely impaired. For example, walking and chewing gum might prove a difficult task, depending on how good the subject is at chewing gum, because the focus on walking can never be removed if the subject is to continue walking. Rather, they must be able to focus on maintaining the body state and the motor actions being performed simulatenously.

Where it has been proposed that both the basal ganglia and the cerebellum are directly involved in the timing of movements, this model predicts that the cerebellum is not involved in timing of practiced movements. In such movements, where the commands are projected directly from the PM and SMA to M1, circumventing cerebellar processing, it would be expected that cerebellar damage would not affect motor performance.

Damage to PM and SMA to M1 connections

If the connections between the PM and SMA and M1 were lesioned, it would be expected that complex actions would be reduced to only those which can be focused on simultaneously, similar to the damage to the cerebellum impairing the maintainance of body state and motor actions at the same time. For example, a patient might still be able to play piano with their right hand, but keeping a previously learned calypso pattern playing in their left hand at the same time would prove to be a very taxing action, whereas before damage to the area it might have required little effort. Now the cerebellum must process every high level command given, as opposed to the PM and SMA evoking the learned relationship between high and low level commands in

M1.

Damage to CB to M1 connections

Damage to the connections between the cerebellum and M1 would be expected to significantly impair focused control of movements and motor learning. The system converting high level to low level commands would be removed from the loop, able to maintain the desired body state but unable to issue new commands to M1. Complex actions would be reduced to those which had previously had connections formed between the PM and SMA and M1. Additionally, it might be expected that after extended periods of unuse, weakening and disappearing connections between the PM and SMA to M1 would be unable to be revived, decreasing the repetoire of available complex actions.

Damage to M1

Although damage to M1 would essentially end most voluntary motor actions, it would be expected that if this framework is correct pepole with severe M1 damage should still be able to balance themselves and perform very basic voluntary motor action patterns, although the definition of what exactly the repetoire of these actions would be is unclear.

4.5 Future work

Building on the research presented here there are many possible directions for future research. Some of the most exciting include examining how removing or 'damaging' components of the system affects overall performance, and looking for correlations with data from patients with corresponding damage. Another exciting direction will be the implementation of a motor control system through this framework where multiple limbs and the body state are controlled simulataneously for a simple simulated body.

The framework presented here, while accounting for many of the characteristics of motor control, makes numerous simplifications to the actual control system structure in the brain. There are also several inconsistencies with biological function that are yet unresolved. This section will identify some of the issues that could be addressed

by this framework in the future.

4.5.1 Biological considerations

Further dissection of system components

Currently the components in the system are grouped together at a very high level. One of the main areas of focus for future research will be to further divide up these components more appropriately based on function and neuroanatomy. For example, the PM and SMA component encompasses a very large area of cortex with a multitude of functions. The next stage of study will look at determining the interconnection and function of the PMd, PMv, SMA, and pre-SMA which make up the current component.

BG for high level model correction and reward

The basal ganglia has been identified as being involved in reward-based activity [114, 49], and suggested to provide a reward signal according to system performance. An additional role for the basal ganglia in this model then would be working in conjunction with the CMA, which is currently thought to be involved in reward-based processing [105], to appraise action execution at a high level and correct the high level models kept in the PM and SMA appropriately. This is in agreement with the Series Hypothesis of basal ganglia function proposed in [138], which suggests that the basal ganglia is responsible for input dimensionality reduction and action selection. It will be important to examine the dynamics and plausibility of correcting the high level models through the basal ganglia.

One possible way to test this would be through examination of patients who have undergone a thalamotomy, for treatment of Parkinson's disease symptoms, specific to sites recieving projections from the basal ganglia or cerebellum and projecting to the PM and SMA. If the basal ganglia is responsible for correcting the high level models then it would be expected that removing the cerebellar input to PM and SMA would affect high level model improvement significantly, and less than when basal ganglia related thalamic sites are removed.

Corticomotor neurons in PM

In this framework, M1 is the only area modeled with CM neurons. In reality, the PM has been found to have at least as many CM neurons [101]. These CM neurons in the PM project mainly to the more distal muscles, such as those of the hand. Nearby areas in the SMA have also been shown to be highly involved in the control of hand formation [81]. Studies have correlated PM areas with fine-grained control of hand movement, such that the activity increases with higher precision hand control [91]. However these areas act to modulate the activity of the CM cells in M1, rather than those of the PM [106], so it currently remains unclear as to the specific function of the CM cells in the PM. Further research will be required to integrate possible theories of function of these cells into the current framework.

Connections yet unaccounted for

The brain is an incredibly interconnected and complex system, and the projections described in this model are a drastic, but necessary, simplification of the neuroanatomy. There are many connections that are yet unaccounted for by this model. Here, several of the major ones and their potential functions are briefly discussed.

S1 to M1: There are strong reciprocal connections between S1 and M1 not modeled in this framework. Here, only the projections from M1 to S1 are accounted for. One of the possible functions of the connections from S1 to M1 would be to inform M1 of the action resulting from the activation of a particular area. Additionally, it has also been observed that a lack of input to M1 from the sensory neurons associated with a particular area induces a topical reorganization of the motor cortex [44]. However, currently no overarching function or purpose has been associated with these connections in this model.

M1 to CB: Currently, the cerebellum is modeled as recieving input only from the PM and SMA and the brain stem. However, M1 also projects to the cerebellum. One possible use of this information is to help develop low level internal models. Instead of estimating the incurred descending motor commands issued from M1, or recieving the information relayed through the brain stem, M1 might directly project them to the cerebellum as well. Currently, however, not enough supporting evidence has been found to include this connection.

Parietal association cortex to PM and SMA: The PM and SMA do not actually directly recieve information from S1. Instead, these areas communicate with

a secondary sensory processing area, known as the parietal association cortex. For simplicity, in this model the two areas have been bundled together as one system component. It seems likely in the context of this framework that the parietal association cortex would be responsible for the transformation of the low level state representation into a high level state for use by the PM and SMA. As the sensory processing centers are outside of the scope of this work though, addressing this simplification is not of immediate concern.

4.5.2 Mathematical considerations

Compositionality of high and low level commands

Not discussed here is a technique for decomposing the optimal control signals generated by a linear Bellman controller [133]. This compositionality of optimal actions allows a command to be reformulated as a weighted summation of previously learned component commands operating in the same environment and passive dynamics. This decomposition will be added as a function of the cerebellum in the future, and incorporated into the low level information sent to M1. For more information on compositionality of optimal actions, see [133].

Currently, M1 is responsible for performing and analog function on the low level commands recieved from the cerebellum, implementing them as efficiently as possibly through synergies based on motor primitives and basic movements. However, the exact methods through which this function can be best implemented for low level control signals has not yet been analytically examined. For a complete model of the framework presented here to be implemented this issue must be addressed.

Incorporating mathematics for CPG control

One of the functions proposed to be performed by the cerebellum is maintaining a specified body state or movement pattern subconsciously while focus is given to a separate complex motor action. To implement this functionality in a biologically plausible fashion it will be necessary to examine the control and modulation of central pattern generators (CPGs) and reflex circuits in the spine. This topic of control through CPGs is largely addressed in the field of robotics, and much work has been recently done under a biological inspiration [77, 78, 141, 46]. The plausibility of current control techniques being implemented in biological systems will need to be

examined when reviewing their methods.

Iteration of high level control

Iterating the high level control process, where more detailed implementation instructions are specified at each iteration, will be a important area of research when examining the function of the PM and SMA in this framework.

In situations where a high level model is being used to guide movement through an area, it would be interesting to examine the different strategies used to navigate around obstacles. It seems likely that a sub-optimal state cost modification technique would be used in situations where the obstacles are irregularly encountered. When optimal action is required however, the slower but more effective reloading of the passive dynamics matrix would be expected. Taking this information and computing limb specific motor actions might involve several iterations before a control signal that can be sent to the cerebellum or M1 for implementation has been created. Examining the most efficient methods for high level control of multi-goal sequences will be a very important area of research, pertaining directly to the function of the PM and MSA, for this framework in the future.

It is the goal of the research started here to develop a neural circuit level implementation of the motor control system in the brain that can accurately replicate the functions and activation patterns of the real neural circuitry. This thesis has provided a foundation by laying out a control theoretic, biologically plausible framework that should support such future research.

Bibliography

[1] O. Adam and J. Jankovik. Symptomatic treatment of huntington disease. *Neurotherapeutics*, 5:181–197, 2008.

[2] R. Albin. The functional anatomy of basal ganglia disorders. *Trends in Neurosci*, 12:366–375, 1989.

[3] G. Alexander. Basal ganglia-thalamocortical circuits: parallel substrates for motor, oculomotor, 'prefront' and 'limbic' functions. *Prog Brain Res*, 85, 1990.

[4] G. Allen. Converence of cerebral inputs onto dentate neurons in monkey. *Exp. Brain Res.*, 23:459–469, 1997.

[5] R. Andersen. Cognitive neural prosthetics. *Trends in Cognitive Sciences*, 8:486–496, 2004.

[6] P. Arena, L. Fortuna, M. Frasco, and G. Sicurella. An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion. *IEEE Trans Sys Man Cybern*, 34:1823–37, 2004.

[7] C. Asanuma. Anatomical evidence for segregated focal groupings of efferent cells and their terminal ramifications in the cerebellothalamic pathway of the monkey. *Brain Res*, 5(3):266–298, 1983.

[8] C. Asanuma. Distribution of cerebellar terminations and their relation to other afferent terminations in the ventral lateral thalamic region of the monkey. *Brain Res Rev*, 5(3):237–266, 1983.

[9] M. Athans. The role and use of the stochastic linear-quadratic-gaussian problem in control system design. *Transaction on automatic control*, 16:529–552, 1971.

[10] T. Aumann. Cerebello-thalamic synapses and motor adaptation. In *The Cerebellum*, pages 68–77. Springer, 1 edition, 2002.

[11] M. Barinaga. Remapping the motor cortex. *Science*, 268:1696–9, 1995.

[12] M. Barinaga. Remapping the motor cortex. *Science*, 268:1696–9, 1995.

[13] A. Bastian. Learning to predict the future: the cerebellum adapts feedforward movement control. *Current Opinions in Neurobiology*, 16:645–649, 2006.

[14] H. Bergman. Physiological aspects of information processing in the basal ganglia of normal and parkinsonian primates. *Trends in Neuroscience*, 21:32–38, 1998.

[15] I. Bernabucci. A biologically inspired neural network controller for ballistic arm movements. *J Neuroeng Rehabit*, 3:33, 2007.

[16] E. Boccardi. Utilisation behaviour consequent to bilateral sma softening. *Cortex*, 38:289–308, 2002.

[17] E. Botterell and J. Fulton. Functional localization in the cerebellum of primates ii. lesions of midline structures (vermis) and deep nuclei. *J Comp Neurosci*, 69:47–62, 1938.

[18] E. Botterell and J. Fulton. Functional localization in the cerebellum of primates iii lesions of the hemispheres (neocerebellum). *J Comp Neurosci*, 69:63–87, 1938.

[19] D. Branco. Fundamentos e atualidades sobre a interpentrancia de areas omtoras corticais. *Braz J Neurol Psychiatry*, 1(0):29–33, 1996.

[20] D. Branco. Functional variability of the human cortical motor map: Electrical stimulation findings in perirolandic epilepsy surgery. *Journal of Clinical Neurophysiology*, 20(1):17–25, 2003.

[21] D. Brooks. The role of the basal ganglia in motor control: contributions from pet. *J Neurol Sci*, 128:1–13, 1995.

[22] M. Chesselet and J. Delfs. Basal ganglia and movement disorders: an update. *Trends in Neurosci*, 19:417–422, 1996.

[23] S. Chou. Effects of object size on intralimb and interlimb performance during bimanual task in stroke patients. *Journal of Biomechanics*, 2006.

[24] P. Cisek and J. Kalaska. Neural correlates of mental rehearsal in dorsal premotor cortex. *Nature*, 431:993–996, 2004.

[25] C. Davis. Neural prosthetics. *The Lancet*, 370:558, 2007.

[26] M. R. Delong. Chapter 43 - the basal ganglia. In *Principles of Neural Science*, pages 854–868. McGraw-Hill, 4 edition, 2000.

[27] E. Disbrow. Thalamocortical connections of the parietal ventral area (pv) and the second somatosensory area (s2) in macque monkeys. *Thalamus and Related Systems*, 1:289–302, 2002.

[28] K. Doya. What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Networks*, 12:961–974, 1999.

[29] K. Doya. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology*, 10, 2000.

[30] J. Doyle, B. Francis, and A. Tannenbaum. *Feedback Control Theory*. Macmillan Publishing Co, 1990.

[31] T. Elbert. Increased cortical representation of the fingers of the left hand in string players. *Science*, 270:305–7, 1995.

[32] W. Erihagen and E. Bicho. The dynamic neural field approach to cognitive robotics. *J Neural Eng*, 3:R36–54, 2006.

[33] T. Freeman, P. Sandberg, and O. Isacson. Development of the human striatum: Implications for fetal striatal transplantation in the treatment of huntington's disease. *Cell Transplantation*, 4:539–545, 1995.

[34] Fritsch and Hitzig. On the electrical excitability of the cerebrum. *Arch f. Anat*, pages 300–332, 1870.

[35] J. Fulton. Forced grasping and gropin in relation to the syndrome of the premotor area. *Arch Neurol Psychiat*, 1934.

[36] G. Gallese. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.

[37] V. Gallese. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 1996.

[38] C. Galletti, P. Battaglini, and P. Fattori. Functional properties of neurons in the anterior bank of the parieto-occipital sulcus of the macaque monkey. *Eur J Neurosci*, 7:2486–2501, 1996.

[39] G. Garraux. Talk-related interaction between basal ganglia and cortical dopamine release. *J Neurosci*, 27:14434–14441, 2007.

[40] P. George. Fabrication and biocompatibility of polypyrrole implants suitable for neural prosthetics. *Biomaterials*, 26:3511–3519, 2005.

[41] A. Georgopoulos. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J Neurosci*, 2:1527–1537, 1982.

[42] A. Georgopoulos. *The representation of movement diretion in the motor cortex: Single cell and population studies*. John Wiley and Sons, 1 edition, 1984.

[43] A. Georgopoulos, A. Schwartz, and R. Kettner. Neuronal population coding of movement direction. *Science*, 233:1416–1419, 1986.

[44] C. Ghez and J. Karkauer. Chapter 33. In *Principles of Neural Science*, pages 654–674. McGraw-Hill, 4 edition, 2000.

[45] C. Ghez and W. Thach. Chapter 42 - the cerebellum. In *Principles of Neural Science*, pages 833–853. McGraw-Hill, 4 edition, 2000.

[46] F. Gomez-Rodriguez. Aer auditory filtering and cpg for robot control. In *Proceedings of IEEE Symposium on Circuits and Systems*, pages 1201–1204, 2007.

[47] H. Gomi. Temporal firing patterns of purkinje cells in the cerebellar ventral paraflocculus during ocular following responses in monkeys. *J Neurophysiol*, 80:818–838, 1998.

[48] A. Gonshor and G. Melvill Jones. Short-term adaptive changes in the human vestibulo-ocular reflex arc. *J Physiol*, 256:361–379, 1976.

[49] A. Graybiel. The basal ganglia and cognitive pattern generators. *Schizophrenia Bulletin*, 5:1688–1703, 1997.

[50] A. Graybiel. Neurotransmitters and neuromodulators in the basal ganglia. *Trends in Neuroscience*, 13:244–254, 2003.

[51] A. Greybiel. Building action repetoires: memory and learning functions of the basal ganglia. *Current Opinion in Neurobiol*, 5:733–741, 1995.

[52] J. Grezes. Activations related to 'mirror' and 'canonical' neurounes in the human brain: an fmri study. *Neuroimage*, 2003.

[53] J. Grezes and J. Decety. Does visual perception of object afford action? evidence from a neuroimaging study. *Neuropsychologia*, 40:212–222, 2003.

[54] R. Hari. Activation of human primary motor cortex during action observation: A neuromagnetic study. *PNAS*, 95:15061–15065, 1998.

[55] M. Haruno, D. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Computations*, 13:2201–2220, 2001.

[56] R. Hassler. Anatomy of the thalamus. In S. Thieme, editor, *Introduction to Stereotaxis with an Atlas of the Human Brain*, pages 230–290. P. Bailey, 1 edition, 1959.

[57] K. Hepp. European brain research. *Nature*, 1994.

[58] J. Hoover and P. Strick. Multiple output channels in the basal ganglia. *Science*, 259:819–821, 1993.

[59] E. Hoshi. Neurons in the rostral cingulate motor area monitor multiple phases of visuomotor behavior with modest parametric selectivity. *J Neurophysiol*, 2005.

[60] M. Isoda and O. Hikosaka. Switching from automatic to controlled action by monkey medial frontal cortex. *Nature Neurosci*, 9:240–248, 2007.

[61] E. Jones. *The Thalamus*. Plenum Press, 1985.

[62] M. Kawato. The cerebellum and vor/okr learning models. *Trends Neurosci*, 15:445–453, 1992.

[63] M. Kawato. Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology*, 9:718–727, 1999.

[64] M. Kawato, K. Furukawa, and R. Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57:169–185, 1987.

[65] M. Kawato, K. Furukawa, and R. Suzuki. A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics*, 57:169–185, 2004.

[66] T. Kitama. Motor dynamics encoding in cat cerebellar flocuculus middle zone during optokinetic eye movements. *J Neurophysiol*, 82:2235–2248, 1999.

[67] Y. Kobayashi. Temporal firing patterns of prukinje cells in the cerebellar ventral paraflocculus during ocular following responses in monkeys. *J Neurophysiol*, 80:832–848, 1998.

[68] A. Krainik. Role of the supplementary motor area in motor deficit following medial frontal lobe surgery. *Neurology*, 57:871–878, 2001.

[69] A. Krainik. Role of the supplementary motor area in motor deficit following medial frontal lobe surgery. *Neurology*, 57:871–878, 2001.

[70] I. Leroi, D. Collins, and L. Marsh. Non-dopaminergic treatment of cognitive impairment and dementia in parkinson's disease: A review. *Journal of the Neurological Sciences*, 248:104–114, 2006.

[71] W. Li and E. Todorov. An iterative optimal control and estimation design for nonlinear stochastic system. In *Proceedings of IEEE Conference on Decision and Control*, volume 45, pages 1439–1453, 2006.

[72] D. Liu and E. Todorov. Hierarchical optimal control of a 7-dof arm model. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 50–57, 2009.

[73] R. Llinas and K. Walton. Cerebellum. *The synaptic organization of the brain*, 4:255–288, 1998.

[74] G. Loeb, I. Brown, and E. Cheng. A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research*, 126:1–18, 1999.

[75] E. Lukhanina. Functional organization of the ventrolateral nucleus of the thalamus. *Neurophysiology*, 26(6):378–390, 1994.

[76] A. Lundberg and F. Weight. Functional organization of connections to the ventral spinocerebellar tract. *Exp Brain Res*, 12:295–316, 1971.

[77] T. Matsuo. Biomimetric motion control system based on a cpg for an amphibious multi-link mobile robot. *Journal of Bionic Engineering*, 5:91–97, 2008.

[78] T. Matsuo and K. Ishii. A cpg control system for a modular type mobile robot. In *International Congress Series*, pages 206–209. 1301 edition, 2007.

[79] Y. Matsuzaka and J. Tanji. Changing directions of forthcoming arm movements: neuronal activity in the presupplementary and supplementary motor area of monkey cerebral cortex. *J Neurophysiol*, 76:2327–2342, 1996.

[80] J. Meyer-Lohman. Cerebellar participation in generation of prompt arm movements. *J Neurophysiol*, 40:1038–1050, 1977.

[81] S. Miyachi. Differential roles of monkey striatum in learning of sequential hand movement. *Experimental Brain Research*, 5:1–5, 1997.

[82] O. Monchi. Functional role of the basal ganglia in the planning and execution of actions. *Ann Neurol*, 59:257–264, 2006.

[83] F. Mussa-Ivaldi. Nonlinear force fields: a distributed system of control primitives for representing and learning movements. In *Proc IEEE Int. Symp. of Computational Intelligence in Robotics and Automation*, pages 84–90, 1997.

[84] F. Mussa-Ivaldi, S. Giszter, and E. Bizzi. Motor-space coding in the central nervous system. In *Cold Spring Harbor symposia on quantitative biology*, volume 55, pages 827–835. Cold Spring Harbor Laboratory Press, 1990.

[85] F. Mussa-Ivaldi, S. Giszter, and E. Bizzi. Linear combinations of primitives in vertebrate motor control. In *Proc. Natl Acad. Sci. USA*, volume 91, pages 7534–7538, 1994.

[86] K. Nakamura. Effects of local inactivation of monkey medial frontal cortex in learning sequenctial procedures. *J Neurophysiol*, 82:1063–1068, 1999.

[87] K. Nakano. Cortical connections of the motor thalamic nuclei in the japanese monkey. *Stereotact Funct Neurosurg*, 60:42–61, 1993.

[88] P. Neilson and M. Neilson. An overview of adaptive model theory: solving the problems of redundancy, resources, and nonlinear interactions in human movement control. *J Neural Eng*, 2:S279–312, 2005.

[89] R. Nieuwenhuys. *The Human Central Nervous System: A Synopsis and Atlas*, volume 2. Berlin, 1981.

[90] U. of Florida. *UFLA Review of neuroanatomy.*

[91] E. Olivier. Precision grasping in humans: from motor control to cognition. *Current Opinions in Neurobiology*, 17:644–648, 2007.

[92] R. Passingham and J. Aggeleton. Syndrome produced by lesions of the amygdala in monkeys (mucaca mulatta). *J. Comp. Physiol Psychol*, 1981.

[93] K. Pearson and J. Gordon. Chapter 36 - spinal reflexes. In *Principles of Neural Science*, pages 714–737. McGraw-Hill, 4 edition, 2000.

[94] D. Pellegrino. Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91:176–180, 1992.

[95] W. Penfield and E. Boldrey. Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. *Brain*, 60:389–443, 1937.

[96] W. Penfield and T. Rasmussen. *The cerebral cortex of man. A clinical study of localization of function.* The Macmillian Company, 1952.

[97] M. Petrides. Impairments on nonspatial and externally ordered working memory tasks after lesions of the mid-dorsal part of the lateral frontal cortex in the monkey. *Journal of Neuroscience*, 1985.

[98] M. Petrides. Local morphology predicts functional organization of the dorsal premotor region in the human brain. *Journal of Neuroscience*, 1997.

[99] M. Raichle. Practice-related changes in human brain functional anatomy during non-motor learning. *Cerebral Cortex*, 4:8–26, 1994.

[100] G. Rebecca. Mental practice and acquisition of motor skills: Examples from sports training and surgical education. *Obstetrics and Gynecology Clinics of North America*, 33:297–304, 2006.

[101] A. Riehle and E. Vaadia. *Motor cortex in voluntary movements.* CRC Press, 2000.

[102] U. Rinne. Treatment of early parkinson's disease. *Parkinsonian and Related Disorders*, 7:59–62, 2000.

[103] G. Rizzolatti. Functional organization of inferior area 6 in the macaque monkey. *Experimental Brain Research*, 1988.

[104] T. Roppongi. Perospirone in treatment of huntington's disease: A first case report. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 31:308–310, 2006.

[105] G. Russo. Neural activity in monkey dorsal and ventral cingulate motor areas: comparison with the supplementary motor area. *J Neurophysiol*, 2002.

[106] C. Saffer. Chapter 44 - brain stem, reflexive behaviour, and cranial nerves. In *Principles of Neural Science*, pages 874–889. McGraw-Hill, 4 edition, 2000.

[107] J. Sanes. Rapid reorganization of adult rat motor cortex somatic representation patterned after motor nerve injury. In *Proc Nat Acad Sci USA*, volume 85, pages 2003–2007, 1988.

[108] J. Sanes and J. Donoghue. Plasticity and primary motor cortex. *Annual Review of Neuroscience*, 23:393–415, 2000.

[109] G. Schell. The origin of thalamic inputs to the arcuate and supplementary motor areas. *J Neurosci*, 4:539–560, 1984.

[110] H. Scherberger. Neural prosthetics for reaching. *Encyclopedia of Neuroscience*, pages 213–220, 2009.

[111] M. Schieber and L. Hibbard. How somatotopic is the motor cortex hand area? *Science*, 261:489–93, 1993.

[112] M. Schieber and L. Hibbard. How somatotopic is the motor cortex hand area? *Science*, 261:489–93, 1993.

[113] O. Schiller and I. Chou. The effects of frontal eye field and dorsomedial frontal cortex lesions on visually guided eye movements. *J Neurol Neurosurg Psychiatry*, 72:77–85, 1998.

[114] W. Schultz. Reward prediction in primate basal ganglia and frontal cortex. *Neuropharmacology*, 37:421–429, 1998.

[115] A. Schwartz. Brain-controlled interfaces: Movement restoration with neural prosthetics. *Neuron*, 52:205–220, 2006.

[116] L. Shi, F. Luo, D. Woodward, and J. Chang. Neural responses in multiple basal ganglia regions during spontaneous and treadmill locomotion tasks in rats. *Experimental Brain Research*, 157:303–314, 2003.

[117] H. Shibasaki. Both primary motor cortex and supplementary motor area play an important role in complex finger movement. *Brain*, 116:1387–1398, 1993.

[118] M. Shidara. Inverse-dynamics model eyemovement control by purkinje cells in the cerebellum. *Nature*, 365:50–52, 1993.

[119] K. Shima and J. Tanji. Role for cingulate motor area cells in voluntary movement selection based on reward. *Science*, 1998.

[120] H. Shimazu. Macaque ventral premotor cortex exerts powerful facilitation of motor cortex outputs to upper limb motoneurons. *The Journal of Neuroscience*, 2004.

[121] Y. Smith, M. Bevan, E. Shink, and J. Bolam. Microcircuitry of the direct and indirect pathways of the basal ganglia. *Neuroscience*, 86:353–387, 1998.

[122] M. Sommer. The role of the thalamus in motor control. *Curr Opin Neurobiol*, 13:663–670, 2003.

[123] M. Strotzer. *Once century of brain mapping using Brodmann areas*, volume 1. Urban and Vogel, 2009.

[124] J. Tani, R. Nishimoto, and R. Paine. Achieving.

[125] J. Tanji and K. Shima. Role for supplementary motor area cells in planning several movements ahead. *Nature*, 371:413–4–16, 1994.

[126] J. Tanner. Perusing psychology. *http://www.brainybehavior.com*, 2009.

[127] W. Thach. A role for the cerebellum in learning movement coordination. *Neurobiol Learn Mem*, 70:177–188, 1998.

[128] D. Thaler. The functions of the medial premotor cortex. i. simple learned movements. *Exp. Brain Research*, 102:445–460, 1995.

[129] E. Todorov. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nature Neuroscience*, 3(4):391–397, 2000.

[130] E. Todorov. On the role of primary motor cortex in arm movement control. *Progress in Motor Controll*, 3:125–166, 2002.

[131] E. Todorov. From task parameters to motor synergies: A hierarchical framework for approximately optimal control of redundant manipulators. *Journal of Robotic Systems*, 22:691–710, 2005.

[132] E. Todorov. Eigenfunction approximation methods for linearly-solvable optimal control problems. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 161–168, 2008.

[133] E. Todorov. Compositionality of optimal control laws. In *NIPS*, 2009.

[134] E. Todorov. Efficient computation of optimal actions. *PNAS*, 106:11478–11483, 2009.

[135] E. Todorov. Parallels between sensory and motor information processing. In *The Cognitive Neurosciences*. MIT Press, 4 edition, in press.

[136] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of American Control Conference*, pages 300–306, 2005.

[137] A. Travis. Neurolocial deficiencies following supplementary motor area lesions in macaca mulatta. *Brain*, 78:174–198, 1955.

[138] B. Tripp. A search for principles of basal ganglia function. *PhD Thesis*, University of Waterloo, 2009.

[139] E. Tunik, S. Frey, and S. Grafton. Virtual lesions of the anterior intraparietal area disrupt goal-dependent on-line adjustments of grasp. *Nature Neuroscience*, 2005.

[140] J. Vegte. *Feedback Control Systems*. Prentic Hall, 1994.

[141] W. Wang. The cpg control algorithm for a climbing worm robot. In *Proceedings of IEEE Industrial Electronics and Applications*, volume 3, 2008.

[142] M. Weinrich and S. Wise. The premotor cortex of the monkey. *Journal of Neuroscience*, 1982.

[143] R. Wichman and M. DeLong. Functional and pathological models of the basal ganglia. *Curr Opin Neurobiol*, 6:751–758, 1996.

[144] A. Wing. Motor control: Mechanisms of motor equivalence in handwriting. *Current Biology*, 10:245–248, 2000.

[145] S. Wise. The primate premotor cortex: past, present, and preparatory. *Annual Review of Neuroscience*, 1985.

[146] D. Wolpert. Internal models in the cerebellum. *Trends in Cognitive Sciences*, 2:338–347, 1998.

[147] C. Yeo. Discrete lesions of the cerebellar cortex abolish the classically conditioned nictitating membrane response of the rabbit. *Behav Brain Res*, 13:261–266, 1984.