



TÉCNICO
LISBOA

Cerebellum Model for Movement Coordination

Leonardo Filipe Farinha Marques

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Andreas Miroslaus Wichert

Examination Committee

Chairperson: Prof. João António Madeiras Pereira

Supervisor: Prof. Andreas Miroslaus Wichert

Member of the Committee: Dr. Francisco Jorge Dias Oliveira Fernandes

July 2020

Abstract

A computational model for motor coordination is proposed based on theories on cerebellar function. The role of the cerebellum in coordinating motion has been well established but recent findings suggest additional roles in sensory prediction and cognition. We start by reviewing the works of Marr and Albus, some of the most influential theories about the cerebellum. We also look at a less known theory by Mechsner, which proposes that the cerebellar cortex performs a coupling operation, grouping elementary movements executed frequently together. With this theory as a starting point, we propose a computational model extended with the findings on sensory prediction and concepts from internal model theories. A model of this nature could later be applied to the field of robotics, permitting the execution of smooth and natural motion. We implemented a simple model capable of guiding an agent in a two-dimensional environment from a starting position to a goal using simple elementary movements. The evolution of said model is described in detail in this dissertation. One of its features is a strategy that is capable of making simple combinations of elementary movements, creating new movements. The model also has a method to refine the learned associations in order to make small adjustments to the agent's trajectory. Other strategies were also implemented and we did experiments to compare the performance of the different possible configurations. The final version of the model, however, is still far from our proposed solution.

Keywords — Cerebellum, Motor Control, Sensorimotor Control, Internal Models, Computational Neuroscience

Resumo

É proposto um modelo computacional para coordenação motora baseado em teorias sobre a função do cerebelo. O papel do cerebelo na coordenação de movimentos está bem estabelecido mas descobertas recentes sugerem papéis adicionais na predição sensorial e cognição. Começamos por rever as obras de Marr e Albus, umas das mais influentes teorias sobre o cerebelo. Também analisamos uma teoria menos conhecida da autoria de Mechsner, que propõe que o córtice cerebelar executa uma operação de emparelhamento, agrupando movimentos elementares executados frequentemente em conjunto. Com esta teoria como ponto de partida, propomos um modelo computacional expandido com as descobertas sobre previsão sensorial e conceitos de teorias sobre modelos internos. Um modelo desta natureza poderia ser mais tarde aplicado no campo da robótica, permitindo a execução de movimentos suaves e naturais. Implementámos um modelo simples capaz de guiar um agente, num ambiente de duas dimensões, de uma posição inicial a um objetivo usando apenas simples movimentos elementares. A evolução deste modelo é descrita em detalhe nesta dissertação. Uma das suas características é uma estratégia capaz de fazer combinações simples de movimentos elementares, criando novos movimentos. O modelo também possui um método para aprimorar as associações aprendidas de modo a fazer pequenos ajustes à trajetória do agente. Também foram implementadas outras estratégias e fizemos experiências para comparar o desempenho das diferentes configurações possíveis. A versão final do modelo, porém, ainda se encontra longe da nossa solução proposta.

Palavras-Chave — Cerebelo, Controlo Motor, Controlo Sensorimotor, Modelos Internos, Neurociência Computacional

Contents

1	Introduction	7
2	Background and Related Work	9
2.1	The Cerebellar Architecture	9
2.2	David Marr (1969)	12
2.3	James Albus (1971)	13
2.4	Franz Mechsner (1996)	15
2.5	Sensory Prediction	17
2.5.1	Sensory Attenuation	18
2.5.2	Error Detection and Correction	19
2.5.3	Prediction Driven Execution of Movement	20
2.6	Internal Models	20
2.7	Cognitive function	22
3	Proposed Solutions	23
3.1	Solution Architecture	25
3.2	Applications	26
4	Methodology	28
5	The Evolution of the Model	30
5.1	First Version of the Model	31
5.1.1	The Agent's Movement	31
5.1.2	Goals	31
5.1.3	The Model Cerebrum	32
5.1.4	Reacting to Recent Movement Decisions	32
5.1.5	The Model Cerebellum	33
5.1.6	How the Model Behaves	33
5.1.7	Problems With This Version	34
5.2	Second Version of the Model	35
5.2.1	The Grid	36
5.2.2	The Model Cerebrum	37

5.2.3	Reaching the Goal	38
5.2.4	The Model Cerebellum	38
5.2.5	Problems With This Version	39
5.3	Third Version of the Model	40
5.3.1	The Relative Grid	40
5.3.2	The Agent's Movement	42
5.3.3	Goals and How to Meet Them	42
5.3.4	The Model Cerebrum	43
5.3.5	The Model Cerebellum	43
5.3.6	How the Model Behaves	44
5.3.7	Problems With This Version	44
5.4	Fourth Version of the Model	45
5.4.1	Goals and How to Meet Them	45
5.4.2	The Model Cerebrum	45
5.4.3	The Model Cerebellum	46
5.4.4	How the Model Behaves	46
5.4.5	Problems With This Version	47
5.5	Fifth Version of the Model	49
5.5.1	The Model Cerebellum	49
5.5.2	How the Model Behaves	50
5.5.3	Problems With This Version	51
5.6	Sixth Version of the Model	52
5.6.1	Goals and How to Meet Them	52
5.6.2	The Agent's Movement	52
5.6.3	The Relative Grid	52
5.6.4	Resolution Update Strategies	53
5.6.5	The Model Cerebrum	56
5.6.6	The Model Cerebellum	56
5.6.7	How the Model Behaves	56
5.6.8	Problems With This Version	57
5.7	Seventh Version of the Model	57
5.7.1	The Model Cerebellum	57
5.7.2	Problems With This Version	57
5.8	Eighth Version of the Model	58
5.8.1	The Agent's Movement	58
5.8.2	The Graphical Representation	58
5.8.3	Generating the Start and Goal Positions	59
5.8.4	Problems With This Version	59
5.9	Ninth Version of the Model	61

5.9.1	The Model Cerebellum	61
5.9.2	The Point Threshold Strategy	61
5.9.3	Problems With This Version	61
6	Experiments	63
6.1	Seventh Version of the Model	63
6.1.1	Setup	63
6.1.2	Results	63
6.2	Eighth Version of the Model	64
6.2.1	Training Runs	65
6.2.2	Field Dimensions	67
6.2.3	Confidence Threshold	69
6.2.4	Summary of the Results	71
6.3	Ninth Version of the Model	72
6.3.1	Setup	72
6.3.2	Results	72
7	Conclusion	74

List of Figures

2.1	The cerebellar architecture.	10
2.2	The layers of the cerebellar cortex separated by a dashed line. The figure portrays the compact and efficient organization of Purkinje cells and parallel fibers.	11
2.3	Application of a forward model in a system.	21
3.1	Architecture of the proposed model.	25
4.1	Trace produced by a cerebellar patient (below) while trying to reproduce the trace above. From a book on cerebellar function by André-Thomas [1].	29
5.1	An example of the grid dividing the environment.	36
5.2	An example of the same movement being represented in grid with different cell density.	37
5.3	This grid displays two movements that interfere with each other.	39
5.4	These grids display all places where the same movement can be performed.	40
5.5	The execution of a movement represented on a relative grid.	41
5.6	Comparison of movement interference in the absolute grid and in the relative grid.	42
5.7	How different versions of the same movement are represented on a relative grid.	43
5.8	An example of multiple choices when selecting a movement.	45
5.9	An example of two possible trajectories that take the least number of steps to reach the proposed goal.	46
5.10	An example of a model cerebellum lost in the environment.	47
5.11	An example of a movement pattern and the sub-patterns it includes represented on the relative grid.	48
5.12	Combining elementary movements to produce new movements.	48
5.13	A screenshot of the simple graphical representation used to visualize the trajectory taken by the agent when controlled by the model cerebellum.	50
5.14	A movement pattern that resembles the letter Z, represented on the absolute grid.	50
5.15	An example of how using the grid may produce a movement pattern that does not properly take the agent to the goal.	51
5.16	How the relative field and grid are generated.	53
5.17	Different methods of dividing the relative grid to produce a new grid of higher resolution.	54
5.18	Comparison of different resolutions using the same goal circle radius.	54

5.19	The zoom resolution update strategy.	55
5.20	The tile resolution update strategy.	55
5.21	An image produced by the model displaying a movement performed by the model cerebellum with aid from the model cerebrum.	59
5.22	An illustration of the looping problem.	60
6.1	Variation of the success rate with the number of training runs.	66
6.2	Variation of the indecision rate with the number of training runs.	66
6.3	Variation of the distance deviation percentage with the number of training runs.	67
6.4	Variation of the success rate with the size of the environment.	68
6.5	Variation of the distance deviation percentage with the size of the environment.	68
6.6	Variation of the indecision rate with the size of the environment.	69
6.7	Variation of the success rate with the confidence threshold.	70
6.8	Variation of the indecision rate with the confidence threshold.	70
6.9	Variation of the distance deviation percentage with the confidence threshold.	71

List of Tables

6.1	Comparison of the success rate of the different combinations of resolution update strategies and choice strategies.	63
6.2	Comparison of the point threshold strategy with the other strategy combinations using 1500 training runs.	72
6.3	Comparison of the point threshold strategy with the other strategy combinations using 3000 training runs.	73

Chapter 1

Introduction

The cerebellum has been the object of study for researchers for many years. One of its most fascinating features is that, although it only constitutes a small portion of the human brain's volume, it contains the majority of its neurons [2]. From many studies, its role on motor coordination has been well established [3–6]. Many theoretical models have been proposed to explain how this brain structure contributes to coordinate movements. Some of the most influential were the works of Albus and Marr [7, 8]. However, these will not be the only models reviewed in this dissertation.

Studies that are more recent make clear that the cerebellum does not only participate in motor coordination. The work of authors such as Blakemore et al. [9], Nixon et al. [10] and Bell et al. [11, 12] bring forth evidence that this brain structure has the capacity to predict the sensory responses resulting from performing an action. As we will present in this dissertation, this ability complements its operation in motor control as it allows it to perform corrections to a movement by comparing its prediction with the actual resulting sensory input. Besides that, by making these types of predictions it is possible to attenuate self-generated sensations to better interpret external input. As will be described later in section 2.5 this property is essential to certain animal species such as the mormyrid fish that uses electrical impulses to communicate and locate food.

Additionally, the ability to predict the sensory consequences of an action supports many proposed theories that the cerebellum contains internal models of the motor system. Aside from its motor learning capabilities, the cerebellar cortex is also implicated in classical conditioning. We will also focus on some of these theories and their implications in this work.

The goal of this project is to create a computational model inspired by cerebellar function. Such a model would prove to be useful in the engineering field as it could allow robotic limbs to produce smooth and natural motion.

We developed a simple model that is designed to guide an agent from a start position to a goal in a two-dimensional environment using simple movements. As the model evolved, we implemented multiple strategies to be used in deciding which movement the agent should perform. Strategies were developed to determine how associations are learned and updated.

The resulting model is capable of guiding an agent to a goal position using simple movement patterns

learned by the model cerebellum. We made some experiments to test how the model performed in the proposed task and how the different strategies compare to each other. In this dissertation, we describe the implementation of this model and our experiments in detail.

The dissertation is organized as follows. In the next chapter, we will provide a brief explanation of the structure of the cerebellum with the aim to give readers less versed in neuroscience enough information to better understand the concepts introduced in the theoretical models. The following sections in that chapter will provide a more in depth view of the proposals of Marr [8], Albus [7] and Mechsner [13]. The chapter ends with a section where we review some studies that reveal the role of the cerebellum in predicting the consequences of actions. These will be followed by some proposals made pertaining to internal models and cognition in the cerebellum.

In chapter 3, we will present our proposed solutions for the model. The following chapter describes the methodology we used to implement our model. We then get to the chapter where we describe the implementation and evolution of our model, followed by a chapter detailing the results of our experiments on it. Finally, we conclude the dissertation providing a summary of the knowledge we acquired, the results we achieved with the model and what is yet to be done.

Chapter 2

Background and Related Work

2.1 The Cerebellar Architecture

Given the heavy dependency of this project on neuroscientific knowledge, we find that it would be highly beneficial to provide a succinct explanation of the cerebellar architecture. As the readers of this dissertation might come from a computer science background, it would help give a basic understanding of the concepts and vocabulary associated with the cerebellum, which will be used frequently throughout this dissertation.

Neurons are the main components of the nervous system. They are cells composed of three main structures: the soma, which is the body of the cell; the dendrites, thin ramifications that receive input from other neurons and propagate it to the soma; and the axon (or nerve fiber), that carry signals from the cell body to other neurons.

Connections between neurons are called synapses. Synapses deliver signals unidirectionally from one cell to another. Typically, the axon of one neuron will connect to the dendrites or soma of another, forming a synapse. Neurons are electrically excitable and thus, there are excitatory synapses and inhibitory ones. If the excitatory inputs overwhelm the inhibitory inputs in a short time frame and bring the cell past a voltage threshold, the cell fires and a signal is sent through its axon.

The cerebellar cortex, consisting of grey matter, is the outer layer of the cerebellum and the most relevant structure for this work. It is the site where the neuronal cells responsible for massive processing ability of the cerebellum reside. The organization of the cortex and its connections to other structures can be seen in figure 2.1.

Mossy fibers are one of the inputs into the cerebellar cortex and can branch to many of its folds. They carry information from many points of the nervous system such as the cerebrum, responsible for cognition and conscious processing; the spinal cord, the major pathway for communication between the nervous system and the body; and the vestibular system, responsible for balance [14]. Mossy fibers perform excitatory synapses with granule cells and Golgi cells in structures called cerebellar glomeruli. These fibers seem to bring context information to the cerebellum, consisting of proprioceptive and motor information [4, 15–17].

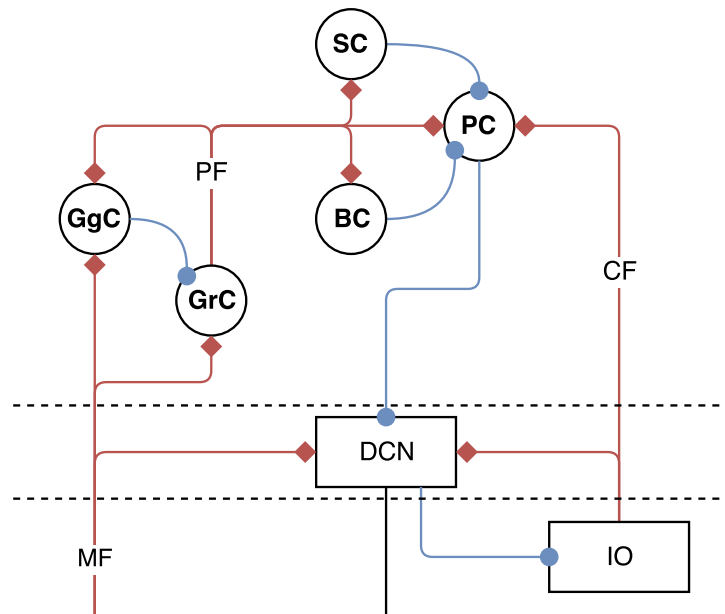


Figure 2.1: The cerebellar architecture. The first dashed line separates the cerebellar cortex (above). The second dashed line separates the cerebellum (above) from the inferior olive and the rest of the central nervous system (below). Red diamond-ended connections represent excitatory synapses while blue circle-ended ones depict inhibitory synapses. MF: mossy fiber; GrC: granule cell; GgC: Golgi cell; PF: parallel fiber; BC: basket cell; SC: stellate cell; PC: Purkinje cell; CF: climbing fiber; IO: inferior olive; DCN: deep cerebellar nuclei.

Granule cells are the most numerous neurons in the cerebellum and also the entire brain [2]. The granule cell axon moves up toward the outmost layer of the cerebellar cortex where it splits into two ends, which extend very far in opposite directions, performing excitatory synapses with the cells they go through. The fact that all granule cell axons run parallel to each other gives them the name parallel fibers. Each cell receives input from four to five mossy fibers on average, each in a different glomerulus. Their function has been proposed to be to preprocess input from mossy fibers before delivering it to Purkinje cells, making discrepancies between similar inputs more prominent [7, 8]. A recent study suggests that granule cells are also responsible for processing reward information, adding an additional type of context input to the cerebellum [18].

Golgi cells have an inhibitory effect on granule cells and possess two large dendritic arborizations. One of them extends upward, receiving excitatory inputs from parallel fibers, and the other goes the opposite direction, receiving input from mossy fibers. This allows Golgi cells to regulate granule cells based on feed-forward information from mossy fibers or feedback from parallel fibers. A recent study by Duguid et al. [19] has found that most of the time, in response to a mossy fiber input, a Golgi cell inhibits a granule cell before the same input reaches the latter.

According to the works of Eccles et al. [20], each cerebellar glomerulus contains one mossy fiber termination (called a rosette), dendrite branches from about 20 distinct granule cells and dendritic and axon terminal from one Golgi cell.

Purkinje cells are one of the most important types of neuron in the cerebellar cortex and are the origin of its only output, which inhibits cells located in the deep cerebellar nuclei. Their suggested function is to learn input patterns provided by parallel fibers and respond appropriately to them. Their large, flat

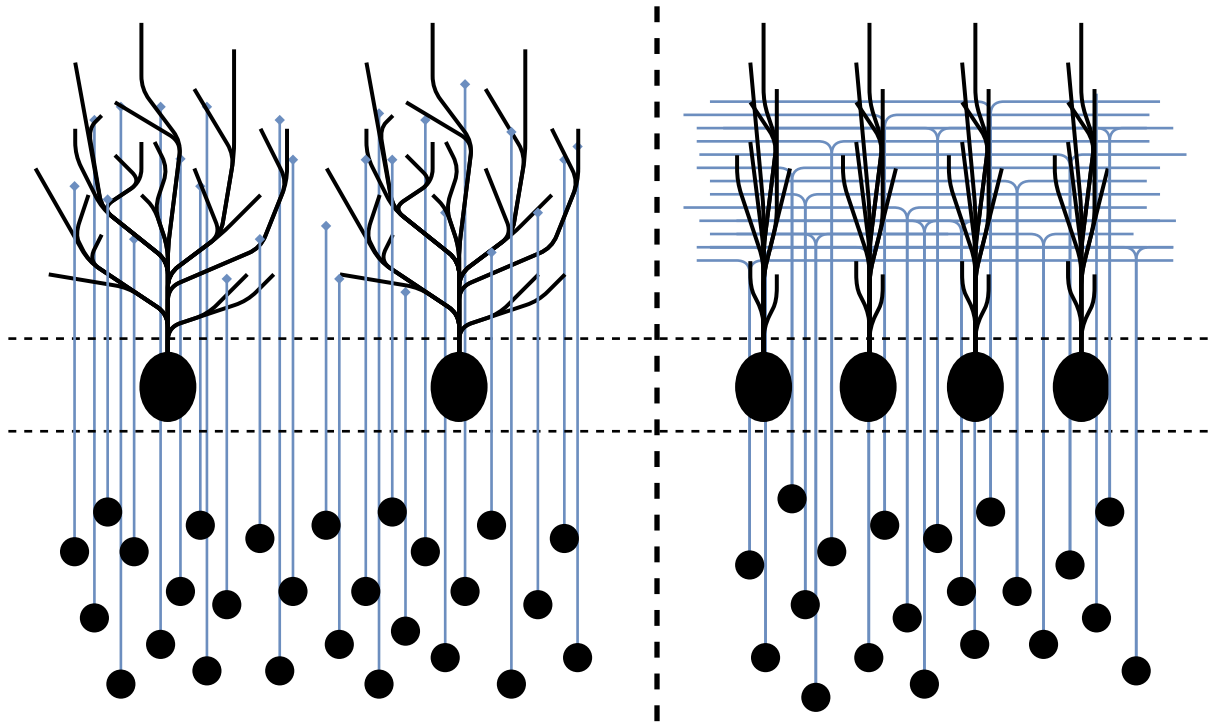


Figure 2.2: The layers of the cerebellar cortex separated by a dashed line. The figure portrays the compact and efficient organization of Purkinje cells and parallel fibers. Bottom: granular layer. Middle: Purkinje layer. Top: Molecular layer.

and dense dendritic tree allows them to receive excitatory inputs from an enormous amount of parallel fibers, which run perpendicular to them. Each cell may synapse with about 200000 parallel fibers. They also receive a strong excitatory input from a single climbing fiber, originating in the inferior olive outside the cerebellum. However, a climbing fiber can contact multiple Purkinje cells. They climb the cell, intertwining around its dendrites, hence their name. Basket and stellate cells have inhibitory effects on Purkinje cells but their exact role in cerebellar function is still not well defined.

The quantitative proportions of the different cells and synapses, some of which presented in this section, are of interest as they could be used as a guidance to implement a more detailed model or to guide its fine-tuning in order to improve its performance. An example of such a study is the work of Pinzon-Morales and Hirata [21], which examines the consequences of varying the number of granule cells in a simulation. The quantitative data presented comes from the work of Eccles et al. concerning the cat cerebellum.

The cerebellar cortex is divided in three layers, presented in figure 2.2: the granular layer, where the granule cells reside, the Purkinje layer populated by the somata (cell bodies) of the Purkinje cells, and the molecular layer, occupied by Purkinje dendrites, parallel fibers, basket cells in the lower portion and stellate cells in the upper portion.

The vast interest on the cerebellum comes from this highly regular organization of the few types of cells present. This regularity suggests that all areas of the cerebellum perform a similar type of computation, which could help identify its fundamental function and makes it attractive for the design of computational models.

2.2 David Marr (1969)

Much of the appeal of Marr's theory of the cerebellar cortex [8], published in 1969, comes from the fact that it provides an explanation to how its function can be achieved by its neuron organization. The theory describes how each cell and synapse could contribute to the operation and output of the cerebellum.

Marr proposes that a complex motion is composed of simpler movements, called elemental movements. Each Purkinje cell affects the execution of a single elemental movement.

According to this theory, the cerebellum learns to execute complex movements by associating patterns of context information with elemental movement patterns. The two types of patterns are proposed to be associated by the Purkinje cells, which provide the only output of the cerebellar cortex. Once an association is made, the simple occurrence of the context should trigger the appropriate movement pattern without the need of conscious intervention by the cerebrum. Once the Purkinje cells respond to a context, the resulting action would form the context for the next part of the movement, triggering the appropriate cells and continuing the movement automatically.

Context input is provided by mossy fibers and describes the current state of the body with proprioceptive information. This is the sensory information about the position of the different body parts or about resistance to their movement. The information about the movement pattern to be associated is provided by climbing fibers. Their excitatory influence on Purkinje cells is so strong that when they are active the cells fire immediately.

Another way to describe the association is that a set of active mossy fibers (the input pattern) is associated to a set of firing Purkinje cells (the output pattern). Each Purkinje learns to recognize a number of input patterns, firing accordingly. The set of cells firing to an input pattern constitutes the output pattern. It is important to note that, in this theory, synapses are considered to be binary, and they are turned either on or off. Therefore, input patterns can be represented by binary values where each bit corresponds to a mossy fiber input. The same goes for output patterns and Purkinje cells.

This system creates a problem. If a Purkinje cell learns too many patterns, they will start to overlap and the cell will begin to fire to incorrect input patterns. This limits the learning capacity of each cell. However, the mossy fibers do not contact Purkinje cells directly. Their input is first processed by the granule cells, which greatly outnumber mossy fibers, before delivering it to Purkinje cells via parallel fibers. This process transforms the initial input pattern into a larger set of inputs with increased redundancy and more easily recognizable. The existence of granule cells, therefore, increases the learning capacity of a Purkinje cell.

As stated, granule cells are suggested to have the purpose of discriminating similar input patterns, intensifying their discrepancies. These cells seem to have one to seven dendritic trees, called claws due to their shape, each of which receive input from distinct mossy fibers, 4 or 5 on average. However, the limit of seven claws per cell introduces a limit to their discriminatory power. It is suggested by Marr that when this is not enough the control reverts to the cerebrum.

The learning signal is provided by climbing fibers, exciting the Purkinje cells that should be firing on a given input pattern. The input carried by these fibers comes from the inferior olive and is suggested to

contain information resulting from conscious processing by the cerebrum, signaling the execution of an elemental movement. It is proposed that, at the time of climbing fiber excitation, synapses from parallel fibers active on the Purkinje cell are strengthened, reinforcing that the cell should respond to that input pattern. Since synapses are considered to be binary, when they are strengthened their weights change from zero to one.

Marr considered that synapses between parallel fibers and Purkinje cells were the only modifiable synapses on the cerebellar cortex. The threshold of granule cells to mossy fiber input was proposed to be regulated by Golgi cells, which sample both parallel fiber and mossy fiber activity. By keeping the number of granule cells that fire on an input pattern low, more patterns can be stored and easily recognized by Purkinje cells. According to this theory, the number of patterns that can be stored by a single Purkinje cell is about 200 different inputs.

Tyrrell and Willshaw made a simulation of Marr's theory using the quantitative proportions he presented for the cell organization [22]. The authors also provide a detailed summary of the theory with many helpful illustrations. They compare it to the concept of associative memory [23], which was pioneered by Steinbuch [24, 25].

The authors of the simulation also highlight certain inaccuracies with the theory and its problems. The main problem is that it is not possible to forget learned patterns. Since all synapses can only have one of two possible weight values, it is not possible to prioritize certain parts of a pattern. This makes it impossible to base the decision to recognize the pattern on the more important portions of an input, ignoring the portions that are not very relevant. Another problem caused by fixed binary synapses is that it is not possible to improve a learned movement by adjusting synapse weights. Even if they could be altered, the fact that weights are binary reduces the variability available.

An additional issue of binary synapses is that they are not biologically accurate and the way they are altered seems to be inaccurate as well. Marr proposes that synapses are strengthened by climbing fibers, a process called long-term potentiation. However, studies made on climbing fibers indicate that the opposite happens and these synapses are weakened in a process known as long-term depression [14, 26–28]. This means that weights should start high and decrease with learning.

Despite being well established that climbing fiber activity provokes synaptic weakening on Purkinje cells, recent studies suggest that synaptic strengthening also occurs in the cerebellum, although through different mechanisms [29, 30].

2.3 James Albus (1971)

Albus' proposed theory expands on Marr's work but contrasts it in certain points. In his paper from 1971, he also brings the theory of cerebellar function closer to a computational model for pattern-recognition by relating it with the concept of the Perceptron, developed by Rosenblatt [31].

The Perceptron was designed to work similarly to neuronal cells. The cells receive excitatory and inhibitory inputs of variable strength. If the sum of the inputs delivered to a cell exceed a threshold, the cell fires. The standard Perceptron possesses association cells, which preprocess the initial input, and

response cells, which receive the output of the first type of cell. Each association cell will have a different weight on a response cell. It is the proper adjustment of these weights that allows the Perceptron to classify input patterns.

Albus considers the standard Perceptron inadequate for the complexity of patterns likely occurring in the nervous system. The more complex the input patterns become the likelihood of them being recognized decreases rapidly. An alternative would be the binary decoder Perceptron. In this model, there is one association cell for each possible input pattern. This, however, means that the number of association cells increases exponentially.

Finally, Albus proposes something in-between the two aforementioned types, which it calls the expansion recoder Perceptron. In this case, there are 100 association cells per input. If input was treated as a binary pattern, there would 100 cells per bit. This adds enough redundancy to apply restrictions to association cells. The author suggests that only 1% or fewer cells should be active for any input pattern.

Notice that this is similar to the function proposed by Marr for the granule cells. These cells exist between the mossy fibers (the input) and the Purkinje cells (the response cells) and make input patterns more easily recognizable, like association cells. Albus considers granule cells to be the recoders of the cerebellum as there are 100 to 600 more parallel fibers than mossy fibers.

Both authors agree that Golgi cells are responsible for keeping granule cell activity at a constant rate. However, Marr considered that Golgi cell dendritic arborizations were non-overlapping, basing his assumption on the work of Eccles et al. [20]. However, this idea seems to be biologically implausible [22]. In his theory, Albus corrects this, considering that Golgi cells do overlap. These cells regulate parallel activity both by their input and by mossy fiber input. The parallel fiber path involves delays since the granule cell has already fired, but more accurate. The mossy fiber path is faster to stabilize parallel fiber activity, but less accurate. According to the study mentioned in section 2.1, Golgi cells seem to use the mossy fiber path most of the time [19].

As mentioned before, climbing fiber activity causes a Purkinje cell to fire automatically. Albus goes into more detail into this interaction as it is essential for how a Purkinje cell learns to respond to input patterns. It seems that after firing from climbing fiber excitation, the activity of the cell pauses momentarily. This moment is called the inactivation response and the author suggests that parallel fiber synapses to the cell are adjusted at that time. This is one point where this theory contrasts Marr's. Marr states that the synapses are strengthened by climbing fiber activity while Albus argues that they are weakened, as supported by the studies referenced in the previous section.

A great analogy is made for how a Purkinje cell learns by relating it to classical conditioning. Basically, climbing fiber activity is an unconditioned stimulus (US) that provokes an unconditioned response (UR): the Purkinje cell firing and pausing. At around the same time the US occurs, mossy fiber activity is happening, which is the conditioned stimulus (CS). After several expositions, the isolated occurrence of the CS will trigger a conditioned response similar to the UR triggered by the climbing fibers. In other words, Purkinje cells will respond to the isolated mossy fiber input without climbing fiber activity.

Albus argues against Marr's idea that only excitatory parallel fiber synapses on Purkinje cells are adjustable stating that inhibitory inputs on these cells should also be modifiable. His reasons are that if

only excitatory inputs are weakened, eventually the cell will not fire on any pattern; adjusting inhibitory weights increases significantly the storage capacity of a cell; and that varying only on set of weights makes learning slower. By considering that there is plasticity in both excitatory and inhibitory inputs, it is suggested that the number of patterns recognizable by a Purkinje cell increases to 200000 from the 200 figure provided by Marr.

The combination of both of these influential theories came to be known as the Marr-Albus theory of cerebellar function. Although the theory led to many studies that confirmed some of its predictions [6, 16, 32], there is still disagreement as to the actual functioning of cerebellum [33, 34].

2.4 Franz Mechsner (1996)

Franz Mechsner proposed in 1996 a new theory in which the cerebellum performs a coupling operation that creates associations between simple movements that are often performed together. This coupling operation would thus allow accurate and smooth execution of complex movements. The model we will propose later in this dissertation will have this theory as a base.

The essential concept of the theory is the elementary movement (EM). Just like elemental movements in Marr's work, elementary movements are the simplest forms of movement or muscle activation, such as bending a finger or activating the abdominal muscles. By executing multiple EMs together complex motions can be performed. However, in this theory EMs are variable, as the brain can vary their intensity, speed and duration of execution.

One of the main ideas of the theory is that the cerebrum has limited control capacity. Therefore, there is a limit to the number of EMs that can be combined into a single motion. Not only that but also the speed and accuracy of a complex movement will be affected by the limited attention span of the cognitive apparatus.

This is where the cerebellum comes in. The coupling operation performed by the cerebellum groups EMs frequently executed together into EM clusters. The cerebrum can then control these EM clusters as if they were simple EMs. Consequently, this means that by combining EM clusters with EMs or other EM clusters, the cerebrum can control a larger number of EMs simultaneously and, therefore, execute movements that are even more complex. On the other hand, it can also use the control capacity freed up by the creation of EM clusters to increase the speed of a complex motion or increase its accuracy.

In other words, the coupling operation allows the cognitive apparatus to add more variability to a learned movement, giving it the ability to adapt to different contexts and situations.

Research has found that the cerebellar cortex is divided into many small compartments whose cells exhibit common properties, denominated microzones [35–37]. The present theory is constructed all around this concept. Purkinje cells from the same microzone inhibit the same group of cells in the deep cerebellar nuclei. Additionally, climbing fibers seem to synapse with cells within the same microzone.

Mechsner proposes that each microzone controls the execution of a single elementary movement. However, each EM has multiple microzones controlling it. This property allows for refined control of movement, as it is suggested that participation of multiple microzones is necessary to elicit movement.

The set of microzones manipulating the same EM is part of a cerebellar module.

For each EM there is a cerebellar module responsible for its activation. Mechsner explains that just as the cerebellar cortex is divided into microzones, the cerebellar nucleus is divided into compartments. Each compartment provides the command to execute a specific elementary movement. A cerebellar module is thus composed by a nuclear compartment and by multiple microzones. The module's microzones project exclusively to the compartment that provides the output of the module. Naturally, all microzones controlling the same EM are contained in the same module and no other microzone is present.

Another consequence to take into consideration is that the existence of multiple microzones in a module means that a single microzone is not enough to determine the output of its module, just like a single parallel fiber has little influence over a Purkinje cell.

Every microzone and compartment belongs exclusively to a single module. However, there are synapses between microzones belonging to different modules. Parallel fibers originating from granule cells from a microzone will establish synapses with Purkinje cells in neighbouring microzones and not only with the ones in the same microzone. In this theory, microzones that have parallel fibers in common are considered neighbours. It is this interaction between microzones that allows the coupling of elementary movements.

At the start, when the cerebellum is still untrained, all parallel fiber synapses to Purkinje cells are at their maximum strength. This condition is added to the proposal that Golgi cells keep overall parallel activity constant. From this combination results that the Purkinje cell will be excited beyond its threshold and thus constantly inhibiting its corresponding nuclear cell, preventing its elementary movement from being executed. As learning progresses, parallel fiber synapses will weaken. If an input pattern is presented that activates mostly weakened fibers for a particular Purkinje cell, its excitation might dip below its threshold. When this happens, the corresponding nuclear cell will not be inhibited anymore and it will send the signal to execute the EM. However, in order to actually elicit the EM, this process must occur in a big enough number of Purkinje cells since each individual cell has little influence in the overall inhibition.

Mechsner suggested that climbing fibers connected to a microzone signal that the corresponding elementary movement was executed. When climbing fiber activity is present, synapses from active parallel fibers on the Purkinje cell are weakened, including the ones originating from neighbouring microzones. This means that if enough synapses of parallel fibers from neighbouring microzones are weakened, the inhibition of Purkinje cells from a microzone will be stopped by the execution of its neighbours' EMs. Thus, its EM will also be executed. In other words, the execution of a group of EMs will automatically trigger the execution of other EMs. This is the idea behind the coupling operation and the creation of EM clusters.

It is important to note, however, that the participation of multiple microzones from the same module is necessary to trigger the respective elementary movement. In addition, in order to create an EM cluster, a set of EMs must be executed many times. This is to avoid the establishment of clusters of EMs that are rarely performed together. Additionally, a forgetting rule is necessary to forget any undesired clusters

that may be created.

One of the main differences between Mechsner's model and previous works on cerebellar operation is that learned patterns can change over time and be forgotten and can respond to varying inputs and contexts. It is easy to see how this property is an improvement over older models. When a person learns a new movement, the motion improves over time with the repetition of the action. Additionally, deficiencies in the movement can be fixed and bad habits forgotten with enough training. It is also important to note that the person will not be able to perform the motion in the exact same conditions every time and will have to adapt it to the current situation. Taking all of this into consideration, a model that produces fixed patterns in response to fixed input is not an accurate representation of the operation of the cerebellum.

2.5 Sensory Prediction

For our work, we plan to expand Mechsner's model, which we described in detail in the previous section. For this purpose, one of the main features we find missing is the processing of sensory information, in particular, the prediction of incoming sensory stimuli.

A lot of research has been conducted on cerebellar function and physiology since the works of Marr and Albus have been published. One of the most relevant findings is the fact that the cerebellar cortex not only receives sensory input but also is capable of making predictions about the sensory consequences of an action [10, 38–40]. In other words, it predicts the information captured by our senses after executing a motor command. Examples of such information are the position of an arm after moving, tactile stimuli resulting from an action, acoustic information related to speech articulation and visual stimulus.

It is not hard to understand how the incorporation of sensory information in motor learning could greatly increase performance, learning rate and adaptability of skilled motions. We therefore believe it is worthwhile to explore the research made on this topic and how it could influence a model of cerebellar function.

Sensory input and its prediction can be used to enhance learning and performance in many ways. Evidence of this has been found in numerous experiments [38, 41–43]. The enhancements we will cover in this section are the prediction and attenuation of self-generated stimuli, error detection and correction, and driving the execution of movement using the expected consequences.

For most of this section, we will discuss the role of the cerebellum in predicting the sensory resulting from our own actions. However, it is important to note that the cerebellar cortex seems to be able to make sensory predictions independent of our movement. This is supported by the work of Cerminara et al. [44], which shows that cats are able to make predictions on the trajectory of a moving object even when the view of it is obstructed.

2.5.1 Sensory Attenuation

It has been common knowledge for centuries that people are not able to tickle themselves [45, 46]. A series of studies made by Blakemore et al. investigated the cause of this effect and reveal that the cerebellum seems to be responsible for it [9, 47, 48]. The authors started by studying the differences in neural activity between self-produced and external tactile stimuli, given that the former is perceived to be less intense [47]. For the experiment, they used a device capable of applying a tactile stimulus to the subject's left palm. The device was designed so that the subject's right hand or another person could operate it. In other words, it allowed the stimulus to be self or externally generated. The results showed that there was significantly less neural activity during self-produced stimuli. Additionally, it was found that self-produced movements that create tactile stimuli generate less activity in the cerebellum than those that produce no stimuli.

Based on these findings, the authors proposed that the cerebellum could be responsible for predicting the sensory consequences of a movement. Furthermore, they suggested that this prediction would then be used to attenuate activity from self-generated stimuli. In a following study, Blakemore et al. [9] tested this hypothesis with an experiment performed on six volunteers using PET to analyze their neural activity.

Subjects were asked to move a small rod in a regular sinusoidal movement. The rod was attached to an optical encoder. By moving it, a robot would reproduce the same motion with a soft piece of foam that would touch the subject's left palm. With this apparatus, it was possible to introduce a delay between the movement of the right hand and its reproduction by the robot by delaying the transmission of the signal. Each run would then have a delay of approximately 0Hz, 100mHz, 200mHz or 300mHz, distributed randomly in 12 runs per subject and each being repeated 3 times. Subjects were not informed that any delays would be introduced randomly during the experiment. They were only aware that moving their right hand was responsible for the stimulation on their left palm.

During the runs, the subjects were scanned with positron emission tomography (PET) and functional magnetic resonance imaging (MRI). The analysis of the data gathered showed a positive correlation between delay and activity in the middle right cerebellar cortex in all subjects. The results of their experiment support the notion that the cerebellum compares and signals any difference between the expected sensory information and the actual feedback of the action performed. Furthermore, the authors suggest that the specific area activated seems to depend on the sensory consequences of the movement executed.

The subjects of the study reported a more intense sensation for the runs with higher delays, even if unaware of this factor. By predicting the consequences of our movements and attenuating their effects on ourselves, tickling ourselves becomes quite distinct from being tickled by someone else. This kind of attenuation would be useful to filter the overwhelming amount of sensory input and focus on the essential information, such as external stimuli or a mistake during the execution of an action. The latter is quite important and will be discussed further in the following section.

By predicting the consequences of our movements and attenuating their effects on ourselves, tickling ourselves becomes quite distinct from being tickled by someone else. The system responsible would allow the nervous system to filter predicted stimuli and focus its attention on the unexpected input,

which could indicate an external interference. Unexpected stimuli could also indicate that a mistake in predicting was made, and would help improve further predictions.

Besides its utility as an error detector, there are other interesting uses for this ability. Sensory prediction and attenuation are not unique to humans and can be found in other species with cerebellum-like structures [12, 49–52]. One of the most notable examples observed is the mormyrid fish [11, 53–55]. This species of fish uses a specialized electrical system for location and communication. The sensory consequences of self-generated electrical signals are attenuated so that they do not interfere with the ones generated by other fish or stimuli related to predators or food. Given that this electrical system is this species primary tool and the large cerebellum it possesses [55] it is hard to ignore the significance sensory prediction might have in the cerebellar cortex.

Although the cerebellum makes the prediction, the inferior olive seems to be the one responsible for filtering sensory stimuli based on that prediction. They are proposed to compare the expected and the actual stimuli and signal the unpredicted sensory consequences via climbing fiber inputs to the cerebellum [10, 47, 56, 57].

2.5.2 Error Detection and Correction

As supported in the previous section, the cerebellum generates sensory predictions and compares them to the actual consequences of an action, the reafferent input. If the expected input and the actual input do not match, then it is obvious that either some internal or external factor interfered with the movement or the prediction was inaccurate. The discrepancy between inputs can be used to perform corrections to the movement and to learn and improve future predictions, adapting until the expected and actual outcomes match.

Nitschke et al. [58] made a study based on the hypothesis that the cerebellum makes predictions and adaptations of the sequence of movements required to complete a task by use of an internal model of the target of the movement. The authors designed a test that consisted on the execution of sequential finger to thumb opposition movements. Two different conditions were introduced: a predictive task, in which the present sequence was repeated; and a non-predictive one, in which the sequence was random.

The subjects were monitored with functional MRI during the tests. The sequence to perform was displayed on a monitor and the subjects had switches on their fingertips to record their execution of the task.

As predicted, the task activated different zones of the cerebellar cortex, with the non-predictive condition activating a larger area and additional zones and displaying a higher error rate. One explanation provided for the increased activation area is the fact that the increased error rate causes an increased firing of climbing fibers [59]. Therefore, more processing could be taking place in order to perform error correction.

It is usually considered that climbing fibers serve the purpose of delivering error signals to the cerebellar cortex. Supporting this notion is the fact that climbing fiber activity decreases as learning progresses [10]. Based on studies focusing on classical conditioning, Devor proposes that the inferior olive

filters predicted sensory input in favor of the unpredicted stimuli [60, 61], as suggested in the previous section. This means that only the unexpected input, the error, will reach the cerebellum via climbing fibers. This supports the notion that the cerebellum learns by correcting errors, which was the concept considered in the theory of Albus [62, 63].

Experiments performed by Martin et al. show that lesions to the olivocerebellar system impair adaptation of movements to novel situations [64]. In the experiments, glasses distort the vision of subjects who are asked to throw balls at a target. Cerebellar patients showed slower or non-existent adaptation. Another relevant study comes from Schlerf et al., which shows that both unexpected stimuli and the absence of expected stimuli produce error signals interpreted by the cerebellum [65].

2.5.3 Prediction Driven Execution of Movement

One additional purpose for sensory prediction is to guide the execution of a movement. In other words, instead of choosing the next action based on the actual feedback of the previous one, the action is chosen based on the prediction of its consequence. The advantage resulting from the application of this predictive strategy is that movement becomes smoother because the system is not waiting to react to actual sensory feedback, which has a considerable delay. When tracking a visual target reacting to the movement of the object after it had moved would be slower than acting directly on a prediction of its trajectory. Of course, for this strategy to work a learning process is required, as untrained individuals will produce inaccurate predictions and must learn from their mistakes in order to improve performance.

Nixon et al. have run several animal experiments on the effects of cerebellar lesions on performance in motor tasks dependent on reaction to sensory information [66, 67]. The results of these experiments and their comparison to studies made on human patients are well explained in his 2003 review paper [10]. In summary, the authors found that, in both humans and monkeys, individuals with lesions would not improve to the same level as healthy individuals over extended training in a task.

2.6 Internal Models

The evidence of sensory predictions being developed in the cerebellar cortex provides support to the idea that the cerebellum instantiates internal models of the motor system and of the targets of our actions. These have been mentioned briefly throughout the previous section and will be properly explained in the present one, along with examples of theoretical models and implementations based on this concept.

It is often suggested that the cerebellum might use internal models for motor control [68–71]. An internal model treats the execution of an action with the system (in this case, the central nervous system) as the controller that commands a plant, the part that performs the action (e.g. an arm). Francis and Wonham first suggested the idea in 1976 [72].

There are two types of internal models: forward models and inverse models. Forward models take a copy of the motor command, which is called the efference copy, and uses it to predict the sensory

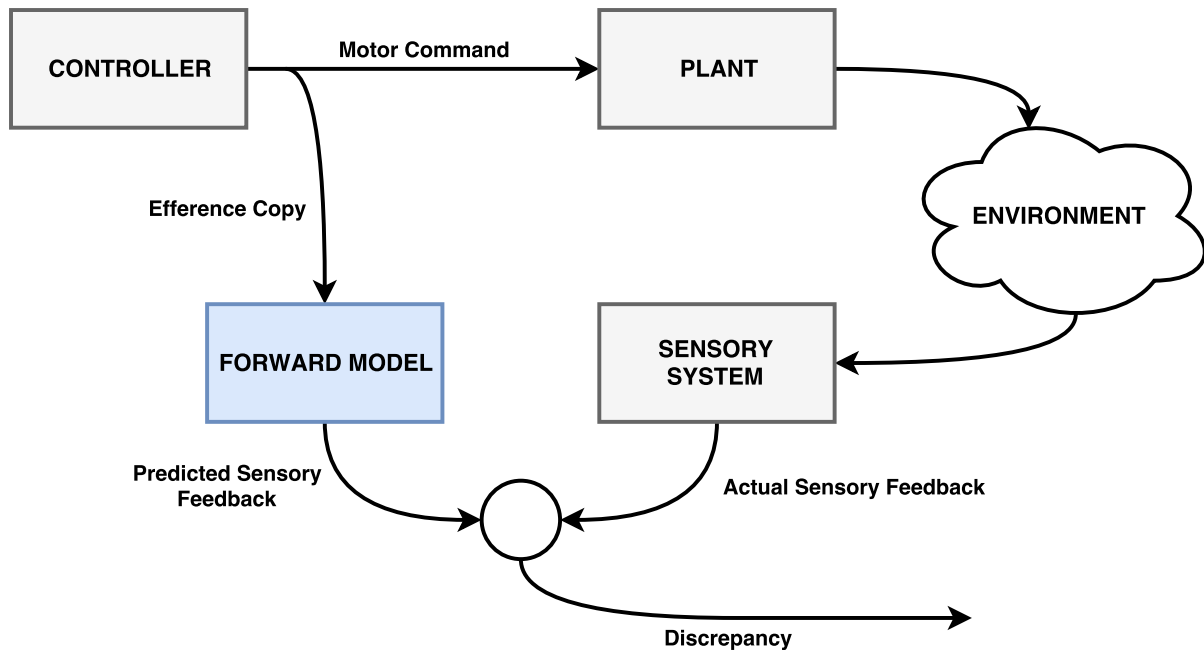


Figure 2.3: Application of a forward model in a system. The discrepancy between predicted and actual feedback can be fed back to the forward model as error signal to improve the model or to the controller to generate the appropriate corrective commands.

information that will result from that command. Differences between the expected and the actual sensory input (called refference) indicate an error that can then be used to adjust the subsequent motor commands [16, 39]. Figure 2.3 shows an example of such a system. The inverse model receives the desired final position for the plant and outputs the motor commands to be performed in order to achieve it. Forward models and inverse models can be combined in order to design solutions that are more complex.

An interesting use of efferent feedback, mentioned by Bell and Grant [11], is as a substitute to sensory information. The efference copy can be used to deduce the state of the body or characteristics of objects such as its weight.

One of the ideas we would like to explore in this project is if it is possible and effective to combine Mechsner’s model with the concept internal models. Mechsner’s model seems to be a bit restrictive, as it does not take the prediction of sensory information into account. Many studies and research papers exhibit test results that demonstrate that the cerebellum generates predictions of the sensory consequences of actions, as described in the previous section. Much of this research suggests that the cerebellum uses forward models for motor coordination [43].

Despite forward models receiving more attention, resulting from the many experiments made related to sensory prediction, inverse models are also relevant. This is because the motor command errors are not directly known and need to be converted from the sensory stimuli delivered as the error signal [68]. The problem is not simple as there are many forms of sensory input such as proprioception, tactile stimulation, visual input, auditory signals, etc. Additionally, some types of input might be more relevant for some motor coordination tasks than others are, such as auditory stimuli for speech articulation.

An interesting idea introduced by Wolpert et al. is the understanding of different contexts [68]. Instead

of only having one model for the cerebellum, there would be multiple instances. Each instance would adapt to a different context and would have the most influence in the motion executed when faced with its context.

This concept is presented as a model that combines multiple pairs of forward and inverse models [73]. In this model, a special module, the responsibility estimator, attributes different weights to each pair of internal models according to their relevance to the current context and situation. Pairs with higher weight values will have a bigger contribution on the movement performed.

2.7 Cognitive function

Evidence suggests that the cerebellum also participates in cognition, meaning it does not only have a role in motor function [74–78].

There is evidence that the cerebellum participates in the acquisition of conditioned responses [79–82] and some authors also suggest a participation in language and reading [70, 78, 83].

Matsuda et al. made an experiment in 2017 to find out how cerebellar neurons control fear conditioning [80]. For the experiment, they used the larvae of zebrafish. Turning off an LED was the conditioned stimulus and an electric shock was the unconditioned stimulus. The results revealed that inhibiting granule cells prolongs the fear response and that activity of cerebellar neurons increases during conditioning. The authors proposed that granule cells control recovery from fear conditioning.

Chapter 3

Proposed Solutions

The cerebellum seems to be a site dedicated to the learning and automation of frequent tasks. In addition, given the links to cognition, these do not necessarily need to be motor related. We think this automation function is greatly aided by the fact that the cerebellum has connections to multiple points in the central nervous system and, therefore, would be able to receive input from many centres. With a trained cerebellum, the cerebrum would only need to provide the initial stimulus and the desired action would be performed automatically without need for further intervention. This function would be of great importance as it relieves the cerebrum from repeated processing and recalling tasks.

A real life example of this operation should provide a clearer picture. One method to increase productivity and prevent mistakes is to create schedules or lists for the tasks we have to do or the items we need to purchase at the supermarket. This relieves us from having to constantly recall what we need to do next and lets us focus on the task at hand. Another example would be cooking recipes or any type of written instructions. We just need to follow the instructions without actually having to think about what we are doing.

Our hypothesis then is that the cerebellum performs a similar function, albeit at a more complex level, for the automation of tasks in the nervous system, in particular motor control. Adding to this function is the ability to fine tune the parameters related to a task. The idea is not unique and has been suggested by many authors such as Marr, Nixon and others mentioned in this dissertation.

As stated in the Introduction, our intention is to expand on Mechsner's model by introducing concepts learned from recent research and proposed theories on cerebellar function. One of the main ideas we want to explore is the processing of sensory information and prediction of the sensory consequences resulting from a movement.

Our idea for how to integrate sensory information into Mechsner's model is to treat sensory stimuli the same way as elementary movements. This way it becomes possible to couple sensory stimuli with movements. In other words, a complex motion would be associated with the corresponding sensory consequences. It is also important to clarify what exactly we consider sensory information. It is not simply tactile stimuli or the position of a body part, but also auditory input, visual input such as the movement of an object, balance information, resistance to movement and others.

With the introduction of sensory stimuli as elements that can be coupled it would be misleading to also use the term elementary movement to refer to these. Therefore, we will begin to use the tentative term elementary stimulus (ES) when discussing sensory input in the context of the proposed model. When speaking generally of both elementary movements and elementary stimuli that are coupled by the model we will use the provisional term elementary components (EC).

In our proposal, context is the conjunction of command and sensory input. Consider a trained cerebellum. When given the initial command by the cerebrum paired with the sensory input being received, this cerebellum should be able to automatically reproduce the adequate skilled motion for the given situation. The executed command and the predicted sensory consequences would then become the context that will trigger the following action. This cycle would continue until the execution of the skilled motion was concluded. Adjustments could be made online via discrepancies from sensory input and active intervention by the cerebrum until the desired and actual outcomes matched. Given that the cerebellum is automating the movement at this point, the cerebrum now has more processing power available to adapt the action to new situations as Mechsner proposed.

Another idea we want to implement is the detection of discrepancies between expected and actual outcomes in order to perform error correction as described in section 2.6. Discrepancies in forward models are used as a learning input to improve the model and this would be an important addition to our proposal.

One of the issues we found is that Mechsner's theory considers climbing fiber input not to be an error signal. This contradicts the findings presented in subsection 2.5.2. The research is consistent in finding that climbing fiber activity peaks during the initial stages of learning, where the greatest amount of errors occurs. Once a subject becomes skilled at a task, climbing fiber activation decreases significantly. While reading the work of Marr [8] we came across a possible solution for this issue that is consistent with the coupling operation model and the research work on this topic. It is as follows.

The cerebrum sends an efferent copy of the motor command to the olivary cells. Since the cerebellum has not learned the sequence of EMs yet, it will fail to reproduce the sequence accurately. Therefore, a signal from the inferior olive must be sent via climbing fibers to facilitate the synapses on the PCs that would produce the required EMs. Once the sequence is learned, commands from the cerebrum will be filtered in the inferior olive since they match the commands sent by the cerebellum, which means no error correction needs to take place. This is just like what happens with predicted sensory input. If it matches the actual input it is attenuated, if not an error signal is sent via climbing fibers.

While on the sensory side the reafferent input is the reference for correctness, on the motor command side the reference is the intentions of the conscious parts of the brain. This means that if an individual decided to consciously adjust a parameter of a learned motion, the climbing fibers would signal the discrepancy in the sequence recalled by the cerebellum. Slowly, with regular and frequent execution of the same adjustment, the learned motion would change into the desired outcome at which point conscious effort to perform it would not be required anymore.

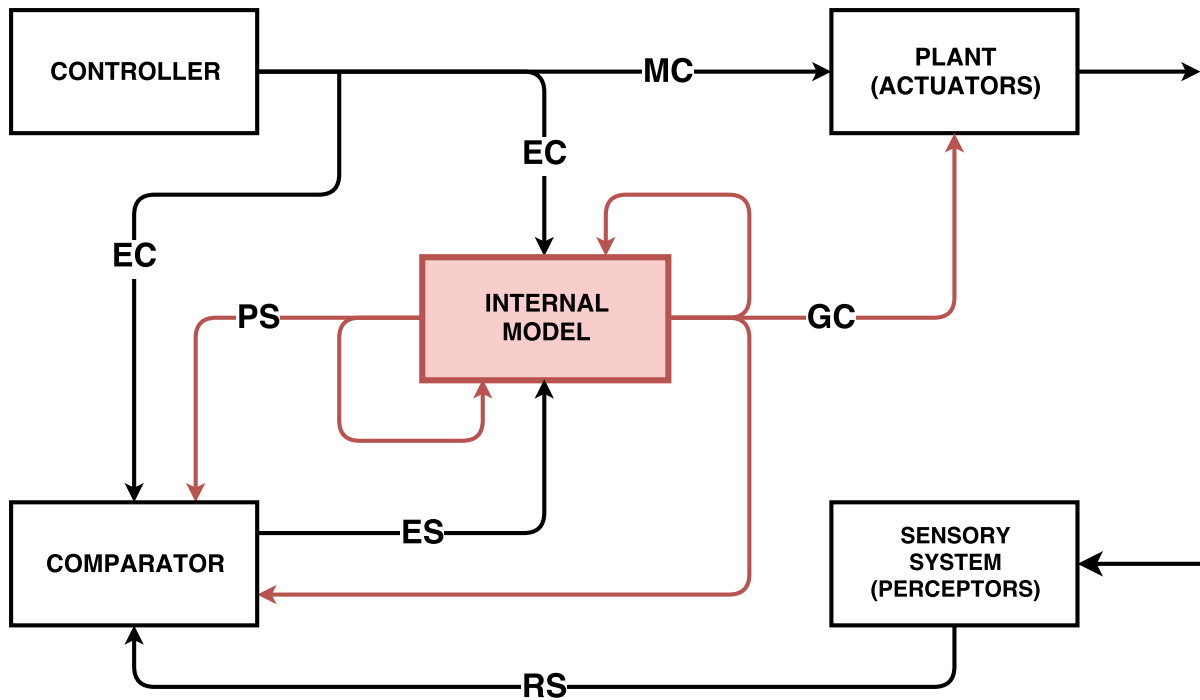


Figure 3.1: Architecture of the proposed model. MC: motor command; EC: efference copy; GC: generated command; PS: predicted stimuli; RS: reafferent stimuli; ES: error signal.

3.1 Solution Architecture

Five distinct types of modules compose the proposed model: controller, plants, internal model, comparator and sensory system. The controller, which performs conscious action and decision making akin to the cerebrum, commands the plants. The plants are the actuators that perform the action commands, acting on the environment. These would correspond to the body parts participating in a movement. The sensory system is composed of the perceptors, which capture sensory stimuli, providing the system with the current state of the environment. The comparator module filters the stimuli delivered by the sensory system, which can be overwhelming in its amount. The filter implemented by the comparator is guided by the internal model, which predicts the outcome of the actions of the system.

The internal model is the central part of the model. Corresponding functionally to the cerebellum, it will implement the coupling operation proposed by Mechsner. It keeps track of the sensory input and the commands issued by the controller, and tries to learn command and stimulus combinations. This learning operation will allow it to automate frequently executed tasks, relieving work from the controller and enabling smooth execution and adaptation of the tasks. This means that the internal model can also send commands to the plants and should be able to coordinate the activity of distinct controllers participating in a task.

The architecture of the model is presented in figure 3.1. Not depicted is how sensory information reaches the controller and what kind of processing it goes through.

Our idea for the functioning of the model cerebellum is as follows. Controllers would send their movement commands to the plants, to the internal model and also to the comparator. The command sent to the cerebellar module is an efference copy as described for forward models. The module makes

a prediction on the sensory consequences of the action and sends it to the comparator. The comparator, based on the internal model's predictions, would filter sensory stimuli arriving from perceptors.

Note that the efference copy of the command is sent to the comparator as well. The purpose of this is to provide to the comparator module the information necessary to detect discrepancies between the action currently learned by the internal model and the desired result. The comparator then sends these discrepancies and the ones pertaining to sensory input as error signals to the internal model, which will use them as learning input to improve itself.

There is, however, one major difference between sensory and action command circuits. While sensory input will always occur and will always be delivered from sensory receptors, motor commands will eventually cease to be sent by the controller as the internal model becomes skilled at reproducing the motion. This is intended since the desired effect is to free the processing capabilities of the controller from repetitive tasks so it can perform higher-level computations. However, this means that the comparator will not have input to detect discrepancies between the desired movement sequence and the one reproduced by the model. In the case of sensory stimuli, expected and actual inputs are always compared. Therefore, the implemented model will have to accommodate this difference in input treatment.

Although the model proposed here implements an internal model of the motor system, we can observe that it is neither strictly a forward model nor strictly an inverse model. When receiving an efferent copy of a motor command it not only generates a prediction of the sensory response but also generates the commands that need to be performed simultaneously or following it.

To summarize, the proposed model will implement Mechsner's coupling operation, enhancing it by adding the ability to predict the sensory consequences of an action. The discrepancies between expected and actual outcomes will be used to train the model.

3.2 Applications

In this section, we want to present some ideas for possible applications for this kind of models. The implementation of these ideas probably will not be possible in the time provided for this project but could be the topic for future work if the proposed solution achieves satisfactory results.

One of the main applications of this kind of computational model is, of course, motor control. In a posterior project, the proposed model could be applied to a robot inspired by biological structures. The model would allow the robot to produce natural and fluid motions with its limbs, as well as aiding it in keeping proper posture and balance. The learning and optimization of skilled movements by robotic arms would prove to be useful both in the engineering and medical fields.

A more specific example could be gaze stabilization. Using the predictive abilities of the model, a camera could be made to follow a target object smoothly by predicting its movement trajectory. In the case of balance and posture, if the model predicted that a certain action would throw the robot off balance it would trigger the necessary counter-actions to keep equilibrium.

We believe that the focus on motor coordination in the human cerebellum is related to the fact that

this is the system human beings possess to act on their environment. Species with novel systems, such as the given example of the mormyrid fish and its electrical system (in subsection 2.5.1), use their cerebellum-like structures to coordinate other kinds of actions. What we intend to impart from this is that it should be possible to adapt this model to other systems unrelated to motor control. It could be applied to virtual or real robots with different mobility features and sets of actuators and perceptors. These systems would have their own set of ECs and the model should be able to adapt to them.

It would be interesting to adapt this solution to types of patterns other than motor patterns or to attempt to simulate specific behaviours, like the ones resulting from cerebellar lesions or from altering the parameters of a task. One such example would be the effects of trying to be perform a task much faster than how it was learned.

The faster we try to perform a task the more we start to jumble the elements of the task together. For instance, when we attempt to write faster than what we are used to, we tend to mix the strokes of consecutive letters or skip them altogether. This could be the effect of the cerebellum adjusting the parameters of the writing motion and missing the essential timings for the composition of each letter. As writing letters is an automated motor task, a learned skill to which we need not think actively, it is plausible that this task is delegated to the cerebellum. It would be interesting to see if was possible simulate such behaviour with a computational model capable of reproducing handwriting.

Other areas that come to mind where such a computational model could prove to be useful are natural language processing, associative memory and speech production.

Chapter 4

Methodology

To implement our basic model of the cerebellum we first needed to choose a problem to solve. The problem should be simple and the implementation of the model would revolve around it. In other words, the model would be adjusted specifically to solve this problem.

Irregularities in individuals with cerebellar lesions can manifest by the decomposition of movements into their basic elements, which are executed separately in sequence instead of simultaneously [1]. The purpose of this task is then to do the opposite: to combine elementary movements into a smooth motion.

We were inspired by figure 4.1. Imagine trying to reproduce the upper trace shown in the figure using only the arrow keys of a keyboard to control a cursor. Basically, the elementary movements involved in this task are the up, down, left and right movement actuators. The variables of the task are quite simple: the longer a direction is held the further the cursor will travel, always at a constant rate. However, the execution of these movements must be highly coordinated in order to produce a smooth trace. Given the difficulty of this task for a human subject, we think it would be impressive if the proposed model could achieve it.

With this in mind, for our problem we chose to move a cursor between two points in a bidimensional space using a set of simple elementary movements. The function of the model cerebellum would be to learn the trajectory to follow when confronted with different scenarios and execute it automatically with little or no assistance. As the model evolved, it would learn to combine elementary movements and to make the trajectory smoother.

To implement the model we decided to use the Python programming language¹. Python is a simple language, which made it very easy to start building the model and experimenting. We began with a very simple implementation of the model and progressively made it more and more complex. In total, the model went through nine different versions, which are described in detail in chapter 5.

At some point during the development of version eight, we introduced a graphical representation for the model. This representation would allow us to visualize the trajectories being produced by the model and better understand its behaviour. We decided to develop it using Pygame², a Python library for making multimedia applications. This library is easy to use and, with its geometry functions, we were

¹<https://www.python.org/>

²<https://www.pygame.org/>



Figure 4.1: Trace produced by a cerebellar patient (below) while trying to reproduce the trace above. From a book on cerebellar function by André-Thomas [1].

able to build a nice visual representation for our model quickly.

For the last few versions we made some experiments to test the performance of the model and the strategies we developed to solve the proposed problem. The experiments and the results are described in chapter 6.

Chapter 5

The Evolution of the Model

The goal of the model is to attempt to simulate the behaviour of the human cerebellum. The cerebellum is capable of performing skilled movements. When learning a new movement pattern there is at first a lot of participation from the cerebrum, the conscious part of the brain. However, the work capacity of the cerebrum is limited and this limits the complexity of the movements it is capable to produce and their speed and accuracy. It is then proposed by researchers such as Mechsner [13] that as the cerebellum learns these complex movements it lowers the load on the cerebrum. This frees up the work capacity of the cerebrum allowing it to make adjustments to the movement, increasing the speed at which it can be performed, its accuracy or even combine it with other movements and make it more complex.

With this model we wanted to attempt something similar. As the model cerebellum learns a movement pattern when exposed to a given scenario it should start taking the initiative and take the effort of the decision from the model cerebrum. The model cerebrum has an algorithmic and well-defined method of making its decision to move in a given situation. The model cerebellum, however, will have a more reaction based decision process, deciding on its actions according to what it has frequently observed the model cerebrum perform.

The main components of the model are the agent, the environment, the cerebrum and the cerebellum. Both the cerebrum and the cerebellum are capable of moving the agent. The cerebrum teaches and the cerebellum learns.

The first versions of the model were very simple and implemented the basic components. As the model evolved, we started experiment with various strategies that determined how the model cerebellum made associations and how it chose the elementary movement the agent should execute. Eventually, a graphical representation for the model was also introduced.

The sections describing each version of the model will all follow a similar structure. Each section starts with a brief introduction of the changes made. Then, there will be multiple subsections describing the different aspects of the model that were changed. There will be a subsection explaining how the agent moves, how the goals are defined and met, the model cerebrum and the model cerebellum. After that, we describe how the model behaves. To finish, the last subsection goes into what problems we find in that version and what improvements could be made. Additional subsections may be added to explain

new concepts introduced in a specific version. In addition, some subsections may be omitted if there was nothing relevant to mention for those aspects of the model.

5.1 First Version of the Model

This is the first version of the model we implemented. We started by defining how the agent can move, what are the goals and how they are met. We then implemented a simple model cerebrum that guides the agent to the goal. The model cerebellum observes the commands issued by the cerebrum, creating associations based on the most recent ones. It uses those associations to contribute to the movement of the agent.

After describing the model cerebrum and cerebellum, we describe how the model itself behaves. We end by explaining the problems we found with the current model and explaining what needs to be improved. The sections dedicated to the subsequent versions will follow a similar structure to the one used for this version of the model.

5.1.1 The Agent's Movement

For the first version of the model, we defined four different elementary movements. These are very simple movements, which will be executed by the agent in order to move towards its current goal. The four movements are: up, down, left and right. The environment is a bidimensional rectangular plane and the agent moves in the direction of its vertical or horizontal axis.

The agent always moves a fixed distance when executing a specific movement unless there is interference from the model cerebellum. The model cerebrum is not able of moving the agent diagonally. However, the movement components added by the model cerebellum could potentially make the resulting movement diagonal.

5.1.2 Goals

In this version, the start position and the goal position were fixed. This meant that every run of the model would produce the same result and behaviour. This setting made it easier to make sure that the basic features of the model were working properly and fix any problems.

The agent had a list of goals in this implementation. After reaching the first goal the model would guide it towards the next one and so on until it arrived at the last goal. The idea behind this list was that the intermediate goals would eventually become checkpoints that the agent would have to go through instead of just moving straight towards the final goal. The intention was that, as the model evolved, the model cerebellum would learn to produce a trajectory that met all of these checkpoints before reaching its destination, executing a complex movement as a result. The cerebellum would then learn to optimize this trajectory, making the movement more smooth and natural.

For the model to accept the agent as being located in a goal position it is necessary to be at its exact position. This did not seem as big of a worry as it may seem because the possible positions for the goals

and the agent are spaced a fixed distance apart. This distance being the distance travelled by the agent after performing an elementary movement. This would make it appear as if all possible positions are organized in a grid, the agent moving from one cell to another until it arrived at the goal cell. However, this is not the case. As will be explained in more detail later, the implementation of the model made the task of reaching the goal more difficult. Additionally, the intention was to make the positioning of the goals more organic, without requiring such a rigid placement.

5.1.3 The Model Cerebrum

The model cerebrum at this point is very simple and it will not grow a lot in complexity in the following versions. As the main focus is in the model cerebellum and its behaviour, we took a lot of liberty in the implementation of the model cerebrum. The role of the cerebrum in this model is to act as a teacher for the cerebellum. The model cerebrum controls the agent, choosing which elementary movements it should perform at a given moment, and the model cerebellum observes its actions, learning how it should behave.

This setup gives us the freedom to design specific learning scenarios and program the model cerebrum accordingly. For instance, we could instruct the cerebrum to follow a trajectory that resembles a letter of the alphabet and try to teach the model cerebellum to reproduce it.

In the current implementation, however, the model cerebrum simply moves towards the current goal. It checks the position of the agent relative to the goal and chooses an elementary movement that would move it closer to its target.

5.1.4 Reacting to Recent Movement Decisions

In this first implementation the model cerebellum does not take into consideration its position on the environment or its position relative to the goal in order to learn how to command the agent to move properly. Instead, our initial idea was to take into consideration a number of the most recent movements performed. Basically, the intention was that the model cerebellum would learn which elementary movement was more likely to be executed after another one. In order to be more confident in its decision the model cerebellum would analyse a number of recent movements instead of only paying attention to the most recent one.

To give an example, if the sequence "right, up, right, up" was performed frequently the model cerebellum should start reacting by issuing the commands to move up or right whenever a similar pattern occurs. Ideally, this would serve to free the model cerebrum of the burden caused by repetitive actions and let it focus on adjusting the movement or make it more complex. On the other hand, by working together with the cerebrum, the cerebellum should be able to move the agent diagonally by adding the appropriate movement component. This could be achieved, for example, by combining the up and right elementary movements.

5.1.5 The Model Cerebellum

In this first iteration of the model, the cerebellum tries to learn which movement is more likely to be the correct action based on what the previous movements were. For instance, if the cerebrum sent the command to move up three steps ago and, in the current step, it decides to move to the left then the association between these two elementary movements is strengthened by the cerebellum. Therefore, the next time the cerebrum chooses to move up, the cerebellum will add a command to move left three timesteps after it happens. The distance covered by this movement to the left will depend on the intensity of the association between the two elementary movements.

It is important to note that an elementary movement may associate with itself thus increasing the distance travelled by the agent each time the model cerebrum executes that elementary movement. It should also be clarified that it is not only the third most recent step that is analysed but also the other more recent steps in-between. These steps will also contribute to influence the movement executed by the agent in the current step. Also more than one elementary movement may be chosen as a result of this analysis. If in the past, two timesteps after moving up, the model cerebellum observed both instances of moving to the left or moving to the right then it would add a left component and a right component simultaneously when the same context is observed again. The intensity of each component will depend on how much each was observed in that context, which influences the strength of the associations.

There is a limit to how much an association between two elementary movements may be intensified. When this limit is reached, further observations of one elementary movement being performed after another will not strengthen the corresponding association. Of course, other associations can still be strengthened. This measure serves to limit the distance the agent can travel in each timestep. At the max strength, the movement component added by the model cerebellum will move the agent the same distance as an elementary movement issued by the model cerebrum.

5.1.6 How the Model Behaves

At the beginning of each timestep, the model cerebrum chooses which elementary movement the agent should execute. Then, based on this command and previous ones, the model cerebellum adds its own contribution to the current action. In addition to this, the model cerebellum strengthens the associations between elementary movements by observing the sequence of movements performed by the model cerebrum so far. This implementation of the cerebellum establishes associations between elementary movements and the time interval that separates their execution by the agent. An elementary movement may also associate with itself.

The idea is that movements executed recently will influence the current action of the agent. A movement associated to the recent one will also be executed in the current timestep along with the command from the model cerebrum. However, unlike a movement command issued by the model cerebrum, which moves the agent a fixed distance, the distance travelled will depend on the strength of the association. Additionally, if the recent movement is associated with multiple movements all of these will add up to the

movement issued by the cerebrum in varying degrees. The movement executed by the agent is then the result of adding the movement chosen by the cerebrum and the movement components calculated by the cerebellum.

The model cerebellum keeps track of these associations and updates them at the end of each timestep, after the movement to execute has been chosen.

5.1.7 Problems With This Version

In order to reach the goal the agent must reach the exact position where the goal is located. There is no margin of error and, given the method used to establish the results of a movement command, it would be very unlikely to arrive at the accepted location. What ends up happening is that the agent keeps jumping back and forth around the goal without stopping, unable to fine-tune its movement to get closer and closer to the correct position. The model at this stage is unable to recognize it has fallen into this situation and take actions to solve it. In later versions of the model, we explore some ideas focused on stopping this from happening, allowing the agent to move into an acceptable distance to the goal.

The method the model cerebellum uses to intervene does not produce a behaviour similar to its suggested biological function. The cerebellum should learn to replicate the movement performed by the cerebrum. However, what ends up happening is that the model cerebellum begins to intensify the movement that is executed, increasing the distance travelled by the agent and making it unable to make fine adjustments to the movement. In other words, the model cerebellum is reinforcing an elementary movement instead of learning to predict when a new movement in a different direction should be performed, which is what was intended. The idea was that the model would use recent actions to predict what the next action was likely to be whether it was to continue forward or change direction.

This behaviour is the opposite of what we want the model to achieve. Although it would be beneficial if the model was also capable of making some big and fast movements, it is of no point if it is unable to perform the small corrections necessary to stay on the correct trajectory. Another behaviour we hoped to see was the combination of movements that would move the agent diagonally on the environment. However, this type of movement is unlikely to occur because of the issue described above.

One of the biggest problems is that the model does not unlearn any associations it makes between elementary movements. More specifically, once the model strengthens an association there is no method of decreasing it. Although it could be nullified by strengthening the association with the opposite movement, this situation does not happen in practice and, even if it did, what would happen is that eventually neither of the associations will have any influence on the resulting movement. Imagine the association between moving up and moving to the left. Every time the agent moves up it will add some movement to the left afterwards. If the association between these two elementary movements is completely strengthened, there is nothing to stop this movement pattern from happening every time the agent moves upwards unless the association between moving up and then right is intensified. If this occurs, the added movement to the left will become smaller and smaller until it is completely nullified by the right movement. However at this point both associations have been maxed out which means it has become

impossible for the model cerebellum to learn to move to the left or to the right after moving up.

For this version, the model cerebellum takes into consideration the last three elementary movement commands issued by the model cerebrum. If all of these movements were associated with moving up for their respective timestep intervals, then the elementary movement "up" would be added to the current action in varying degrees. In an extreme scenario, the resulting action would move the agent four times the normal distance for an elementary movement issued by the model cerebrum: three movement components to move up at maximum strength added by the cerebellum plus the elementary movement "up" issued by the cerebrum.

Another problem is that the cerebellum never becomes independent from the cerebrum. The model cerebellum is always looking at the commands issued by the model cerebrum to decide how it should influence the resulting movement performed by the agent.

5.2 Second Version of the Model

In order to solve some of the problems present in the previous version we decided to divide the environment using a grid. Thanks to this feature, we were able to design a simple method for the model cerebellum to associate the movement of the agent with its position on the environment. Furthermore, the creation of the grid made easier to establish a success condition and determine if the agent has in fact reached the goal.

This version was a transition to a new implementation based on simplifying the model cerebellum's perception of the environment. As such, the cerebellum does not participate in moving the agent nor does it learn to recognize movement patterns. It was left out because the focus at the time was to make sure the grid system was working properly.

During this implementation, we started to study the idea of having the model cerebellum associate the possible positions in the environment with an elementary movement. More specifically the cerebellum would associate the cells of the grid with elementary movements. This idea, however, was only put into practice in the next version.

We also started considering separating the execution of the model into two runs: a training run and a test run. For the first run, the model would simply observe the behaviour of the model cerebrum without intervening. The cerebrum would guide the agent alone towards the various goals. During this phase the cerebellum would attempt to learn the movement pattern. For the second run, we would test the ability of the model cerebellum by asking it to reproduce the movement it just observed without any assistance from the model cerebrum.

This concept would simplify the implementation of the model. As we had observed in the previous version things can get messy easily if we let both the cerebrum and the cerebellum act at the same time. It is difficult to coordinate their behaviour, as they will interfere with each other's planning. It also makes it harder to understand the behaviour we are observing and understand what is going wrong. By separating the two, the model becomes simpler and it makes it easier to analyse them individually. It also becomes easier to see if the cerebellum is in fact learning to perform a complex movement, which

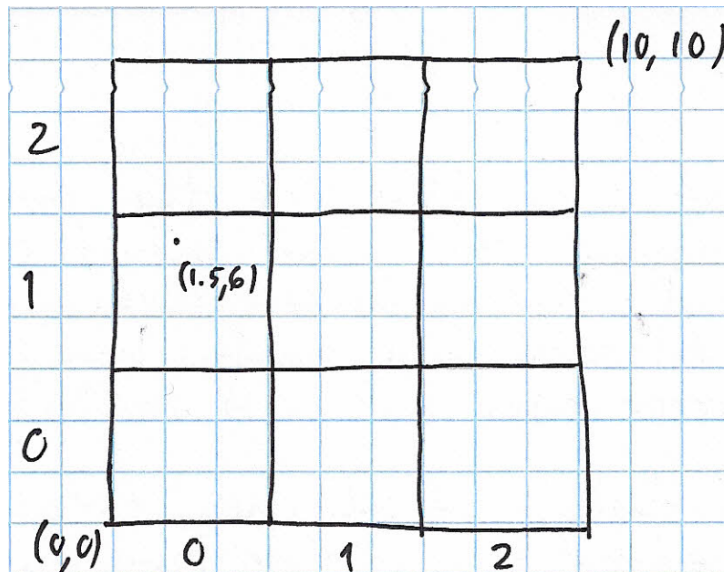


Figure 5.1: An example of the grid dividing the environment. The point (1.5; 6) will be represented by the cell that contains it, the cell (0; 1).

is the main goal of the model.

5.2.1 The Grid

The model divides the bidimensional environment using a grid. This process divides the playing field into rectangular shaped regions of equal area, which we will refer to as cells. To ensure a correct implementation and make the analysis of the model's behaviour easier we started with a grid of a low number of cells. For this implementation we used a three by three grid with a total of nine cells. This gave the agent enough cells to move around, a good variety of start and goal position combinations and made behaviours simple to understand.

The current version of the model takes the positions of the agent, checkpoints and the goal and translates them into the corresponding cells of the grid. Figure 5.1 helps illustrate the concept. The position with coordinates (1.5; 6) is located in the region of the cell (0; 1) and therefore the model translates the coordinates as such.

After performing the translation of all relevant positions, all decisions and learning done by the model are based on these translated coordinates. Therefore, when the agent moves it is considered it moves from a cell to another instead of moving a certain distance in the real environment. It should be easy to see how this sort of implementation may become an issue, as the agent might travel a short distance and stay in the same cell instead of crossing into another one. We will discuss this issue and what we did to solve it later on.

It can be easily inferred that the bigger the division of the environment the more precise the movement will be. The more we divide the field the more cells we will have and the smaller each step will be. This in turn allows the agent to make small adjustments to its trajectory.

The more we increase the number of cells in the grid the more the resulting trajectory begins to look like a smooth line with curves instead of jagged lines. This phenomenon is similar to the proposed

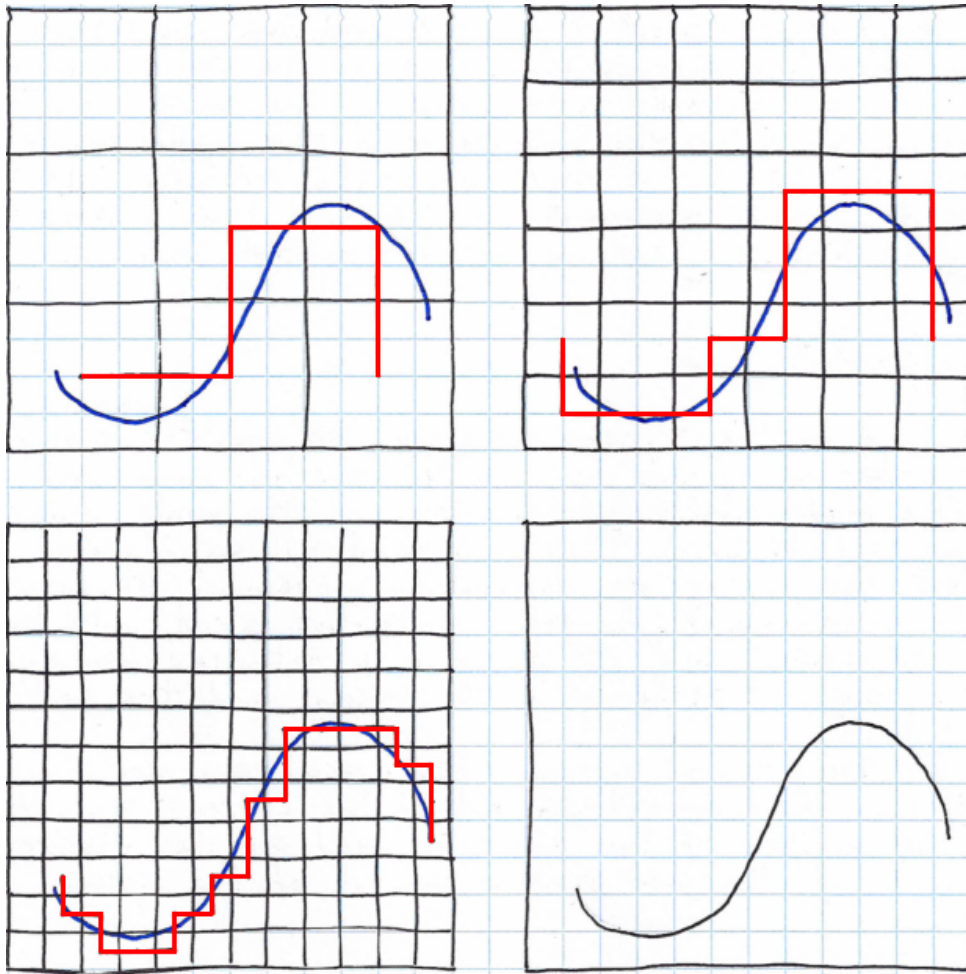


Figure 5.2: An example of the same movement being represented in grid with different cell density. The grid on the bottom left allows a more accurate reproduction of the movement pattern.

function of the cerebellum. When a person begins to learn a new complex movement, they are inexperienced. Their movements are clunky and uncoordinated. However, as the person practices, the movement becomes smoother and more accurate. It is thought that the cerebellum is an integral part in learning skilled movements freeing the work capacity of the cerebrum. We explore the concept of increasing the number of cells in order to improve the trajectory of the agent in later versions of the model.

5.2.2 The Model Cerebrum

The algorithm for the model cerebrum's decision process remained fairly the same from the previous version. The only difference is the adaptation to the movement on the grid. The cerebrum still tries to move straight to the goal using one of the elementary movements that will move the agent closer to it. In this version, instead of ordering the agent to move a fixed distance from its position, the model cerebrum orders the agent to move from one cell to another.

5.2.3 Reaching the Goal

In this version of the model, we consider that the agent reaches a goal, either a checkpoint or the final goal, if it reaches the cell where it is located. Like the previous version, once a goal is reached the agent moves to the next one. This goes on until it arrives at the last goal and the run is finished.

Given how the grid system and its coordinate system work, it may happen that two different goals are located in the same cell. It may even happen that two goals in the same cell occur one after the other in the sequence of goal. In this scenario, the model considers both goals to be met. At first, we required the agent to meet each goal at a different timestep. However, because the agent does not have the ability to stay still in the same spot, the agent would have to move to an adjacent cell then move back to the goal cell. In later versions, a new elementary movement will be added to allow the agent to hold its position.

For the moment, the concept of the grid solves the issue of determining whether the agent has arrived at a goal or not which we had in the previous version.

5.2.4 The Model Cerebellum

At this stage, our idea for the model cerebellum was that it would learn to associate the elementary movement used to reach a cell of the grid with the elementary movement executed at that cell. The cerebellum, however, was not implemented in this version of the model and therefore this idea was not put to test until later. In the next version, we put this idea into practice and improve it. We also expand on the idea of the grid.

When the agent arrives at a cell, the model cerebellum would choose a movement based on the one used to reach that specific cell. This allows the model to learn to produce different behaviours based on the context of the action. Depending on the movement being performed or the current goal, the model cerebellum will make different decisions for the same position on the environment.

Besides the introduction of the grid system, we had another idea. In the first version, we had a problem with the method the cerebellum used to choose which movement to execute. This method was based on the strength of the associations established and had the ability of combining elementary movements. Unfortunately, there were problems caused by the fact that the associations could not be weakened and by the interference the elementary movements did to each other.

For the reason above, we decided to use a simpler approach focusing on probability. Basically, the model cerebellum would keep track of the frequency which elementary movement was chosen for a given situation. When the cerebellum was tasked with reproducing the movement, it would choose the elementary movement with the biggest percentage of observations for the current context. In the event of a draw between two or more EMs or in case that not enough observations were made to make a decision a different method would be used. Possible ideas were to ask the model cerebrum to choose the elementary movement for that timestep or to select an EM randomly.

Combining these two ideas means the model would need two pieces of information to determine the current context: the cell of the grid where the agent is currently at and the elementary movement used to

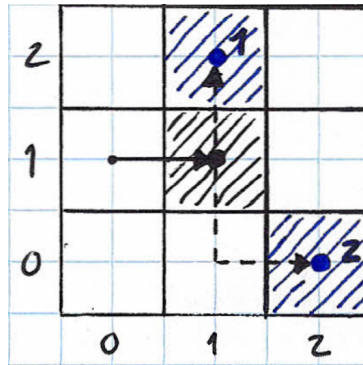


Figure 5.3: This grid displays two movements that interfere with each other. The blue cells represent the goal cell of each movement while the black cell is the current position of the agent. The dashed arrows indicate the movements the model cerebellum learned for each goal. The solid arrow indicates how the agent reached the problem cell.

reach that cell. The cerebellum will observe the elementary movements that are executed by the model cerebrum at a given context. Then, when presented with the same context, it will choose the elementary movement it observed the most.

5.2.5 Problems With This Version

After taking a closer look at the grid concept, a problem becomes apparent. It occurs if the model cerebellum tries to learn two movements with different goal positions that pass through the same cell. Additionally both movements need to enter the cell in question using the same elementary movement. Keep in mind that the current cell and the EM used to reach it are the context information used by the model cerebellum to make a decision. If the described conditions are met then the two movements the model cerebellum is trying to learn will interfere with each other.

Figure 5.3 illustrates the issue. In this case, the model cerebellum has learned two different movements. The first one ends at the cell (1; 2) and the second one at the cell (2; 0). Both movements pass through cell (1; 1) and enter it from cell (0; 1) after performing the EM "right". When the cerebellum tries to recreate movement number one and reaches the cell (1; 1) it might not be able to decide correctly. This is because it will have both an association to move up towards goal one and an association to go down towards goal number two. As explained earlier the model cerebellum is supposed to choose the elementary movement it observed most of the time for a specific context. This means that if movement number two was observed more times than number one then the decision will be to move down even though it is the incorrect choice for goal one.

From this issue it becomes clear that the cerebellum needs to have information about the goal of the movement in order to improve the context information it uses to learn and make decisions.

Another problem with this approach is that it does not consider that the same movement pattern can be performed in different areas of the environment. In other words, the model cerebellum will consider that these are all different movements and will learn about each one separately. It would be much more productive to be able to recognize that all of these movements are the same one and aggregate all the learning done for each.

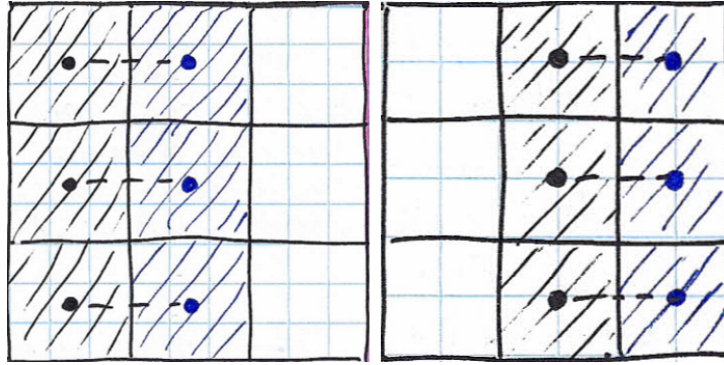


Figure 5.4: These grids display all places where the same movement can be performed. The black cell is the starting position and the blue cell is the end position.

Figure 5.4 displays a simple movement pattern and all possible areas on the environment where it can be performed. In total, we have six possible locations. That means the model cerebellum has the potential to learn six separate versions of the same movement. This is obviously not ideal because if we combined all the observations done into a single version the cerebellum would learn much faster and become much more skilled at the movement than any of the six separate versions were. The existence of separate versions of the same movement has another glaring problem. It is likely that the model cerebellum will become more experienced at the movement in one version than another. It makes no sense for the cerebellum to be skilled at performing the movement at one area of the playing field and then be terrible at another. Of course, there is not much that can go wrong with a movement as simple as the one illustrated. However, the same cannot be said of a longer and more complex movement pattern.

5.3 Third Version of the Model

As mentioned in the previous section, in this version we attempted making the model cerebellum recreate the path taken by the model cerebrum when controlling the agent. We also expand on the idea of using a grid to divide the environment by introducing the concept of relative position to the goal. In short, the model cerebellum does not make associations with specific regions of the playing field. Instead, it takes into consideration the current region where the agent is located relative to the region of the goal. This is useful because equivalent agent and goal configurations may happen in different places in the environment.

Contrary to the first version of the model, in this one it is not possible to combine multiple elementary movements into a single action taken by the agent. This was done simply to facilitate the implementation of the new features we had planned. The capability of combining EMs is reintroduced later.

5.3.1 The Relative Grid

As we explained in the previous section, the concept of the grid had some issues and, therefore, we tried to improve it in this implementation. Instead of having only the grid described before, we decided

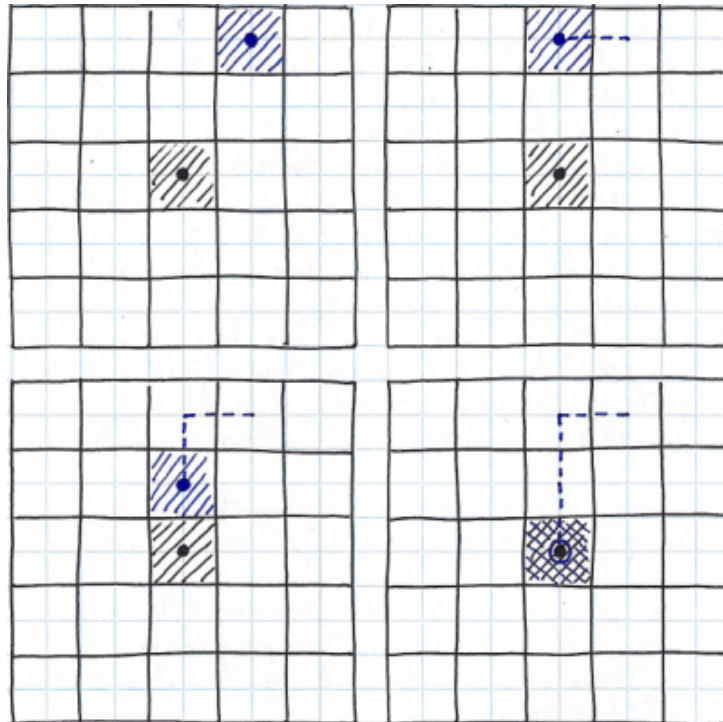


Figure 5.5: The execution of a movement represented on a relative grid. Black is the cell of the agent, blue is the cell of the goal. Starting from the top-left, the elementary movements performed are: right, up and up. On the relative grid, it looks as if the goal is moving left, down and down.

to create a second grid. The model cerebrum will use the first one while the model cerebellum will use the new one. The second grid will give the cerebellum the ability to understand the current goal in the context of the movement and solve the issue of learning multiple versions of the same movement.

We call this second grid the relative grid. This grid shows the position of the current goal in relation to the position of the agent in the simple grid. This means that the agent is always positioned at the center cell of the relative grid making it appear as if the goal is the one moving. Figure 5.5 displays an example of the agent moving towards a goal represented in relative grid.

The first grid will be referred to as the absolute grid as it shows the real regions in the environment where the agent and the goal are located. The relative grid is necessarily larger than the absolute grid. This is because it should be able to represent all agent position and goal position combinations that are possible in the absolute grid. For instance, a three by three absolute grid will have a five by five relative grid. Therefore, the formula to obtain the dimensions of the relative grid is: $\langle absolute_grid_dimension \rangle * 2 - 1$ Note that neither the environment nor the grid and its cells have to be squares. However, we did make our experiments with square fields and grids to keep the distance travelled by each elementary movement consistent.

Figure 5.6 displays various agent position and goal position combinations and how those are represented in the relative grid. Note that the position of the agent in the relative grid is always the center for all four situations. The figure also displays possible paths that the agent can take to reach each of the four goals while travelling the shortest distance possible. Notice that the proposed movements for goals one, two and four all pass through the cell (1; 1) of the absolute grid. If the model cerebellum were

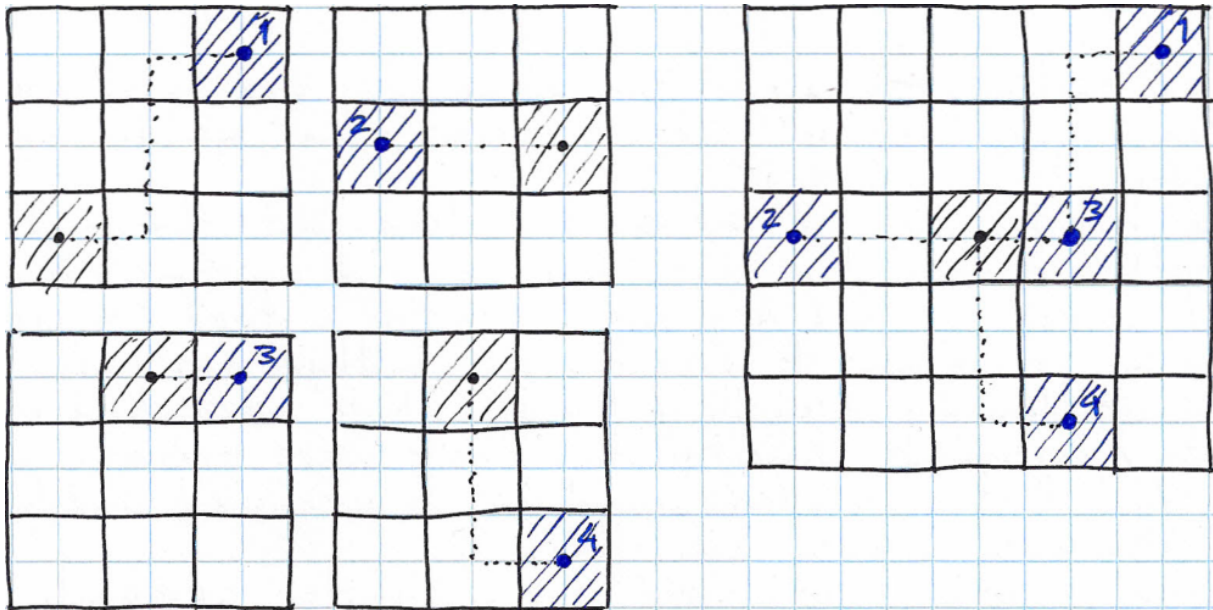


Figure 5.6: On the left, four different movements represented on the absolute grid. On the right, the same four movements represented on the relative grid. Black cells and blue cells represent the starting and end positions, respectively.

to learn these movements using the information provided by the absolute grid, they would interfere with each other. However, if we look at these same movements represented on the relative grid we can see that they would not disturb each other. This feature of the relative gives the model cerebellum the ability to learn multiple movement patterns, which is something we intended from the start. This solves one of the issues we had with the grid concept. Nevertheless, it is important to note that if the agent did not move straight to the goal, instead choosing a more complicated trajectory, this problem could still occur.

The relative grid provides a solution to another issue of the absolute grid as is displayed in figure 5.7. On the two grids on the left, the same movement pattern is represented on different areas of the environment. In the initial concept, the model cerebellum would learn two separate versions of the movement but in the relative grid these two movements are actually represented as the same one. This means that by using the relative grid the model cerebellum will only learn one version of the movement. The cerebellum will be able to recognize that both movements are the same thus solving the issue of fragmenting the learning between multiple versions.

5.3.2 The Agent's Movement

In this implementation, we did not make any changes to the way the agent moves on the environment. It still travels a fixed distance with each elementary movement, moving from one cell to another.

5.3.3 Goals and How to Meet Them

As was explained in the previous subsection the goal positions are represented as cells both in the absolute and relative grids. The model starts by translating all the goal positions into cells of the absolute grid. These are used by the model cerebrum to select the appropriate elementary movement to move

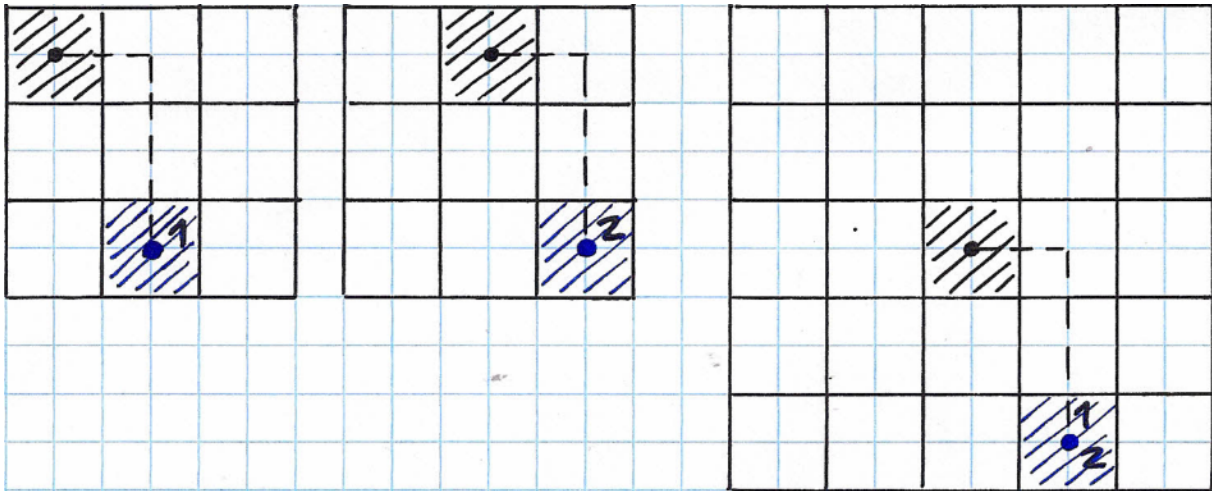


Figure 5.7: On the left, the same movement performed in different places of the environment as represented on an absolute grid. On the right, both versions of the movement have the same representation on the relative grid. Black cells and blue cells represent the starting and end positions, respectively.

the agent in the environment. As for the relative grid, only the current goal is represented and its position is obtained using the current positions of the agent and the goal on the absolute grid.

Like in the previous version, the agent meets a goal when it reaches the cell of the current goal. As the position of the agent is fixed on the relative grid, it will appear as if the goal reaches the cell of the agent.

5.3.4 The Model Cerebrum

The behaviour of the model cerebrum remains the same as the previous versions: a simple algorithm that moves the agent straight towards the current goal. The resulting trajectory covers the shortest Manhattan distance between the start and final positions.

5.3.5 The Model Cerebellum

The behaviour of the model cerebellum is the one described in subsection 5.2.4 but instead it uses the relative grid, which solves some issues as we explained earlier.

The model cerebellum will use as context information the relative position of the current goal and the elementary movement used to reach that configuration. The cerebellum will learn to associate the context information with the elementary movement executed by the cerebrum in that context. When reproducing the movement the cerebellum will choose the elementary movement observed more frequently at a given context.

The model cerebellum learns and performs in separate moments. First, it observes the model cerebrum executing the movement. From this observation, it makes associations and stores them. Then it is prompted to reproduce the movement pattern without assistance from the cerebrum. During this phase, it reacts to the current context based on the associations established earlier. No learning is done in this phase. If at a given moment the model cerebellum is incapable of choosing which elementary movement

to execute it selects one at random. We realise this is neither the best solution nor one that would align with the behaviour of a real cerebellum. This behaviour will be changed in a later version of the model.

5.3.6 How the Model Behaves

For the reasons we mentioned in section 5.2 we decided to separate the execution of the model cerebrum and the model cerebellum in this version. This way we can observe the behaviour of the cerebellum clearly and make adjustments. For the current implementation, there are only two runs to keep things simple.

First, there is a training run where only the model cerebrum controls the agent. The cerebellum observes the elementary movements selected by the cerebrum and learns the movement pattern by making use of the relative grid.

After the training run, the model prompts the model cerebellum to attempt to recreate the movement pattern observed. We call this run the test run. During this phase, the cerebellum acts alone, controlling the agent. As explained in subsection 5.3.5 it will choose the elementary movement observed more frequently for its current context. In case the model cerebellum does not possess information on how to behave in its current context, it will choose an elementary movement at random. Likewise, if there is a draw between two or more elementary movements the cerebellum will select one randomly from the ones in question.

When we actually run the model, the cerebellum is able of reproducing the exact movement performed by the cerebrum. However, this is not a very exciting result as the cerebellum only observes one movement pattern and there is only one training run. We will detail on the issues involved in the following subsection. The result we observe is that the model cerebellum is always one hundred percent certain of the elementary movement to execute at any given context.

5.3.7 Problems With This Version

The main problem of this version is that the model cerebellum only learns one movement pattern which turns out not to be much of a challenge. As there is only one training run, the model cerebellum will not have problems reaching the goal. Because it only observed one movement pattern, it knows the exact elementary movement to perform at each cell.

The fact that the starting position of the agent and the goals are fixed adds to the problem. Even if we performed multiple training runs, the behaviour of the model cerebrum is consistent meaning it would choose the same exact trajectory every run. The model cerebellum would simply observe the same movement repeatedly. This would not produce any uncertainty when the cerebellum had to make the decision on how to move the agent during the test run.

The current behaviour of the model is not very interesting, as the model cerebellum is not learning how to react to varying scenarios. Even when performing the same movement a person does not produce the exact same trajectory every time, especially if it is a movement they are still not familiar with. When a person is beginning to learn a new movement, their execution will be less consistent and

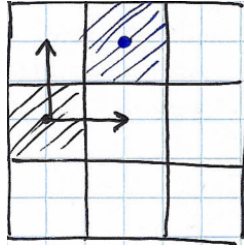


Figure 5.8: An example of multiple choices when selecting a movement. The arrows indicate the two possible elementary movements the model cerebrum can select to move the agent closer to the goal.

there will be some experimentation to find the correct technique.

With the current execution plan of making a single training run followed by a test run we cannot properly assess whether the model cerebellum is capable of learning multiple movements or not. This is important because this was one of the reasons we implemented the relative grid. In order to do a proper assessment we need to make sure that the cerebellum observes multiple movements by adding more training runs. Additionally, each run should be unique so that we can see how it behaves in multiple scenarios.

5.4 Fourth Version of the Model

Starting with this version, all the runs are randomized. This means that the set of goals the agent needs to pass through are chosen at random for each run. This allows us to see how the model cerebellum is able to adapt to different scenarios.

The behaviour of the model cerebrum was adjusted in this version in order to introduce more variation to the movements it performs. Additionally, multiple training runs are performed in order to teach multiple movement patterns to the model cerebellum. The conditions to meet a goal remain the same and the agent's movement is also unchanged.

5.4.1 Goals and How to Meet Them

The methods used to represent goals on the model are the same as the previous version. The conditions required to meet a goal also remain unchanged. The only difference is that new goals are generated for every run so that the model is exposed to new situations. This change will make it easier to find unexpected and unintended behaviour of the model cerebellum.

5.4.2 The Model Cerebrum

We wanted to make the behaviour of the model cerebrum less predictable in this version. The reason for doing so was to study how the model cerebellum would behave in scenarios where it is not fully confident on which elementary movement to perform. Another reason was to give the cerebellum a method to combine elementary movements. The details on how this is achieved are described in subsection 5.4.3.

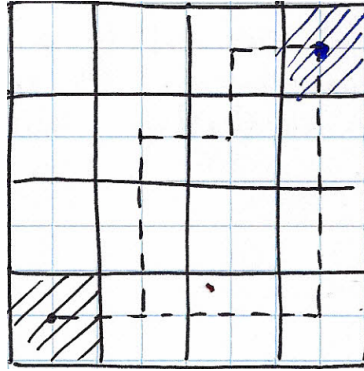


Figure 5.9: An example of two possible trajectories that take the least number of steps to reach the proposed goal.

Often, when moving the agent there are situations where more than one elementary movement can be used to move it closer to the current goal. Figure 5.8 shows an example of such a situation. In its current position, the agent can either choose to move up or to the right because both will shorten the distance to the goal. In previous implementations of the model cerebrum, it would give priority to one of the possible choices of movement. For this implementation, we decided to make the choice randomly.

This decision does not affect the ability of the model cerebrum to guide the agent towards the goal. It means, however, that the resulting trajectory can be very different each time the same start and end positions combination is presented. Figure 5.9 displays two different movement patterns that can be used by the model cerebrum to reach the same goal.

5.4.3 The Model Cerebellum

There is not much difference between this version and the previous one. The model cerebellum will still choose elementary movements at random in the situations described in subsection 5.3.5. There is a new concept in this implementation, which will help improve the cerebellum's behaviour in the next version. We introduced a confidence threshold to the decision process. This means that to choose an elementary movement the cerebellum must be sufficiently confident in it. In other words, it must have observed that elementary movement for a significant portion of the occurrences of that specific context.

A consequence of this change is that it will be easier for multiple elementary movements to be eligible to be performed. When multiple EMs beat the confidence threshold, the cerebellum chooses one randomly. If none reach the threshold then one is chosen at random from all available elementary movements, which means the agent may move in any direction.

5.4.4 How the Model Behaves

In this version, we perform multiple training runs. As stated in subsection 5.4.1, new goals are generated for every run. This means that the model cerebellum will be exposed to multiple movement patterns.

As for the test runs, there is only one but it is also randomized. Because of this fact, it may happen that the model cerebellum has never observed the start and goal combination presented on the test run.

Because of this configuration, it becomes important to have some information of how much the

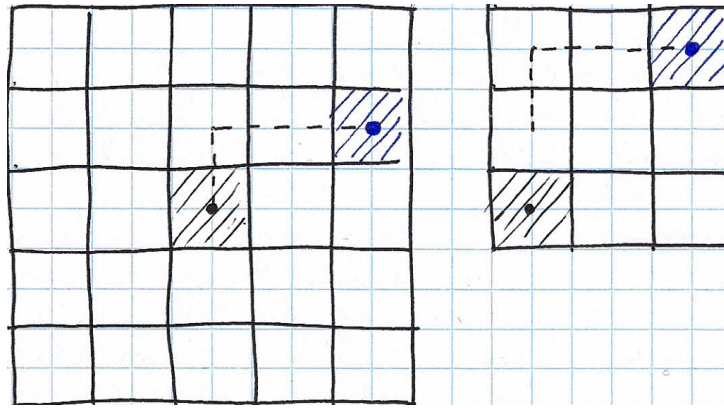


Figure 5.10: The relative grid on the left displays a movement pattern the model cerebellum has learned. On the absolute grid on the right, if the agent moves up it will arrive at a position where the cerebellum can apply the pattern learned.

model cerebellum has learned and how much it struggles during the test run. With that in mind, we started tracking some statistics about the decisions made by the cerebellum. This version keeps track of the number of times it was able to make a decision for a timestep, the times it could not because the threshold was not met and the times it had no information to make a decision.

Tracking statistics about the model helps us understand how the number of training runs and the number of cells of the absolute grid influence the quality of the learning of the cerebellum.

One thing that is important to note is that whenever the model cerebellum gets lost and starts moving randomly it may arrive at a situation it has seen before. Figure 5.10 shows an example of how this could happen. The model cerebellum has never observed the elementary movement it should select for the agent's current position relative to the goal. This means it will have to choose an elementary movement at random. If it ends up selecting to move up it will arrive at a familiar context and will be able to find a way towards the goal on its own. If it ends up moving to the right, however, the model cerebellum will remain lost and moving randomly.

It is also noteworthy that movement patterns have sub movement patterns. Figure 5.11 illustrates this property. The first relative grid representation on the top-left shows the movement pattern learned by the model cerebellum. The other three are sub-movement patterns that are also learned when the first one is observed.

5.4.5 Problems With This Version

On the topic of tracking statistics, it becomes clear that only performing one test run is not enough to evaluate the performance of the model cerebellum. That is because the scenario presented during the test run may be either one the cerebellum has already observed or a completely new one. The statistics can either show a great performance by the cerebellum or show that it got lost. In order to improve the quality of the statistics the model will need to perform multiple test runs to gather enough information.

The idea of the confidence threshold could also be improved. Given that multiple elementary movements may beat the threshold, the model cerebellum could execute them in the same timestep. In the situation illustrated in figure 5.12, the model cerebellum would observe the model cerebrum moving the

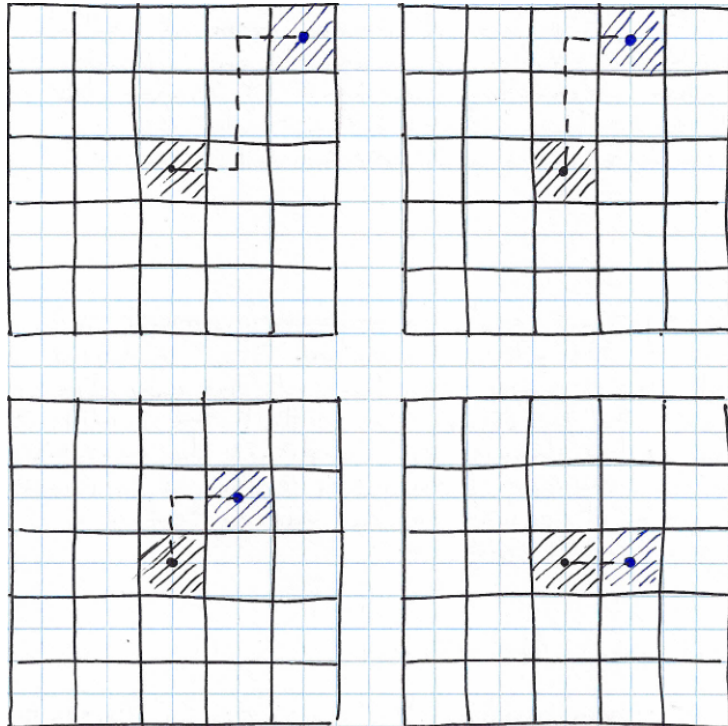


Figure 5.11: An example of a movement pattern and the sub-patterns it includes represented on the relative grid.

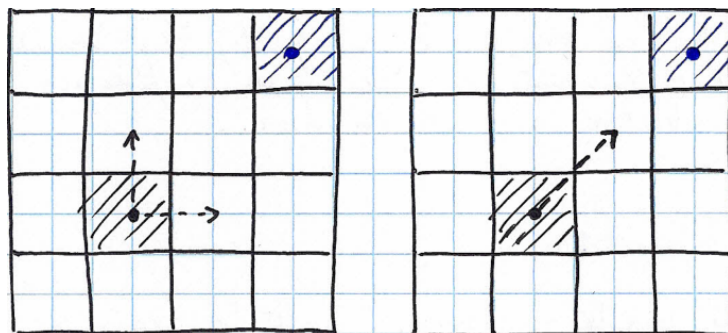


Figure 5.12: Combining elementary movements to produce new movements. The elementary movements up and right displayed on the left grid can be combined to produce the diagonal movement displayed on the right grid.

agent either up or to the right. This means that the cerebellum can build confidence in both elementary movements and, therefore, combine them. This would result in the agent moving diagonally which would achieve a shorter path towards the goal. A problem would occur if the model cerebellum attempted to execute opposite elementary movements as these would nullify each other. However, this does not happen as the model cerebrum always chooses a direct path towards the goal.

Another aspect that should be improved is how the model behaves when the model cerebellum is not capable of selecting an elementary movement. Choosing a movement randomly does not appear to be a realistic approach. Additionally, by observing the model running we saw that the model cerebellum could get lost for a very long period if the number of training runs is too low when compared to the size of the absolute grid. A better alternative would be to ask the model cerebrum to select the appropriate elementary movement for that moment while the model cerebellum observes and updates its learning. After that moment, the cerebellum would resume controlling the agent unless it was still lost and unable to make a decision.

5.5 Fifth Version of the Model

This version is very similar to the previous one but it implements some of the simple improvements mentioned in subsection 5.4.5. As such, the model cerebellum is now able to combine elementary movements allowing it to move the agent diagonally on the environment. The model also performs multiple test runs in order to gather more accurate data on the performance of the cerebellum.

Besides these changes, we also started working on some ways to visualize the path taken by the agent on the field. Other concepts like the grid and the behaviour of the model cerebrum have not been modified.

5.5.1 The Model Cerebellum

In this implementation, the model cerebellum continues to use a confidence threshold in order to choose the elementary movement the agent should execute. However all elementary movements that beat the threshold are executed instead of only one. This gives the cerebellum the ability to combine elementary movements. Given the way the model cerebrum chooses movements the result of this change is that the model cerebellum will be able to move the agent diagonally. Contrary to the first version of the model, it is not possible to combine an elementary movement with itself in order to increase the distance travelled. As before, if no elementary movement reaches the confidence threshold the model cerebellum will choose one at random.

Another change was made which is related to the context information used by the model cerebellum. We found out that the model cerebellum was capable of choosing the correct elementary movements and reach the goal even without considering which elementary movement was used to reach a cell of the grid. Therefore, we decided to remove that piece of information from the decision making process to simplify the model.

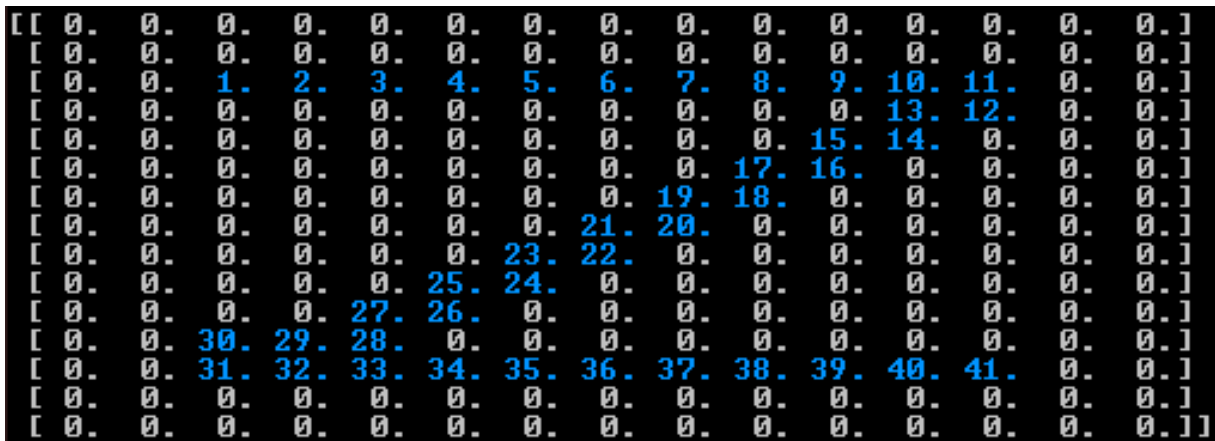


Figure 5.13: A screenshot of the simple graphical representation used to visualize the trajectory taken by the agent when controlled by the model cerebellum.

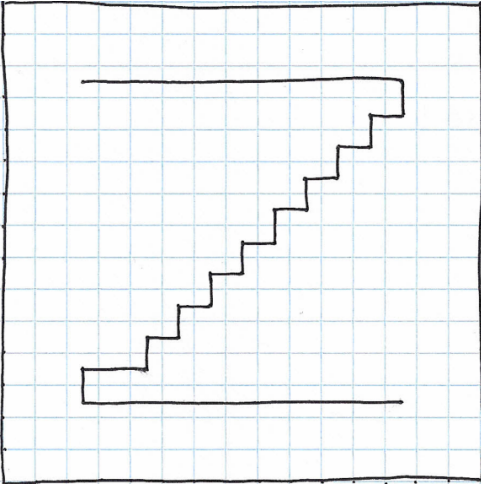


Figure 5.14: A movement pattern that resembles the letter Z, represented on the absolute grid.

5.5.2 How the Model Behaves

Like the previous version, the model performs multiple training runs in order to teach various movement patterns to the model cerebellum. However, in this version the model also performs multiple test runs. This allows us to gather statistical data on how the model cerebellum is performing. This gives us an idea of how much it has learned with the amount of training runs provided.

During the development of this version, we also started working on a way to represent visually the environment and the path taken by the agent. As a start, we made the model print a matrix filled with numbers representing the absolute grid. Positions with a zero represent a cell that was not visited by the agent. Other numbers represent the movement performed starting at step one and incrementing by one for each subsequent step. Figure 5.13 shows an example of a movement pattern performed on a fifteen by fifteen grid.

In this version, we also realized that the model cerebellum should be able to learn any movement pattern the model cerebrum shows it even if it does not move the agent straight to the goal. In order to test this out we made a special scenario where the model cerebrum produces a trajectory that resembles a letter and the model cerebellum was capable of reproducing it. Figure 5.14 displays the movement

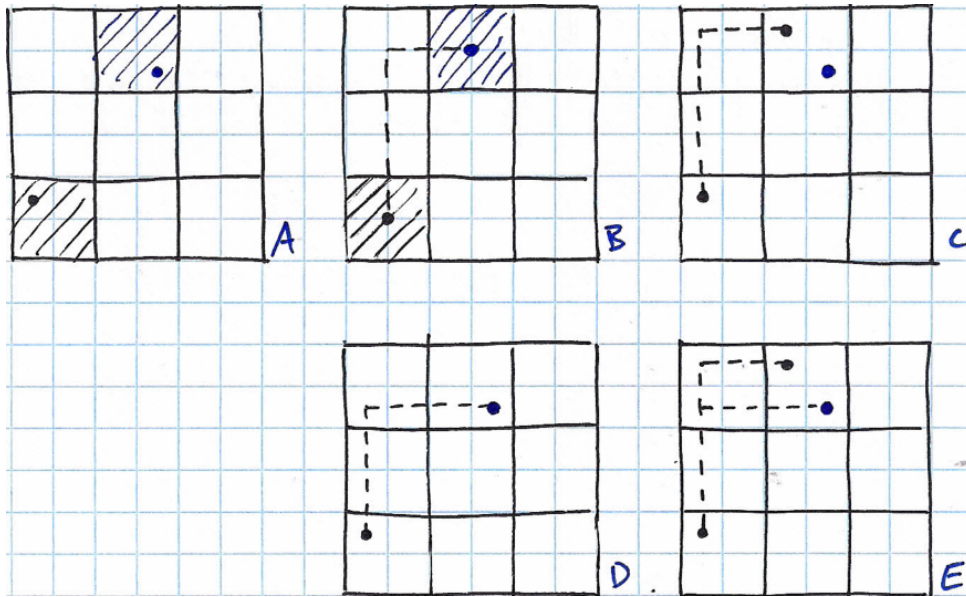


Figure 5.15: An example of how using the grid may produce a movement pattern that does not properly take the agent to the goal.

pattern used in this scenario, which resembles the letter Z. This is the same scenario used for the example of the graphical representation in figure 5.13.

We mentioned in the first version that the model uses a list of goal and that the goals before the final one were supposed to be treated as checkpoints of the movement pattern. However, the model treats each goal as being part of a separate movement pattern. When the agent reaches a goal, the current movement ends and a new one begins at that position.

5.5.3 Problems With This Version

In this version, we became aware of an important problem of the grid concept. This issue is that the movement trajectory produced on the grid may not actually connect the real starting and goal positions on the environment. Remember that the model starts with very specific positions for the agent and the goal, which are then translated into cells of a grid. Each cell encompasses a region of the environment and many positions will be translated into the same cell.

Figure 5.15 illustrates the problem. Grid A displays the original positions and the cells they were translated into. Grid B shows the movement pattern learned using those cells. Then, grid C shows what would happen if we were to apply the movement pattern learned to the real starting position. We can see that despite reaching the cell where the goal is located it does not reach the goal itself. Grid D presents a similar movement that would reach the real position of the goal and grid E makes a comparison between the two movements.

Changes need to be made to the model so that the cerebellum can properly guide the agent to the real goal position.

5.6 Sixth Version of the Model

In this version, we tried to solve the problem described in subsection 5.5.3. In order to do so we started by adjusting the conditions the model uses to recognize that the agent has reached the goal position. In addition to that, we implemented a method to manipulate the number of cells used to divide the environment. We call this method increasing the resolution of the grid. By dividing the field further, the agent will be able to make finer adjustments to its trajectory, which should allow it to move closer to the goal.

In addition to these changes, we also implemented other methods the model cerebellum can use to decide the agent's next movement. We also made it so that the model cerebellum no longer chooses randomly when unsure. Instead, it asks the model cerebrum to intervene.

5.6.1 Goals and How to Meet Them

We determined in the previous version that moving from one cell to another does not necessarily produce a trajectory that moves the agent to the real position of the goal. However, we also find that it is challenging to give the model cerebellum the ability to make the fine adjustments necessary to reach the exact goal position. The cerebellum needs to observe the model cerebrum performing these small movements and have a way to store this information in order to learn them.

Our solution to this problem started by making a compromise. Instead of having the goal position be a single point, we made it a small circle. If the agent reaches this circle then the model considers the goal was reached. The radius of the goal was set so that accuracy was not sacrificed excessively.

5.6.2 The Agent's Movement

The agent moves a fixed distance with each elementary movement but this distance is now independent of the grid being used. The agent no longer moves from the center of one cell to the center of an adjacent one. With the new ability to change the resolution of the grid, that would mean that the distance travelled would decrease as the resolution increased.

For the distance, we chose one that made sure the agent would not overshoot the goal and make it impossible to get inside the goal circle. This implies that the agent moves a short distance with each elementary movement. On lower resolutions, this means that after taking a step the goal may remain at the same cell of the relative grid. This does not interfere with ability of the model cerebrum to guide the agent towards the goal.

5.6.3 The Relative Grid

The distance the agent travels is now independent from the distance between the centers of adjacent cells. This means we needed to make some changes to the relative grid.

The relative grid moves with the agent so that it is always at the very center of the grid, not just in the center cell. The absolute grid cannot be used to make the translation of the position of the agent

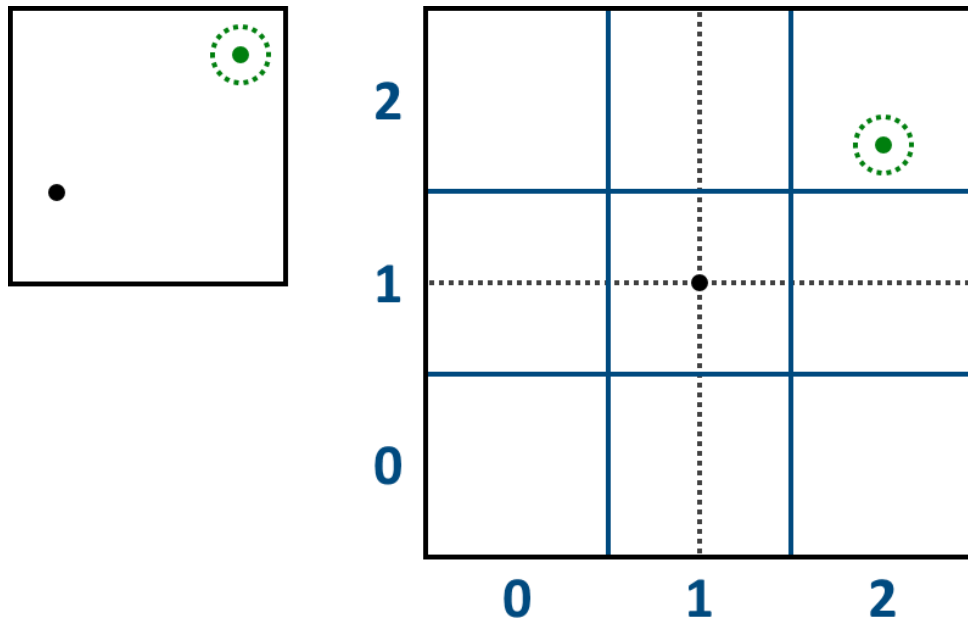


Figure 5.16: How the relative field and grid are generated. On the left is an illustration of the environment with the agent represented with a black dot and the goal and goal circle represented in green. On the right we have the relative field (in black) obtained from the environment on the left. Then, the relative grid (in blue) is obtained by dividing it into 9 cells. The positions of the agent and the goal are translated so that the agent is in the center. With this setup, the goal is currently in cell (2;2) of the relative grid.

relative to the goal because of the problem described in subsection 5.5.3. Therefore, we need to use a different approach to determine the initial size of the relative grid and the size of its cells, and to make the translation.

To do this we created a relative field, which is a transformation of the environment where the agent is always at the center and the goal moves relative to it, similar to the relative grid. Then, we divide this relative field as desired in order to produce the relative grid. For our experiments, we started with a three by three relative grid.

To obtain the position of the goal in the relative grid we start by determining its position in the relative field then translate it to the corresponding cell. Figure 5.16 illustrates how the relative field and grid are obtained and how the goal position is translated.

Because of this change, the absolute grid is no longer necessary.

5.6.4 Resolution Update Strategies

In order to exploit the new properties of the goal position and the agent's movement, we gave the model the ability to adjust the resolution of the grids used. In other words, the ability to divide the environment further, into smaller regions.

When increasing the resolution we start by updating the relative grid. The model divides each cell of the current grid into nine cells. Dividing into four cells would create a smaller increase in resolution but that means there would not be a cell at the center of the grid to be used as the agent's reference point. Figure 5.17 illustrates the issue.

In this version, we start with a low resolution and the agent moves towards the goal using the current

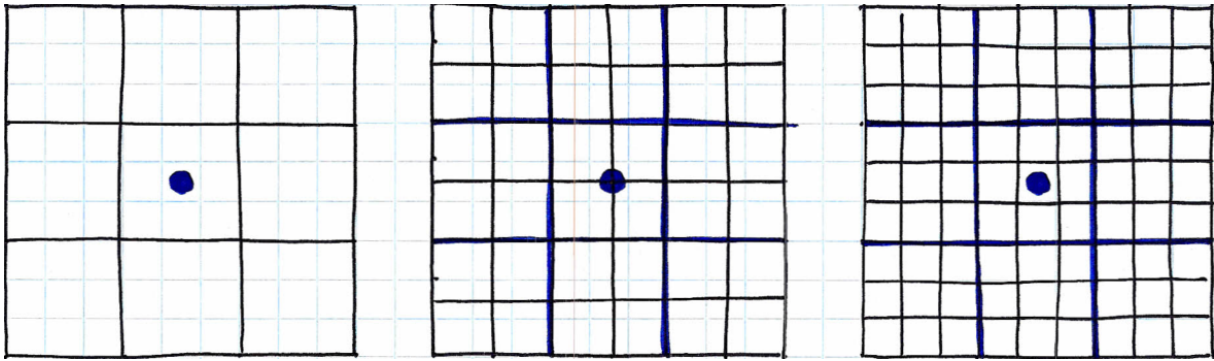


Figure 5.17: Different methods of dividing the relative grid to produce a new grid of higher resolution. The blue dot marks the reference point for the agent. The grid on the left represents the original resolution. The one in the middle shows what happens by increasing the resolution by dividing each cell into four. It does not keep a cell in the center for the reference point. The grid on the right displays the method of division used in this version.

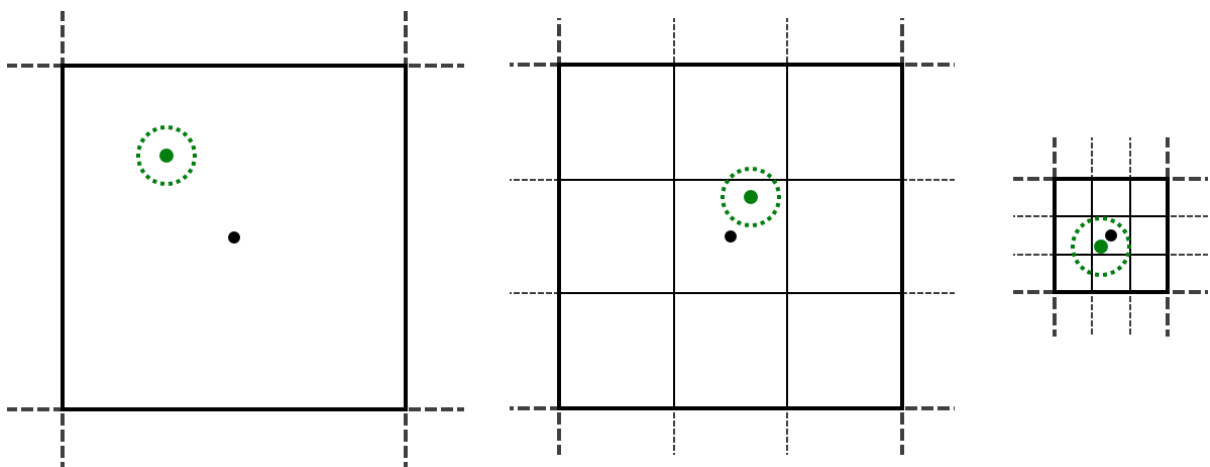


Figure 5.18: Comparison of different resolutions using the same goal circle radius. The figure shows the center portion of three relative grids of different resolutions with a goal at the center cell. The three goals represented are not the same but their circles have the same radius. As the resolution increases, the goal circle occupies a bigger area of the center cell.

goal cell in the relative grid as its reference point. Remember that in the relative grid the goal appears to be moving instead of the agent, which always stays at the center (see figure 5.5). This means the goal cell will change as the agent moves. Once the goal cell becomes the center cell of the relative grid, the model checks if the agent is inside the goal circle. If it is not, the resolution is increased and the agent will move towards the new goal cell. This sequence of events repeats until the agent is able to reach the goal circle, at which point the movement ends.

Figure 5.18 helps us explain how increasing the resolution allows the agent to move closer to the goal. It displays the center of three relative grids of different resolutions with a goal circle with the same radius. The higher the resolution, the more area of the center cell the goal circle occupies. This means the agent is more likely to enter the goal circle when the goal cell becomes the center cell of the relative grid. The further we increase the resolution the smaller the cells become and eventually it will be guaranteed that having the goal arrive at the center cell places the agent inside the goal circle.

Additionally, note that it is possible for the agent to reach the goal circle without having the goal at the center cell of the relative grid. In this scenario, the model considers that the agent has reached the

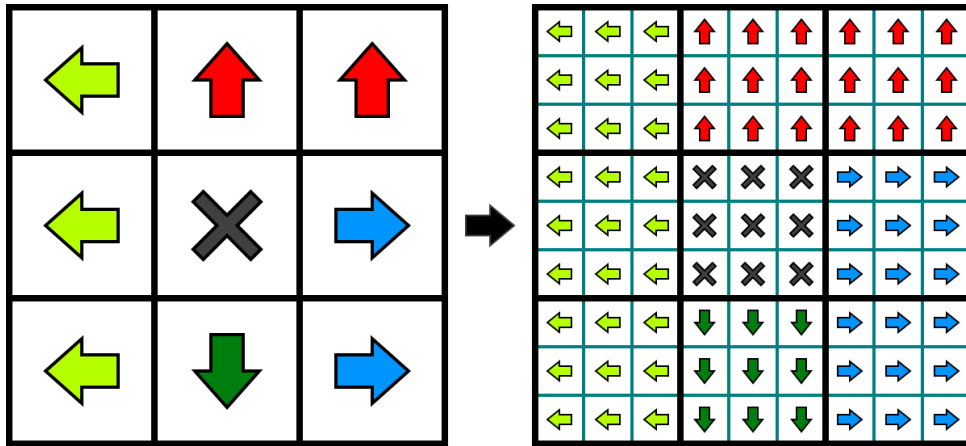


Figure 5.19: The zoom resolution update strategy.

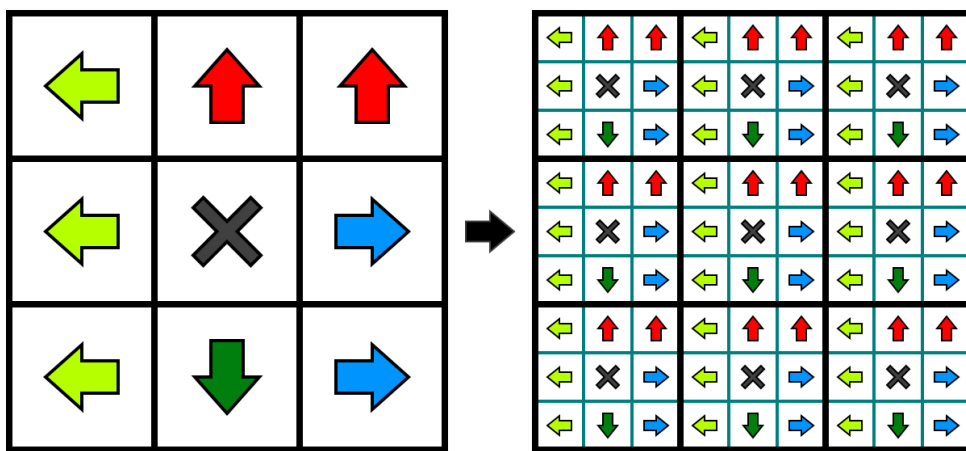


Figure 5.20: The tile resolution update strategy.

goal and the run ends.

The last issue that needs to be addressed is how we should transfer the knowledge acquired with one resolution to the new one. For this task, we thought of two strategies.

The first strategy consists in copying the observations done in one cell to all the new cells it was divided into. Figure 5.19 demonstrates this process. Note that the grids represented are relative grids, which are the ones the model cerebellum uses to learn and make decisions. The elementary movements represented are an example of what could be the most observed elementary movement when the goal is at that cell relative to the agent. The cells marked with an X represent positions where no movement was learned. We call this strategy the zoom strategy.

The second strategy creates a tiling pattern using the lower resolution and its knowledge as a tile. This is illustrated in figure 5.20. We call this the tile strategy.

It is important to note that not only the most observed elementary movement of each cell is transferred to the new resolution. All observations are transferred. Figures 5.19 and 5.20 only show the most observed movement to make the explanation easier to follow.

5.6.5 The Model Cerebrum

The model cerebrum will guide the agent using the goal cell on the relative grid. If the agent is not inside the goal circle once the goal cell becomes the center cell of the relative grid then the resolution is increased and the cerebrum will guide the agent towards the new goal cell. This process continues until the agent arrives at the goal circle.

5.6.6 The Model Cerebellum

The model cerebellum continues to observe the model cerebrum quietly during the training run. As the cerebrum increases the resolution of the grid, the cerebellum uses the new relative grid to store its observations. However, the older resolutions are still preserved.

During the test runs, the cerebellum no longer chooses an elementary movement at random when it is unable to make a decision. Instead, it relies on the cerebrum, asking it to decide. It then observes the movement performed and updates its knowledge.

The model cerebellum starts the test runs at the highest resolution achieved during the training runs. If at some point it is incapable of performing a decision, it switches to a lower resolution. It keeps lowering the resolution until it can make a decision. The idea here is that higher resolutions will have information that is more specific and may not have enough knowledge to act in every scenario. On the other hand, lower resolutions will have more general knowledge, which can be applied to a wider range of situations.

If, at some point, the model cerebellum switches to the lowest resolution and is still unable to select an elementary movement, it will ask the model cerebrum to make the choice.

When the goal cell becomes the center cell of the relative grid but the agent is still outside the goal circle the cerebellum will increase the resolution. This repeats until the agent reaches the goal circle.

Besides the resolution strategies, we also introduced a set of strategies that the model cerebellum can use to choose the elementary movement to perform at a given moment. The strategy can be selected when configuring the model. This allows us to analyze different configurations of the model.

One of the selection strategies is the threshold strategy, which was used in the previous version. All elementary movements that reach the confidence threshold are executed. Another strategy is the highest percentage strategy. In this strategy, the most observed elementary movement for a given context is executed. The last strategy is the weighted random strategy. The model cerebellum selects an elementary movement at random but the probability of a movement being chosen depends on how frequently it was observed. Elementary movements observed more often are more likely to be selected.

5.6.7 How the Model Behaves

We tested all possible resolution strategy and selection strategy combinations. However, we found that all of them performed poorly. The model cerebellum would fail to reach the goal in most of the test runs. We suspect this is because the conditions for the cerebellum to switch resolutions during the test run were poorly thought out.

5.6.8 Problems With This Version

As stated in the previous subsection the act of switching resolution during the test runs seems to cause problems for the model cerebellum.

One of the problems is that whenever the model cerebrum has to intervene only the lowest resolution is being updated. The resolutions where the model cerebellum was unable to make a decision remain unchanged. This means the cerebellum may still not be able to deal with the situation that gave it problems and may have to rely on the cerebrum again.

We decided to simplify the next version by removing the ability to decrease the resolution and focus on analyzing the performance of the resolution update strategies.

5.7 Seventh Version of the Model

In this version, the ability to decrease the resolution was removed from the model cerebellum, as it seemed to cause problems. These issues prevented the cerebellum from successfully guiding the agent towards the goal.

Besides this change, there is not much difference between this version and the previous one.

5.7.1 The Model Cerebellum

Since the model cerebellum is not able to decrease the resolution it does not store the lower resolutions whenever the resolution is increased. The new resolution simply substitutes the previous one.

Whenever the cerebellum is unable to make a decision with the current resolution, it asks the cerebrum to make the choice. The cerebellum then updates the relative grid with the elementary movement observed.

When the goal cell becomes the center cell of the relative grid but the agent is still outside the goal circle the model cerebellum increases the resolution. The cerebellum then continues to guide the agent with assistance from the cerebrum, if necessary. This process repeats until the agent arrives at the goal circle.

If the agent reaches the goal circle before the goal arrives at the center cell of the grid, the run ends and the model still considers it a success. Note that part of the goal circle may be on one cell and part on another one. The goal cell is determined based on the center point of the goal circle.

5.7.2 Problems With This Version

We made some simple experiments to evaluate the performance of different combinations of resolution update strategies and choice strategies. These experiments and the respective results are described in chapter 6.

While the zoom strategy performs quite well, the tile strategy seems to have some problems as it fails frequently. We analyzed the paths taken by this strategy in more detail and found that the agent would

eventually get stuck walking back and forth between two positions.

The cause of this problem is probably in the contradictory elementary movements that are placed adjacent to each other whenever the resolution is increased with this strategy. Figure 5.20, introduced in section 5.6, shows how the new resolution might look like right after the increase. The arrows represent the most observed elementary movements for each cell. The darkened lines represent the cells of the lower resolution. As we can see, at these borders the elementary movements mirror each other. With this relative grid, it is not possible to create paths from every cell that lead to the center cell, where the agent is positioned. Unless the model cerebellum observes more movements from the model cerebrum that correct these contradictions, the agent will get stuck whenever the goal is at those borders.

This also makes clear why the weighted random strategy outperforms the other selection strategies. Whenever the goal is located at those borders relative to the agent, the model cerebellum might randomly choose another elementary movement, of the ones observed, that might get the agent unstuck.

We found that sometimes the model cerebellum also gets stuck when using the zoom resolution strategy. However, the reason for it happening was not clear at this point.

5.8 Eighth Version of the Model

In this version, there are few changes to the behaviour of the model. The distance travelled by the agent is now dependent on the current resolution. We also simplified some aspects of the model. However, the most relevant change is the addition of a proper graphical representation for the model, which displays the path selected by the model cerebellum.

5.8.1 The Agent's Movement

The agent's movement is tied to the current grid resolution. The distance travelled with each elementary movement is the distance between the center points of two adjacent cells. As the resolution increases, the distance travelled becomes smaller allowing for finer adjustments to the trajectory.

5.8.2 The Graphical Representation

In this implementation, we concluded the implementation of a graphical representation for the model. If instructed to, at the end of a test run the model displays an image of the path taken by the agent. The image shows the starting position of the agent, the position of the goal and its goal circle and a line showing the trajectory of the movement performed. The line is broken into multiple segments using dots, which represent the position of the agent at the end of each step. Additionally, the model keeps track of which steps required assistance from the model cerebrum. The segments and dots corresponding to these steps are coloured red while the remaining ones are coloured blue. The starting position is also coloured red. The goal circle and the final position of the agent are coloured green. The rest of the environment is black and the grey dots represent the possible locations for the start and goal positions. Figure 5.21 shows an example of an image produced by the model.

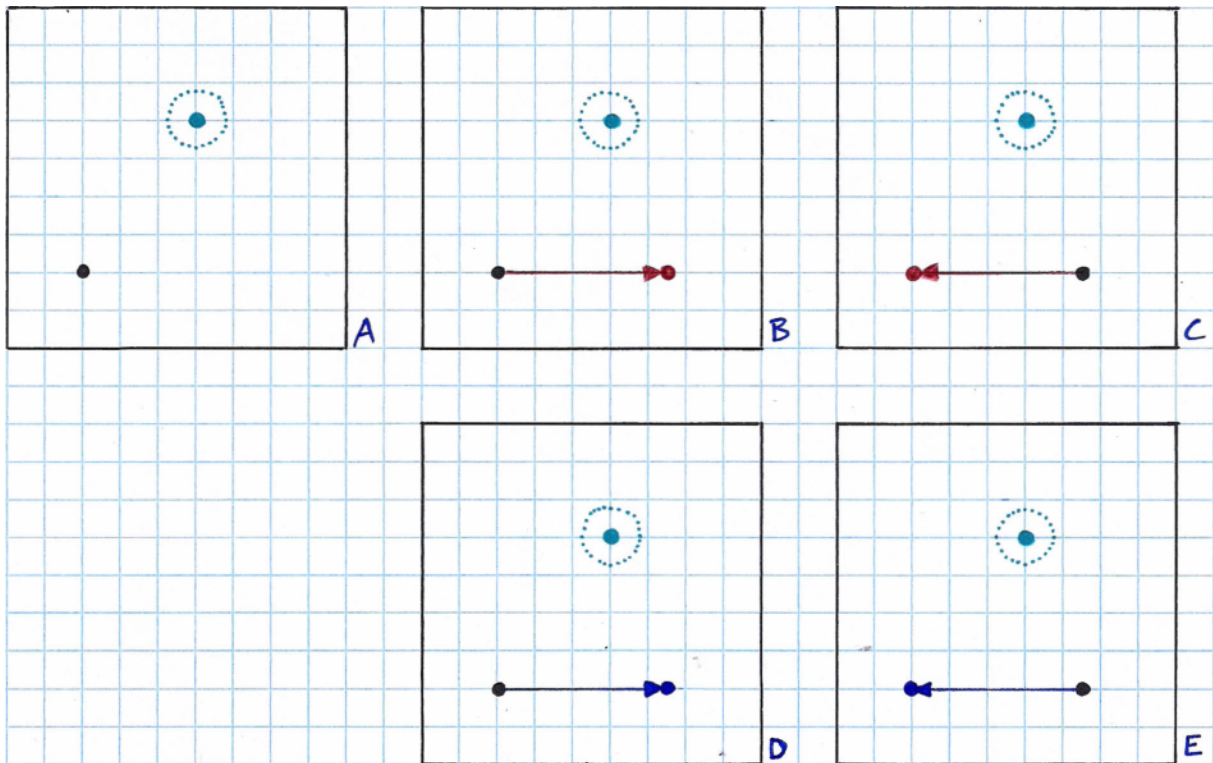


Figure 5.22: An illustration of the looping problem. Image A shows the starting scenario with the goal and goal circle at the top and the starting position at the bottom left. In images B and C the model cerebrum chose the movements and in images D and E the model cerebellum reproduces the movements observed, entering a loop.

though it was only observed once.

The following scenario shows how this property is problematic. In figure 5.22A, the model cerebellum is presented with a completely unfamiliar context and needs to ask the model cerebrum for assistance. There are two possible choices to move closer to the goal: up or right. It randomly chooses to move right as shown in figure 5.22B and the cerebellum learns this association. The new position of the agent is also part of a new context so the model cerebrum is requested again. This time moving either left or right can bring the agent closer to the goal. However, note that the model cerebrum does not consider the distance travelled by each step and is unable to realize that moving to the left will actually increase the distance to the goal. It randomly chooses to go left, moving the agent back to the starting position in figure 5.22C. This time the model cerebellum recognizes the context and moves the agent to the right as learned. This results in the same position observed in figure 5.22B. The cerebellum learned to go left in this context but, in doing so, it moves the agent to the starting position again. It is now stuck in a loop going back and forth these two positions.

This problem happens because the model cerebrum does not judge distances properly and because the model cerebellum establishes associations hastily. The model cerebellum creates an association between a context and an elementary movement with only one observation. It should make more observations in order to give more strength to the association.

5.9 Ninth Version of the Model

In this version, we tried to solve the problem described in subsection 5.8.4. In order to do so we developed a new movement selection strategy based on strengthening and weakening associations using points.

5.9.1 The Model Cerebellum

In order to implement the new strategy some changes were made to the model cerebellum. In this version, the cerebellum does not keep track of how frequently an elementary movement was observed for a given context. Instead, it keeps a score for each elementary movement in each context.

Whenever the model cerebellum observes an elementary movement, it adds a set number of points to its score in that context. All the other movements, which were not observed, will lose a set amount of points for that context.

All elementary movements start with zero points for every context. It is not possible for a movement to have a score below zero points. There is also a maximum amount of points that can be accumulated. Once the maximum score is reached, no more points will be awarded for further observations of the elementary movement in that context. However, all other movements will still lose points for not being selected by the model cerebrum.

5.9.2 The Point Threshold Strategy

This strategy selects elementary movements based on the scores attributed by the model cerebellum while observing the model cerebrum controlling the agent.

For a given context, all elementary movements that beat the defined point threshold will be selected. If no movement is able to beat the target score, the model cerebrum will be asked to intervene. Then, based on the movement chosen the scores will be updated for that context.

Given that only a small amount of points is added for each observation, the model cerebellum is unable to select an elementary movement that it only saw once. This ensures that the cerebellum builds some confidence on an elementary movement before trusting its association with a given context.

It is also possible for multiple movements to beat the point threshold and to be executed together. Therefore, this strategy retains the ability to move the agent diagonally. However, the point threshold, the maximum score, the amount of points added and the amount removed need to be chosen carefully to allow this. This is because whenever a movement earns points the others lose points, which makes it difficult for multiple elementary movements to beat the point threshold.

5.9.3 Problems With This Version

The new point threshold strategy requires much more training than the other strategies in order to achieve a low indecision rate. This means that the model cerebellum must rely on the model cerebrum

for longer and make many more observations. The performance of this strategy and how it compares to others is described in the next chapter in section 6.3.

As an improvement, the model cerebellum should be able to grasp a general idea of the complete movement pattern in the first few observations and assist the model cerebrum thereafter. As the cerebellum becomes more and more confident in that movement pattern it should take the predominant role in moving the agent, only relying on the cerebrum when necessary.

Chapter 6

Experiments

6.1 Seventh Version of the Model

6.1.1 Setup

We tested all six possible combinations of resolution update strategies and choice strategies. The environment used was a field of 50 by 50 units. All agent and goal positions were generated randomly and had 2 decimal points of precision. The agent had a speed of 0.5 units and the goal had a radius of 2 units. Each set of training runs started with a three by three relative grid.

For each configuration, we did 1000 sequences of 300 training runs followed by 20 test runs and then calculated the success rate. The reason for this setup is that in each set of training runs the model cerebellum will be exposed to different movement patterns and its performance during the test runs might not be the same. By executing multiple sequences of training followed by testing, we get a better idea of the actual performance of the model cerebellum with the selected strategy configuration.

Additionally, to prevent the model cerebellum from getting lost for an indefinite amount of time we set the test runs to fail after 500 steps were performed.

6.1.2 Results

The results are presented in table 6.1. For the zoom resolution strategy, all choice strategies performed fairly the same. All three managed to succeed in reaching the goal almost every time with a success

Configuration	Success Rate (%)
Zoom + Weighted Random	99.68
Zoom + Highest Percentage	98.23
Zoom + Confidence Threshold	99.17
Tile + Weighted Random	78.40
Tile + Highest Percentage	53.08
Tile + Confidence Threshold	58.98

Table 6.1: Comparison of the success rate of the different combinations of resolution update strategies and choice strategies.

rate above 98%. The weighted random strategy achieved the best result with a 99.68% success rate.

The configurations using the tile resolution strategy were not as successful. The highest percentage strategy and the threshold strategy succeeded in less than 60% of the runs. The weighted random strategy was the best of the three with a success rate of 78.40%, which is still significantly lower than the results of the configurations using the zoom strategy.

6.2 Eighth Version of the Model

We noticed that if we run a set of tests and then run another set of tests with the same configuration the results may differ. This happens because the start and goal positions for the training and test runs are chosen randomly. Therefore, the model cerebellum will not be trained in the same movement patterns every set nor will it be asked to perform the same movement patterns. To compensate for this we perform multiple sets of testing and then average the results of the test runs. This gives us a more accurate idea of the performance of a certain model configuration.

We tested how varying certain variables of the model affect the performance of the model cerebellum. This is done in addition to testing the different combinations of resolution update strategies and movement selection strategies. We varied three different variables: the dimensions of the playing field (the environment), the number of training runs and the confidence threshold. The confidence threshold is only used by the threshold selection strategy so only this strategy is tested for this variable in combination with the resolution update strategies.

We set a maximum number of steps to finish a test run. If the cerebellum fails to reach the goal within the number of steps allowed, the run is counted as a failure. In theory, the cerebellum should be able to finish every run because the cerebrum will intervene whenever the cerebellum is unable to make a decision. However, the model cerebellum may get lost or stuck because of conflicting movement patterns, mistakes in the implementation or problems with the strategies themselves. The limit of steps is set so the model cerebellum can reach the goal comfortably even if it deviates from the direct path.

To evaluate the performance of the model cerebellum we selected three measures. They were the average indecision rate, the success rate and the distance deviation percentage. All of these are calculated for each stage for all three tested variables.

The indecision rate indicates the percentage of steps made with assistance of the model cerebrum for the steps taken across all test runs in a test set. We measure the indecision rate for each test set and then obtain the average indecision rate for the test stage.

The success rate measures the amount of successful test runs compared to the total test runs for a test stage.

The distance deviation percentage indicates how much the distance travelled by the agent deviates from the expected distance. The expected distance is the Manhattan distance that connects the start and goal positions. The performed movement may deviate from this expectation because the model cerebellum may become lost or due to the execution of diagonal movements. We average this deviation for all test runs of a test stage. The value indicated for the deviation percentage is the percentage added

or subtracted to the expected distance.

6.2.1 Training Runs

Setup

To test how the number of training runs affected the performance of the model cerebellum we did the following. We started by measuring the performance without training the model cerebellum. Then we measured after training with 150 runs. Each stage we incremented the number of runs by 150 until we reached 1500 training runs. We run 100 test sets for each stage. Each test set was composed of the training runs defined by the stage followed by 50 test runs. A 20 by 20 units field was used for all runs and the confidence threshold was set to 35%.

Results

All configurations using the zoom resolution strategy achieved a near 100% success rate after 750 training runs. Configurations using the tile resolution strategy did much worse. The weighted random selection strategy only achieved a 90% success rate after 900 runs while the other two only reached 80% after 1500 runs.

As expected the bigger the amount of training runs the less the model cerebellum would rely on the model cerebrum. The average indecision rate decreases very quickly until the 600 runs stage then slowly converges towards zero. All the strategy combinations had similar performances. There was none that really stood apart from the rest.

For the distance deviation percentage, all the zoom strategy configurations perform much better than the tile strategy configurations. Of particular interest is the threshold strategy combined with the zoom strategy. Starting with the 750 training runs stage, the average distance travelled in a test run is shorter than the expected distance. For instance, the movements executed at the 1350 training runs stage were on average 10.9% shorter than their expectations.

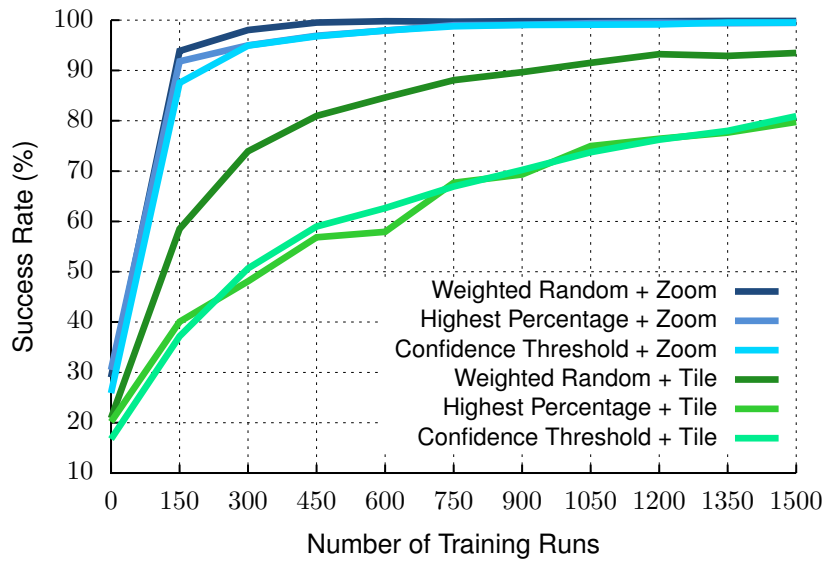


Figure 6.1: Variation of the success rate with the number of training runs.

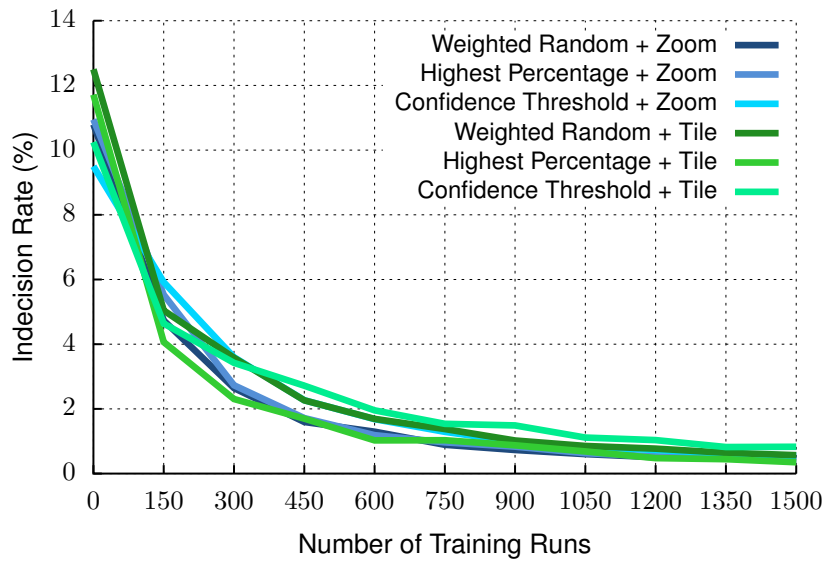


Figure 6.2: Variation of the indecision rate with the number of training runs.

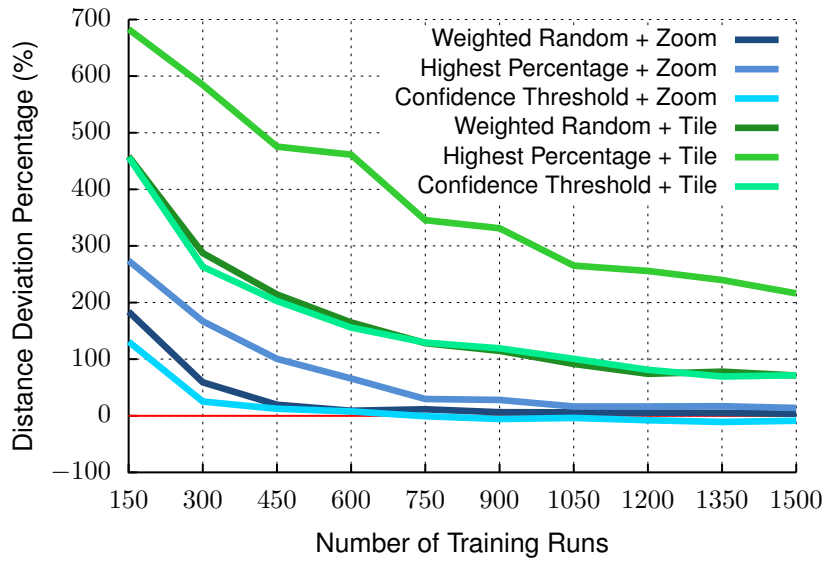


Figure 6.3: Variation of the distance deviation percentage with the number of training runs.

6.2.2 Field Dimensions

Setup

For the field dimensions, we started with a 5 by 5 field and increased each dimension by 10 units ending with a 255 by 255 field. For each stage, there were 100 test sets. Each of these sets was composed of 300 training runs followed by 100 test runs. The confidence threshold was set at 35%.

Results

As expected, the success rate drops as the field becomes larger. The configurations using the tile strategy perform much worse and have a significant drop in performance once the field becomes 95 by 95 units. At this point, the success rate becomes only 8% and rises slowly as the field increases in size.

Something similar happens with the distance deviation percentage. The tile strategy configurations start with very large deviations from the expectation. However, at the 95 by 95 field stage the deviation drops drastically and their performance becomes better than the zoom strategy configurations using the same movement selection strategy.

The average indecision rate increases as the field increases. Configurations using the zoom strategy perform worse in this measure in contrast with the success rate results. This is because the indecision rate only reveals if the model cerebellum is making decisions by itself not if progress towards the goal has been made.

We suspect that the changes when the field becomes 95 by 95 units in size have to do with its combination with the goal circle radius used and the grid resolution achieved. These conditions affect the size of the cells of the grid, how much the resolution is increased and the distance travelled by the agent with each step.

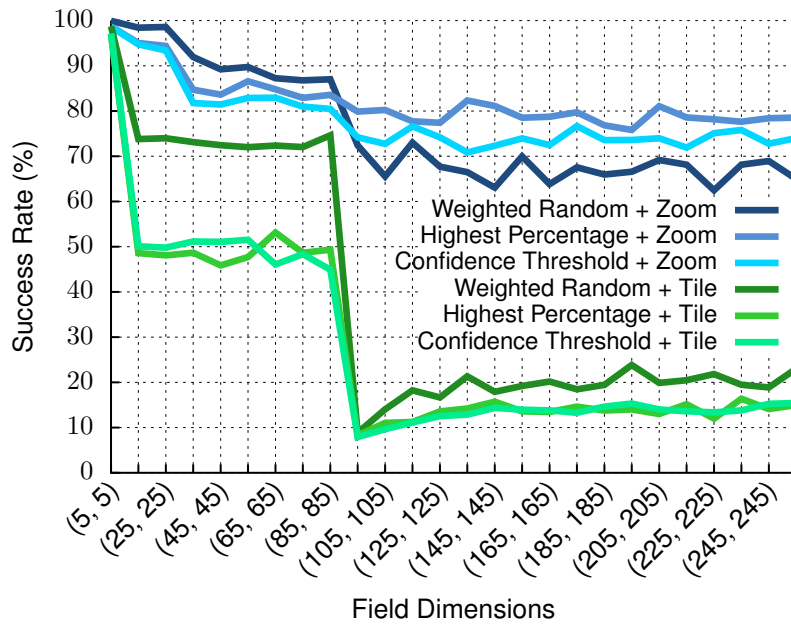


Figure 6.4: Variation of the success rate with the size of the environment.

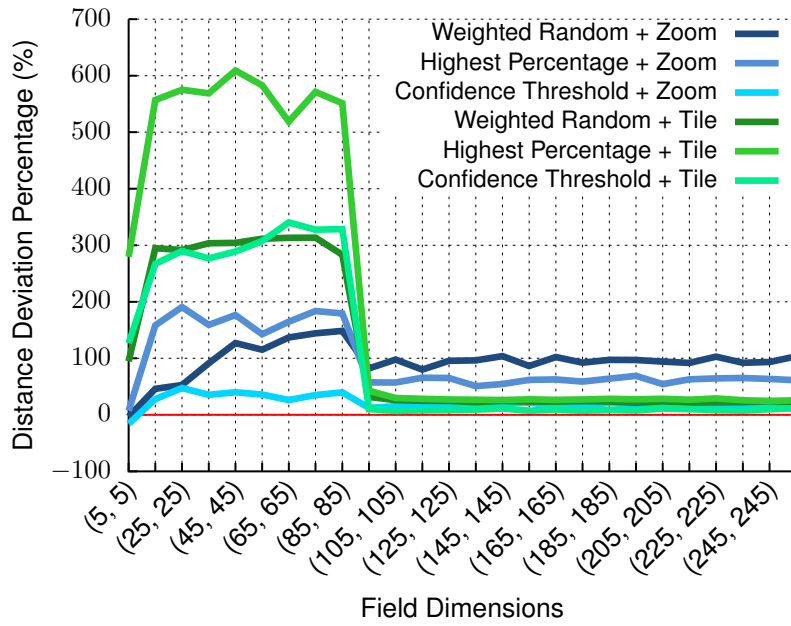


Figure 6.5: Variation of the distance deviation percentage with the size of the environment.

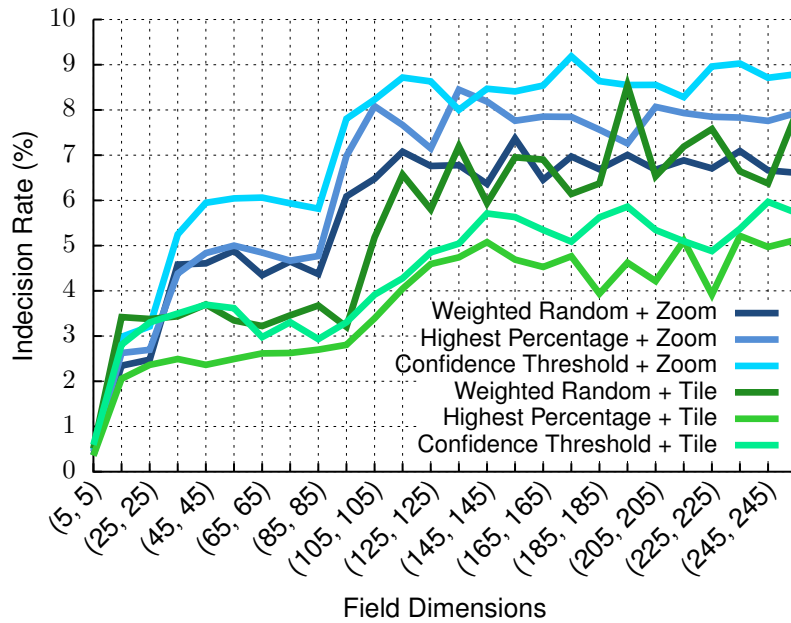


Figure 6.6: Variation of the indecision rate with the size of the environment.

6.2.3 Confidence Threshold

Setup

The confidence threshold is only used by the threshold selection strategy. Therefore, only this strategy is tested for this variable in combination with the resolution update strategies. As a result, only two configurations are compared for this variable. We started with a threshold of 0%, which means every elementary movement that was observed at least once for a context once will be executed. Between each stage, the threshold is increased by 5%. At the final stage, the threshold is 95%. For each set there were 100 test sets composed of 2000 training runs followed by 50 test runs. This number of training runs was chosen to make the differences between each threshold more visible. Finally, a 20 by 20 units field was used for all the runs.

Results

The configuration using the zoom resolution strategy achieves a success rate above 98% with a confidence threshold of 5% and higher. The tile strategy configuration only has a success rate above 90% beginning with the 55% threshold stage.

Both configurations have an average indecision rate of below 1% up until the confidence threshold becomes higher than 50%. After that point the indecision rate increases rapidly. At 90%, the zoom strategy has an indecision rate of approximately 50.6% while the tile configuration is undecided around 47.8% of the time.

In relation to the distance deviation percentage, the tile configuration performs much worse. It always adds at least 20% to the expected distance. On the other hand, the zoom configuration always achieves on average distances shorter than expectation for confidence thresholds below 50%. This is because

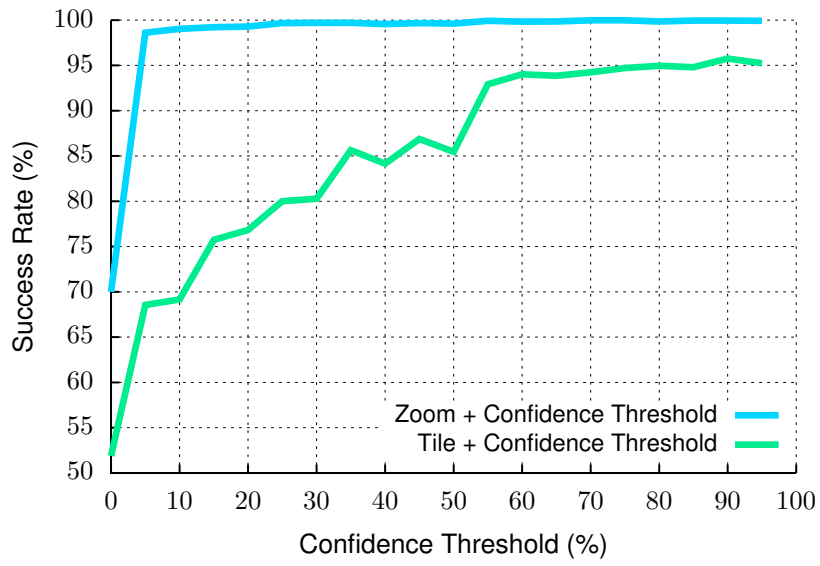


Figure 6.7: Variation of the success rate with the confidence threshold.

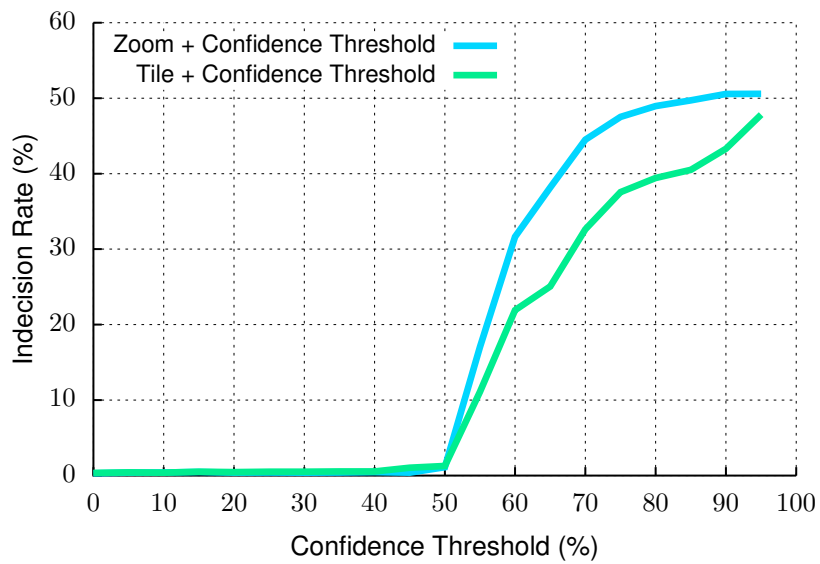


Figure 6.8: Variation of the indecision rate with the confidence threshold.

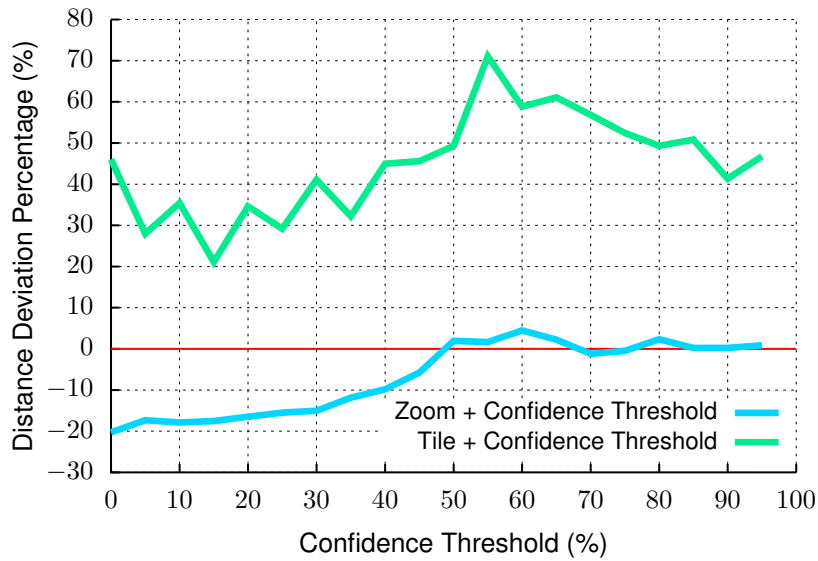


Figure 6.9: Variation of the distance deviation percentage with the confidence threshold.

in those stages it is possible to select multiple elementary movements and combine them to move diagonally. Even with thresholds above 50%, this configuration adds less than 5% to the expected distance.

6.2.4 Summary of the Results

All zoom strategy configurations had a better success rate than the tile strategy configurations. For this measure the highest percentage selection strategy combined with the zoom resolution strategy had the best performance overall.

For the indecision rate, the tile configurations did better when varying the field size but worse when varying the amount of training runs. However, the highest percentage strategy combined with the tile strategy still had the best results for both cases. Despite that, it is important to note that having a better indecision rate does not necessarily mean a better chance of reaching the goal. This particular configuration had the worst success rate both when varying the field size and when varying the number of training runs.

Finally, for the distance deviation percentage the configurations using the confidence threshold strategy did the best. The configuration using the zoom strategy had the best performance and was able to achieve distances shorter than the expectation on average in certain stages. The confidence threshold strategy is able to combine multiple elementary movements and move the agent diagonally which gives it an advantage in this measure.

Configuration	IR (%)	SR (%)	DDP (%)
Zoom + Weighted Random	0.40	99.92	2.50
Zoom + Highest Percentage	0.41	99.52	13.89
Zoom + Confidence Threshold	0.61	99.58	-9.98
Zoom + Point Threshold	48.42	100.00	-1.23
Tile + Weighted Random	0.55	94.52	60.16
Tile + Highest Percentage	0.46	83.42	177.52
Tile + Confidence Threshold	0.76	79.53	61.61
Tile + Point Threshold	53.64	100.00	-4.46

Table 6.2: Comparison of the point threshold strategy with the other strategy combinations using 1500 training runs. IR - Indecision Rate; SR - Success Rate; DDP - Distance Deviation Percentage.

6.3 Ninth Version of the Model

6.3.1 Setup

In order to test the new strategy, we need to define how many points are earned or lost, the maximum amount of points and the point threshold used to select an elementary movement. Our intention when selecting these values was to make it possible for two elementary movements to beat the point threshold and allow the agent to move diagonally.

Whenever the model cerebrum selects an elementary movement, the movement earns 2 points for that context. All the other elementary movements lose 1 point for the same context. The point threshold is 20 points and the maximum amount of points that can be accumulated for a context is 50 points.

With this setup, the following simplistic scenario is possible. Elementary movement A is observed 25 times for a specific context, earning a total of 50 points. All other movements remain at zero points, which is the minimum amount of points possible. Then, elementary movement B is observed 21 times, earning 22 points. Movement A will lose 11 points and remain with 39 points. Both movements beat the point threshold and the model cerebellum will combine them. Even if other elementary movements were to be observed occasionally for this context there is still some margin that allows both A and B to beat the point threshold simultaneously.

We tested this strategy in combination with the zoom and tile resolution update strategies. The other strategy combinations introduced before were also tested using the same testing setup and the results were compared. The same three performance measures were compared: the indecision rate, the success rate and the distance deviation percentage.

For each strategy combination, we ran 100 test sets. Each test set was composed of 1500 training runs followed by 100 test runs. The environment was a field of 20 by 20 units. For the confidence threshold strategy, the threshold was set at 35%. As before, a timestep limit of 1000 steps was defined for the test runs.

6.3.2 Results

Table 6.2 summarizes the results. The new strategy was able to achieve a 100% success rate in both configurations. However, it has a very high miss rate when compared with the other configurations. This

Configuration	IR (%)	SR (%)	DDP (%)
Zoom + Weighted Random	0.15	99.97	0.66
Zoom + Highest Percentage	0.17	99.75	7.17
Zoom + Confidence Threshold	0.26	99.83	-14.52
Zoom + Point Threshold	29.57	100.00	-2.09
Tile + Weighted Random	0.23	97.41	26.77
Tile + Highest Percentage	0.12	88.47	118.16
Tile + Confidence Threshold	0.34	89.93	19.39
Tile + Point Threshold	51.58	100.00	-4.36

Table 6.3: Comparison of the point threshold strategy with the other strategy combinations using 3000 training runs. IR - Indecision Rate; SR - Success Rate; DDP - Distance Deviation Percentage.

is to be expected because with the point threshold strategy the model cerebellum must build enough confidence with an elementary movement before using it in a specific context. The agent moves through many distinct contexts between the start and goal positions and the model cerebellum has to make enough observations for each of them in order to stop relying on the model cerebrum.

Looking at the distance deviation percentage, we see that the trajectories produced by the new strategy are on average shorter than the Manhattan distance to the goal. This means that the strategy was successful in combining elementary movements and moving the agent diagonally.

Repeating the tests using 3000 training runs per test set, the results are as shown in table 6.3. As we can see, the indecision rate has improved significantly for the zoom and point threshold strategy combination but it is still much higher than indecision rate of the other configurations. For the point threshold strategy, there was no change in the success rate and the deviation rate only changed slightly.

Chapter 7

Conclusion

In this dissertation, we reviewed the influential works of Marr and Albus, which sparked a lot of interest in the field of computational neuroscience. We also reviewed a less known theory, proposed by Mechsner. His theory focuses on a property of the cerebellar cortex not explored by previous theories, the existence of microzones. It is suggested that multiple microzones contribute to the execution of a single elementary movement, allowing for more refined and adaptable motions. Mechsner proposed that these microzones also perform a coupling operation that groups together movements frequently executed together, improving their coordination.

However, we believe that this theory could be expanded with the knowledge acquired from more recent research findings. We found that the cerebellum was involved in predicting the sensory consequences of our actions. This ability would be useful in improving the performance in motor coordination as it provides a mechanism for error correction and feed-forward execution of movement.

Based on this concept and the concept of internal models, we proposed a computational model, which implements the coupling operation and expands Mechsner's theory with predictive abilities. If successful, in the future this model could be applied in the engineering field to optimize the mobility of robots.

We then began implementing a model that guides an agent between two points in a two-dimensional space. The model was built from scratch and started out very simple. We train a model cerebellum by observing how the model cerebrum guides the agent. Based on the observations, the model cerebellum makes associations that it can use to reproduce movement patterns.

The model went through many versions and we experimented with many concepts. Some issues we had to solve were how the model cerebellum would select the elementary movement to perform at a given moment and how to refine the trajectory of the agent. For the first issue, we tried various strategies that would use the associations made by the model cerebellum. The last strategy we implemented uses a point system to determine the strength of those associations. Using this system, it is capable of combining multiple elementary movements to produce a new movement. It also solved a problem we had with the model cerebellum making associations prematurely which would sometimes impede it from guiding the agent to the goal.

In order to refine the movements generated by the model we added a way to change the resolution at which the model cerebellum perceives the environment. We did this by creating a view of the environment centered on the position of the agent. This view would start by dividing the surroundings of the agent into a small number of zones and associating elementary movements to these zones. The number of zones is then increased in order to make smaller adjustments to the trajectory and guide the agent closer and closer to the goal. In other words, the model cerebellum starts with a general view of the environment, which then becomes more and more detailed.

The final version of the model, however, still remains far from the ideas described by the theories on cerebellar function described in chapter 2 and our proposed solution. Although it can learn simple movement patterns and reproduce them to guide the agent towards the goal, work still needs to be done in order to make it produce a smooth trajectory and able to adapt movement patterns to varying conditions. An idea to make the movement smoother would be to keep the momentum created by an elementary movement between timesteps. The elementary movements would change from simply changing the coordinates of the agent to instead applying an acceleration to it.

Bibliography

- [1] André-Thomas: Cerebellar Functions. Number 12 in Nervous and Mental Diseases Monograph Series. *Journal of Nervous and Mental Diseases* (1912)
- [2] Herculano-Houzel, S.: Coordinated scaling of cortical and cerebellar numbers of neurons. *Frontiers in neuroanatomy* **4**(March) (2010) 12
- [3] Holmes, G.: The symptoms of acute cerebellar injuries due to gunshot injuries. *Brain* **40**(4) (1917) 461–535
- [4] Powell, K., Mathy, A., Duguid, I., Häusser, M.: Synaptic representation of locomotion in single cerebellar granule cells. *eLife* **4**(JUNE2015) (2015) 1–18
- [5] Lee, K.H., Mathews, P.J., Reeves, A.M., Choe, K.Y., Jami, S.A., Serrano, R.E., Otis, T.S.: Circuit mechanisms underlying motor memory formation in the cerebellum. *Neuron* **86**(2) (2015) 529–540
- [6] Thach, W.: Combination, complementarity and automatic control: a role for the cerebellum in learning movement coordination. *Novartis Foundation Symposium* **218** (1998) 219
- [7] Albus, J.S.: A theory of cerebellar function. *Mathematical Biosciences* **10**(1-2) (feb 1971) 25–61
- [8] Marr, D.: A theory of cerebellar cortex. *The Journal of physiology* **202**(2) (jun 1969) 437–70
- [9] Blakemore, S.J., Frith, C.D., Wolpert, D.M.: The cerebellum is involved in predicting the sensory consequences of action. *Neuroreport* **12**(9) (jul 2001) 1879–84
- [10] Nixon, P.D.: The role of the cerebellum in preparing responses to predictable sensory events. *Cerebellum* (London, England) **2**(2) (jan 2003) 114–22
- [11] Bell, C.C., Grant, K.: Corollary discharge inhibition and preservation of temporal information in a sensory nucleus of mormyrid electric fish. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **9**(3) (mar 1989) 1029–44
- [12] Bell, C.C., Han, V., Sawtell, N.B.: Cerebellum-like structures and their implications for cerebellar function. *Annual review of neuroscience* **31**(1) (jul 2008) 1–24
- [13] Mechsner, F.: A New Theory of Cerebellar Function. *Connection Science* **8**(1) (mar 1996) 31–54
- [14] Purves, D. In: *Neuroscience*. 3rd edn. Sinauer Associates, Inc, Sunderland, Mass (2004) 438–441, 592–597

- [15] Bengtsson, F., Jorntell, H.: Sensory transmission in cerebellar granule cells relies on similarly coded mossy fiber inputs. *Proceedings of the National Academy of Sciences* **106**(7) (2009) 2389–2394
- [16] Huang, C.C., Sugino, K., Shima, Y., Guo, C., Bai, S., Mensh, B.D., Nelson, S.B., Hantman, A.W.: Convergence of pontine and proprioceptive streams onto multimodal cerebellar granule cells. *eLife* **2013**(2) (2013) 1–17
- [17] Ishikawa, T., Shimuta, M., Häusser, M.: Multimodal sensory integration in single cerebellar granule cells in vivo. *eLife* **4** (2015) 1–10
- [18] Wagner, M.J., Kim, T.H., Savall, J., Schnitzer, M.J., Luo, L.: Cerebellar granule cells encode the expectation of reward. *Nature* **544** (mar 2017) 96
- [19] Duguid, I., Branco, T., Chadderton, P., Arlt, C., Powell, K., Häusser, M.: Control of cerebellar granule cell output by sensory-evoked Golgi cell inhibition. *Proceedings of the National Academy of Sciences* **112**(42) (oct 2015) 13099–13104
- [20] Eccles, J.C., Ito, M., Szentágothai, J.: *The Cerebellum as a Neuronal Machine*. Springer Berlin Heidelberg, Berlin, Heidelberg (1967)
- [21] Pinzon-Morales, R.D., Hirata, Y.: A realistic bi-hemispheric model of the cerebellum uncovers the purpose of the abundant granule cells during motor control. *Frontiers in neural circuits* **9**(May) (may 2015) 18
- [22] Tyrrell, T., Willshaw, D.: Cerebellar Cortex: Its Simulation and the Relevance of Marr's Theory. *Philosophical Transactions of the Royal Society B: Biological Sciences* **336**(1277) (1992) 239–257
- [23] Willshaw, D.J., Buneman, O.P., Longuet-Higgins, H.C.: Non-holographic associative memory. *Nature* **222**(5197) (jun 1969) 960–2
- [24] Steinbuch, K.: Die Lernmatrix. *Kybernetik* **1**(1) (1961) 36–45
- [25] Sacramento, J.: Associative Memory for Binary Image Storage and Retrieval
- [26] Ito, M., Sakurai, M., Tongroach, P.: Climbing fibre induced depression of both mossy fibre responsiveness and glutamate sensitivity of cerebellar Purkinje cells. *The Journal of Physiology* **324**(1) (1982) 113–134
- [27] Ogasawara, H., Doi, T., Kawato, M.: Systems biology perspectives on cerebellar long-term depression. *Neuro-Signals* **16**(4) (2008) 300–17
- [28] Lüscher, C., Huber, K.M.: Group 1 mGluR-Dependent Synaptic Long-Term Depression: Mechanisms and Implications for Circuitry and Disease. *Neuron* **65**(4) (2010) 445–459
- [29] Hirano, T.: Long-term depression and other synaptic plasticity in the cerebellum. *Proceedings of the Japan Academy. Series B, Physical and biological sciences* **89**(5) (2013) 183–95

- [30] Hoxha, E., Tempia, F., Lippiello, P., Miniaci, M.C.: Modulation, plasticity and pathophysiology of the parallel fiber-purkinje cell synapse. *Frontiers in Synaptic Neuroscience* **8**(November) (2016) 1–16
- [31] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **65**(6) (1958) 386–408
- [32] Sawtell, N.B.: Multimodal integration in granule cells as a basis for associative plasticity and sensory prediction in a cerebellum-like circuit. *Neuron* **66**(4) (2010) 573–584
- [33] Strata, P.: David Marr’s theory of cerebellar learning: 40 years later. *The Journal of Physiology* **587**(23) (dec 2009) 5519–5520
- [34] Rokni, D., Llinas, R., Yarom, Y.: The Morpho/Functional Discrepancy in the Cerebellar Cortex: Looks Alone are Deceptive. *Frontiers in neuroscience* **2**(2) (dec 2008) 192–8
- [35] Oscarsson, O.: Functional units of the cerebellum - sagittal zones and microzones. *Trends in Neurosciences* **2**(Supplement C) (jan 1979) 143–145
- [36] Apps, R., Garwicz, M.: Anatomical and physiological foundations of cerebellar information processing. *Nature Reviews Neuroscience* **6**(4) (apr 2005) 297–311
- [37] Witter, L., De Zeeuw, C.I.: Regional functionality of the cerebellum. *Current Opinion in Neurobiology* **33** (2015) 150–155
- [38] Cohn, J.V., DiZio, P., Lackner, J.R.: Reaching during virtual rotation: context specific compensations for expected coriolis forces. *Journal of neurophysiology* **83**(6) (2000) 3230–3240
- [39] Bubic, A., von Cramon, D.Y., Schubotz, R.I.: Prediction, cognition and the brain. *Frontiers in human neuroscience* **4**(March) (2010) 25
- [40] Requarth, T., Sawtell, N.B.: Plastic Corollary Discharge Predicts Sensory Consequences of Movements in a Cerebellum-Like Circuit. *Neuron* **82**(4) (2014) 896–907
- [41] Pfordresher, P.Q., Beasley, R.T.E.: Making and monitoring errors based on altered auditory feedback. *Frontiers in psychology* **5**(AUG) (2014) 914
- [42] Konczak, J., Abbruzzese, G.: Focal dystonia in musicians: linking motor symptoms to somatosensory dysfunction. *Frontiers in Human Neuroscience* **7**(June) (2013) 1–10
- [43] Flanagan, J.R., Wing, a.M.: The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **17**(4) (1997) 1519–1528
- [44] Cerminara, N.L., Apps, R., Marple-Horvat, D.E.: An internal model of a moving visual target in the lateral cerebellum. *The Journal of physiology* **587**(2) (jan 2009) 429–42
- [45] Selden, S.T.: Tickle. *Journal of the American Academy of Dermatology* **50**(1) (jan 2004) 93–7

- [46] Harris, C.: The Mystery of Ticklish Laughter. *American Scientist* **87**(4) (1999) 344
- [47] Blakemore, S.J., Wolpert, D.M., Frith, C.D.: Central cancellation of self-produced tickle sensation. *Nature neuroscience* **1**(7) (nov 1998) 635–40
- [48] Blakemore, S.J., Wolpert, D., Frith, C.: Why can't you tickle yourself? *Neuroreport* **11**(11) (aug 2000) R11–6
- [49] Sawtell, N.B., Bell, C.C.: Adaptive processing in electrosensory systems: Links to cerebellar plasticity and learning. *Journal of Physiology Paris* **102**(4-6) (2008) 223–232
- [50] Montgomery, J.C., Bodznick, D., Yopak, K.E.: The cerebellum and cerebellum-like structures of cartilaginous fishes. *Brain, Behavior and Evolution* **80**(2) (2012) 152–165
- [51] Warren, R., Sawtell, N.B.: A comparative approach to cerebellar function: insights from electrosensory systems. *Current Opinion in Neurobiology* **41** (2016) 31–37
- [52] Requarth, T., Sawtell, N.B.: Neural mechanisms for filtering self-generated sensory signals in cerebellum-like circuits. *Current Opinion in Neurobiology* **21**(4) (2011) 602–608
- [53] Bell, C.C.: Sensory coding and corollary discharge effects in mormyrid electric fish. *The Journal of experimental biology* **146** (sep 1989) 229–53
- [54] Han, V.Z., Bell, C.C., Grant, K., Sugawara, Y.: Mormyrid electrosensory lobe in vitro: Morphology of cells and circuits. *Journal of Comparative Neurology* **404**(3) (1999) 359–374
- [55] Hodos, W. In: *Evolution of Cerebellum*. Springer Berlin Heidelberg, Berlin, Heidelberg (2009) 1240–1243
- [56] Gellman, R., Gibson, A.R., Houk, J.C.: Inferior olivary neurons in the awake cat: detection of contact and passive body displacement. *Journal of Neurophysiology* **54**(1) (jul 1985) 40–60
- [57] Lou, J.S., Bloedel, J.R.: Responses of sagittally aligned Purkinje cells during perturbed locomotion: synchronous activation of climbing fiber inputs. *Journal of neurophysiology* **68**(2) (1992) 570–580
- [58] Nitschke, M.F., Stavrou, G., Melchert, U.H., Erdmann, C., Petersen, D., Wessel, K., Heide, W.: Modulation of cerebellar activation by predictive and non-predictive sequential finger movements. *Cerebellum (London, England)* **2**(3) (sep 2003) 233–40
- [59] Thach, W.T.: On the specific role of the cerebellum in motor learning and cognition: Clues from PET activation and lesion studies in man. *Behavioral and Brain Sciences* **19**(03) (sep 1996) 411–433
- [60] Devor, A.: Is the cerebellum like cerebellar-like structures? *Brain Research Reviews* **34**(3) (2000) 149–156
- [61] Devor, A.: The great gate: control of sensory information flow to the cerebellum. *The Cerebellum* **1**(1) (jan 2002) 27–34

- [62] Albus, J.: The Marr and Albus theories of the cerebellum—two early models of associative memory. In: Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, IEEE Comput. Soc. Press (1989) 577–582
- [63] Houk, J.C., Buckingham, J.T., Barto, A.G.: Models of the cerebellum and motor learning. *Behavioral and Brain Sciences* **19**(03) (sep 1996) 368–383
- [64] Martin, T.A., Keating, J.G., Goodkin, H.P., Bastian, A.J., Thach, W.T.: Throwing while looking through prisms. I. Focal olivocerebellar lesions impair adaptation. *Brain : a journal of neurology* **119** (Pt 4(4) (aug 1996) 1183–98
- [65] Schlerf, J., Ivry, R.B., Diedrichsen, J.: Encoding of Sensory Prediction Errors in the Human Cerebellum. *Journal of Neuroscience* **32**(14) (apr 2012) 4913–4922
- [66] Nixon, P.D., Passingham, R.E.: The cerebellum and cognition: Cerebellar lesions do not impair spatial working memory or visual associative learning in monkeys. *European Journal of Neuroscience* **11**(11) (1999) 4070–4080
- [67] Nixon, P.D., Passingham, R.E.: The cerebellum and cognition: Cerebellar lesions impair sequence learning but not conditional visuomotor learning in monkeys. *Neuropsychologia* **38**(7) (2000) 1054–1072
- [68] Wolpert, D.M., Miall, R.C., Kawato, M.: Internal models in the cerebellum. *Trends in cognitive sciences* **2**(9) (sep 1998) 338–47
- [69] Stein, J.: Cerebellar forward models to control movement. *The Journal of physiology* **587**(2) (jan 2009) 299
- [70] Moberget, T., Gullesen, E.H., Andersson, S., Ivry, R.B., Endestad, T.: Generalized role for the cerebellum in encoding internal models: evidence from semantic processing. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **34**(8) (feb 2014) 2871–8
- [71] Azim, E., Alstermark, B.: Skilled forelimb movements and internal copy motor circuits. *Current Opinion in Neurobiology* **33** (2015) 16–24
- [72] Francis, B.A., Wonham, W.M.: The internal model principle of control theory. *Automatica* **12**(5) (1976) 457–465
- [73] Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. *Neural networks : the official journal of the International Neural Network Society* **11**(7-8) (oct 1998) 1317–29
- [74] Stoodley, C.J., Valera, E.M., Schmahmann, J.D.: Functional topography of the cerebellum for motor and cognitive tasks: An fMRI study. *NeuroImage* **59**(2) (jan 2012) 1560–1570
- [75] Strick, P.L., Dum, R.P., Fiez, J.A.: Cerebellum and Nonmotor Function. *Annual Review of Neuroscience* **32**(1) (jun 2009) 413–434

- [76] Tsai, P.T., Hull, C., Chu, Y., Greene-Colozzi, E., Sadowski, A.R., Leech, J.M., Steinberg, J., Crawley, J.N., Regehr, W.G., Sahin, M.: Autistic-like behaviour and cerebellar dysfunction in Purkinje cell Tsc1 mutant mice. *Nature* **488**(7413) (aug 2012) 647–651
- [77] Wolf, U., Rapoport, M.J., Schweizer, T.A.: Evaluating the affective component of the cerebellar cognitive affective syndrome. *The Journal of neuropsychiatry and clinical neurosciences* **21**(3) (2009) 245–53
- [78] Fulbright, R.K., Jenner, A.R., Mencl, W.E., Pugh, K.R., Shaywitz, B.A., Shaywitz, S.E., Frost, S.J., Skudlarski, P., Constable, R.T., Lacadie, C.M., Marchione, K.E., Gore, J.C.: The cerebellum's role in reading: a functional MR imaging study. *AJNR. American journal of neuroradiology* **20**(10) (1999) 1925–30
- [79] Christian, K.M.: Neural Substrates of Eyeblink Conditioning: Acquisition and Retention. *Learning & Memory* **10**(6) (nov 2003) 427–455
- [80] Matsuda, K., Yoshida, M., Kawakami, K., Hibi, M., Shimizu, T.: Granule cells control recovery from classical conditioned fear responses in the zebrafish cerebellum. *Scientific Reports* **7**(1) (dec 2017) 11865
- [81] ten Brinke, M.M., Boele, H.J., Spanke, J.K., Potters, J.W., Kornysheva, K., Wulff, P., Ijpelaar, A.C.H.G., Koekkoek, S.K.E., De Zeeuw, C.I.: Evolving Models of Pavlovian Conditioning: Cerebellar Cortical Dynamics in Awake Behaving Mice. *Cell reports* **13**(9) (dec 2015) 1977–88
- [82] Giovannucci, A., Badura, A., Deverett, B., Najafi, F., Pereira, T.D., Gao, Z., Ozden, I., Kloth, A.D., Pnevmatikakis, E., Paninski, L., De Zeeuw, C.I., Medina, J.F., Wang, S.S.: Cerebellar granule cells acquire a widespread predictive feedback signal during motor learning. *Nature Neuroscience* **20**(5) (mar 2017) 727–734
- [83] Stoodley, C.J., Valera, E.M., Schmahmann, J.D.: Functional topography of the cerebellum for motor and cognitive tasks: An fMRI study. *NeuroImage* **59**(2) (2012) 1560–1570