June 2017

Multi-Scale Spatial Cognition Models and Bio-Inspired Robot Navigation

Martin I. Llofriu Alonso
University of South Florida, mllofriualon@mail.usf.edu

Scholar Commons Citation

Llofriu Alonso, Martin I., "Multi-Scale Spatial Cognition Models and Bio-Inspired Robot Navigation" (2017). *Graduate Theses and Dissertations.*
http://scholarcommons.usf.edu/etd/6888

Multi-Scale Spatial Cognition Models and Bio-Inspired Robot Navigation

by

Martin Llofriu Alonso

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Alfredo Weitzenfeld, Ph.D.
Yu Sun, Ph.D.
David Diamond, Ph.D.
Miguel Labrador, Ph.D.
Wilfrido Moreno, Ph.D.

Date of Approval:
May 17, 2017

Keywords: Reinforcement Learning, Hippocampus, NeuroRobotics, Path Planning, Long-Term Operation

DEDICATION

To my fellow skydiver, and to all those who built the parachute

ACKNOWLEDGMENTS

It seems that after completing a dissertation, is easier to enumerate who you are not grateful to, but I'll try the standard way. Thanks to Alfredo Weitzenfeld for his support, freedom, advise and patience; to the members of the committee for their revisions and challenges; to Jean-Marc Fellous, Marco Contreras and Tatiana Pelc for their biology insights and revisions; to Gonzalo Tejera and Facundo Benavides for their comments and support from far away; to Juan Calderon for the perfectly complementing knowledge in robotics; to Pablo Scleidorovich, I wish we had you a year earlier; to Muhaimen Shamsi for his help with SLAM; and to Nadia Riera, mainly for the patience. Finally, I would like to thank Agencia Nacional de Investigación e Innovación (ANII) and the National Science Fundation (NSF) for their financial support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The rodent navigation system has been the focus of study for over a century. Discoveries made lately have provided insight on the inner workings of this system. Since then, computational approaches have been used to test hypothesis, as well as to improve robotics navigation and learning by taking inspiration on the rodent navigation system.

This dissertation focuses on the study of the multi-scale representation of the rat's current location found in the rat hippocampus. It first introduces a model that uses these different scales in the Morris maze task to show their advantages. The generalization power of larger scales of representation are shown to allow for the learning of more coherent and complete policies faster.

Based on this model, a robotics navigation learning system is presented and compared to an existing algorithm on the taxi driver problem. The algorithm outperforms a canonical Q-Learning algorithm, learning the task faster. It is also shown to work in a continuous environment, making it suitable for a real robotics application.

A novel task is also introduced and modeled, with the aim of providing further insight to an ongoing discussion over the involvement of the temporal portion of the hippocampus in navigation. The model is able to reproduce the results obtained with real rats and generates a set of empirically verifiable predictions.

Finally, a novel multi-query path planning system is introduced, inspired in the way rodents represent location, their way of storing a topological model of the environment and how they use it to plan future routes. The algorithm is able to improve the routes in the second run, without disrupting the robustness of the underlying navigation system.

CHAPTER 1

INTRODUCTION

Over the last century, a strong focus has been devoted to the study of the rodent navigation system [55, 118, 140, 117, 129, 146]. This is mainly due to the interesting properties this system presents. To name a few, rodents can modulate their behavior depending on their needs [149], are able to solve complex mazes [147], perfect routes and motion sequences over time [118, 55], show teleological (purposeful) behaviors [37, 164], maintain a working memory of already visited places [119], find and remember hidden locations [101], make use of shortcuts [163], carry out structured processes of hypothesis testing [164] and navigate back to a home place in complete darkness [96]. Many of these abilities suggested that rodents have an internal topological and metrical representation of space, which was termed the cognitive map [164].

The robustness and flexibility of these animals make them a good source of inspiration for long-term navigation in robotics. At the same time, our knowledge of robotic navigation algorithms allow for further understanding of how rodents carry out tasks and their decision making processes. The fields of computational neuroscience and neurorobotics focus on the study of living animals for two purposes:

1. to improve our understanding of how animals make decisions, with the help of computational tools and knowledge from robotics, i.e. modeling.
2. to draw inspiration from our growing understanding of the inner workings of neuroscience to improve robotics algorithms for navigation and learning, i.e. bio-inspired algorithm design.

This dissertation focuses on modeling the rat navigation system. Rats have a multi-scale representation of their current location, as it is explained in the chapter 3. A model-free reinforcement learning framework based on multiple scales of representation of the animal location is implemented as the core for the models and algorithms presented.

The goal of this dissertation is to study the multi-scale representation in spatial cognition to:

1. model the use of multiple scales of representation in rodent behavior, to shed light on an ongoing neuroscience discussion of whether larger scales influence navigation decisions.

2. analyze how such multi-scale models could improve current robotics algorithms, to create new bio-inspired algorithms for robot navigation and learning.

Throughout this dissertation, models of rodent navigation for increasingly complex tasks are introduced, building up to the model described in chapter 7. The first task consists of navigating to a single goal in an open environment, with a local visually guided component. The next task introduces static obstacles. The final task introduces many new features: multiple goals, semi-dynamic environments, different types of visual stimuli and hippocampus deactivation. Interleaved with the presentation of the models, bio-inspired algorithms for robotic navigation and learning are introduced.

The rest of this dissertation is organized as follows:

- Chapter 2 introduces a brief analysis of the state of the art of path planning and learning, with an emphasis on model-free systems (no map).

- Chapter 3 introduces an analysis of the rodent navigation system as a black box and the analysis of the state of the art of rodent navigation models in this black box framework.

- Chapter 4 describes how the multi-scale navigation system was modeled throughout the dissertation. It lays out the ground concepts to understand the contributed models and algorithms.

- Chapter 5 presents a model of multi-scale navigation that shows the advantages of the larger scales of representation in a Morris maze task. It shows that including larger scales of representation speeds up learning and generates a more coherent policy in an obstacle free environment.

- Chapter 6 presents a new robotics navigation learning algorithm based on a multi-scale representation, which outperforms canonical Q-Learning in a navigation task. This bio-inspired algorithm shows to learn faster than canonical QL in a discrete environment with obstacles. A continuous state-space version is also presented.

- Chapter 7 presents the main contributions of this dissertation. A new real-rat task co-developed with a collaborating lab, and model for this novel task that is able to reproduce the results observed with real rats. A set of predictions that could be empirically tested on real rats are drawn from the model.

- Chapter 8 describes a new robotics navigation algorithm that builds on top of robust reactive path planners like the Bug algorithms. This algorithm is inspired on the way rats store sparse maps of the environment and how they used them to plan future paths.

- Chapter 9 summarizes the contributions and draws conclusions from all the obtained results.

CHAPTER 2

STATE OF THE ART OF ROBOT PATH PLANNING AND REINFORCEMENT LEARNING

This chapter introduces the state of the art for robot path planning and reinforcement learning for this dissertation.

Section 2.1 includes an overview of the state of the art in path planning, with a focus on model free, i.e. no map, path planners. This summarizes the different ways to decide *how* to get to the desired goal, in order to understand the decisions made in the modeling part, as well as to understand the contributions of chapter 8.

Section 2.2 introduces reinforcement learning, the framework that is going to be used to model the learning processes in rats, in order to understand how rat learning is modeled and to understand the contribution of chapter 6.

2.1 Path Planning in Robotics

Path planning consists of finding a sequence of motions to get a mobile robot in a desired configuration or goal. Figure 2.1 shows an example environment used in this section to exemplify the different path planning solutions. This problem has been widely studied and many algorithms have been proposed to solve the problem, under different working hypothesis. For a good review, see Latombe's book [83] and Howie and collaborator's [24].

Path planning algorithms can be classified according to different criteria, in reference to their working hypothesis. Figure 2.2 shows a broad classification of the main path planning algorithms according to four different criteria:

- completeness: this refers to whether the algorithm will always be able to find a path, given one exists. Some algorithms might be complete depending of how much resources are allotted to them. Time complete and probabilistically complete algorithms find a solution given enough time is devoted to the algorithm.

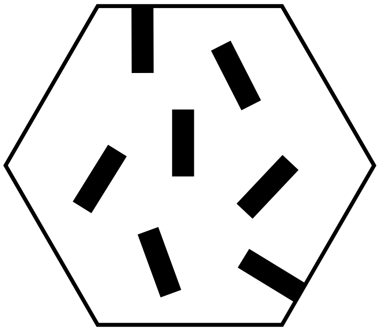


Figure 2.1: A sample map of an hexagonal environment.

- obstacle complexity: the more complex the obstacle representation, the harder it is to compute a path. This criteria focuses on the complexity of the obstacles as represented in the configuration space (c-obstacle)[1].
- dimensionality of the configuration space: this represents the degrees of freedom for a mobile robot. A robot that can only move in a plane, but not rotate, will move in a configuration space of dimension 2. If it can rotate, it will be 3. The configuration space of a flying drone is 6-dimensional.
- what is being optimized: path planning is a multi-objective optimization problem. The path length can be optimized, but also the risk of collision can be minimized. As with these two metrics, there is usually a trade off between the different optimization goals.

Another important criteria to classify path planners is the kind of information they require as input. Some require a structure that describes the free and occupied space, e.g. a map, as shown in Figure 2.1. Other relax this criteria to request for an oracle function: given a point in the configuration space, the function will specify whether it is free or occupied (i.e. a probing function). A third kind requires no map or only local maps to make their decisions.

In the next subsections the four main categories of path planners are reviewed.

2.1.1 Cell Decomposition Path Planners

These methods consist of finding a decomposition of the environment into disjoint cells with the connectivity between them. Once the division is completed, a graph of connected free space is obtained. The path is then found with graph search methods, such as Dijkstra or A*.

[1] In the configuration space, the mobile robot is considered a point that represents its position in space. Thus, obstacles must be transformed (usually enlarged), to account for the fact that the robot is usually not a point.

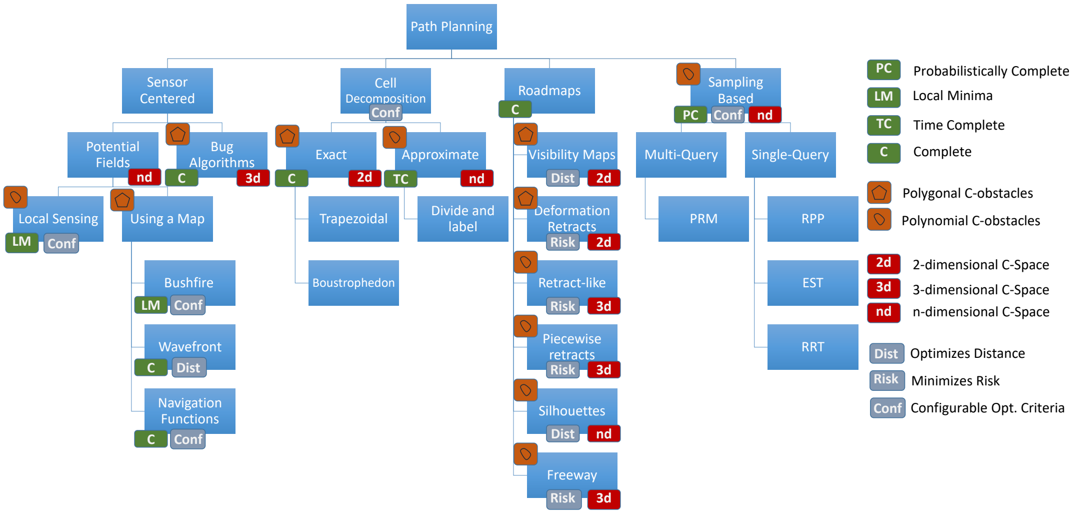


Figure 2.2: Path planning algorithms. Their classification for completeness (green), c-obstacle complexity (orange), c-space dimensionality (red) and optimization criteria (blue) are shown.

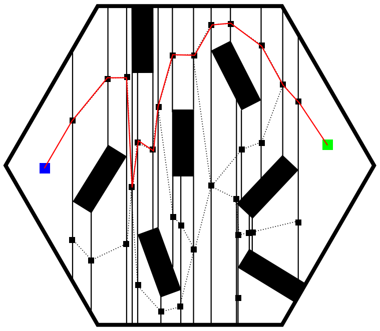


Figure 2.3: Cell decomposition of an environment. The cell decomposition is shown in continuous lines. Black squares represent the nodes. The generated graph is shown in dashed lines. A sample path from the blue square to the green square is shown in red.

These methods need a full map of the environment. Figure 2.3 shows a cell decomposition of the environment, where the middle point of each cell boundary becomes a node in the graph.

Decomposing the configuration space into cells might be a hard problem. Some algorithms are exact: they find a decomposition such that the union of the cells marked as free covers the whole free space. Others find approximate decomposition, in which free cells form a subset of the free space. Approximate decomposition algorithms are easier to compute. However, if all available paths go through some uncovered free space, the algorithm will fail. Exact methods, on the other hand, are complete.

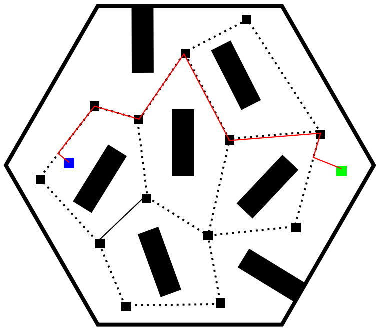


Figure 2.4: Roadmap in an example environment. The generated graph is shown in continuous black lines. Black squares represent the nodes. A sample path from the blue square to the green square is shown in red.

Usually approximate algorithms have recursive ways of decomposing the map, such that if enough time is given to the algorithm, they will find a decomposition good enough to be used to find a path. Thus, they are said to be time complete.

It is important to notice that usually exact methods work under more constrained hypothesis, making them less generic than approximate decomposition.

2.1.2 Roadmap Path Planners

Roadmap planners consist of finding a low dimensional structure that can be used to navigate the map. For example, in a 2d map, a graph containing a few nodes can lay general routes to navigate the space successfully. Once this structure is computed, a path can be found by connecting the initial position and goal position to the graph. These methods need a full map of the environment. Figure 2.4 shows a roadmap computed in an environment.

Roadmap planners belong to a category of importance to this dissertation, that of multi-query path planners. This means that the efforts invested in planning a path can be reused when planning a second path in the same environment.

Roadmaps planners have the disadvantage of always navigating the roadmap. This generic pathway is usually not optimal for every case. In the case of visibility maps, the roadmap involves always going as close as possible to obstacles, risking impact. Deformation retracts can be optimal for risk minimization at the expense of path length.

Although they are complete, the more generic versions of roadmap planners are hard to implement and computationally expensive.

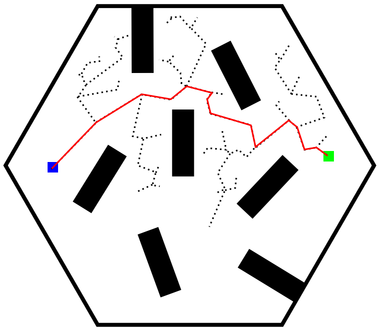


Figure 2.5: Sampling tree and planned path in an example environment. The tree structure formed is shown in dashed black lines. A sample path from the blue square to the green square is shown in red.

2.1.3 Sampling-Based Path Planners

Sampling-based planners use oracle functions to query the map. Thus, they do not need a full map of the environment.

Like roadmap planners, they create a navigable structure by sampling a point from free space and connecting it to the existing structure. Once this structure connects the initial and goal position, a path can be found within it. Figure 2.5 shows the built tree in an environment.

Sampling-based planners converge with probability one if infinite time is allotted. They are relatively easy to implement and can handle complex environments. However, they can be inefficient in simple maps, if the possible path goes through narrow spaces.

2.1.4 Sensor-Based Path Planners

Sensor-based path planners are the most important to this dissertation. Most of them work under the hypothesis of a known robot and goal position, but an incomplete or no map at all. They use local sensor information and the goal position to decide where to go next. As it is explained in chapter 3, this is the information available to rodents as well.

The big disadvantage of these methods is their incompleteness or inefficiency. Some variants are complete and efficient, but they only apply to simple environments and require a full map.

One category of sensor based path planners, artificial potential fields (APF) [74], combine attractive fields, which lead the robot to a certain location, with repulsive fields that keep the robot away from obstacles. Its main drawback is the generation of local minima, where the robot is unable to make progress. Other methods use stateful strategies to circumvent the obstacles and reach the desired goal, which include the early sensor based bug algorithms Bug1 and Bug2 [89].

Some variants of these methods are shown to converge, despite the fact that they are only based on the available sensor information and the robot localization. On the other hand, they tend to traverse paths significantly longer than the minimum length one.

Efforts have been made to overcome these shortcomings. Improvements fall into three different categories:

- improving the efficiency of slow but complete algorithms (e.g. bug algorithms),
- dealing with the completeness problem of otherwise more efficient algorithms (e.g. APF algorithms), and
- dealing with multi-query planning by integrating knowledge of the environment as it becomes available.

2.1.4.1 Improving the Efficiency of Complete Algorithms

Many contributions have focused on improving the original bug algorithms [112]. One notable approach worth mentioning is the Tangent Bug [71]. This algorithm assumes mid-to-long range distance sensors with a relatively fine resolution, e.g. a LIDAR sensor. Then, it uses this information to avoid the need to go straight to obstacles, trying to circumvent them before it reaches them. Tangent bug generates more optimal paths in many environments, but it does not improve them over sessions.

Much work has been devoted to improving the switching logic of the bug algorithms. The Prediction Based Bean Curvature Method (PBCM) [141] uses a previous BCM methods to find openings using the distance sensor profile. The Distance Histogram Bug (DH-Bug) [180] takes the relative orientation of the goal and obstacle to exit the obstacle following mode more prematurely. Zhu et al. [178] keep a graph of the traveled path and check the robot-goal line before exiting the obstacle following mode, to avoid infinite loops. Wai et al. [170] designed a more elaborated switching strategy that takes many variables into account, such as “obstacle in front”, “obstacle around”, distance to the left vs to the right. It shows promising results, but its convergence properties are not proven.

Other methods change the switching logic for a fuzzy sensor-based controller. Motlagh et al. [106] use the relative angle to the closest obstacle and the goal to compute the next action to perform, and add obstacle distance and wheel slippage later [105]. Jarada et al. [65] implement APF as a fuzzy controller.

2.1.4.2 Improving the Completeness of Efficient Algorithms

Huang et al. [64] proposed a visual-based APF that is able to achieve navigation results similar to those of the Tangent Bug.

Zhang et al. [177] used simulated annealing to escape from APF local minima. They also address the issue of the target being to close to the obstacle.

2.1.4.3 Integrating New Knowledge of the Environment

Some methods build on top of underlying sensor-based algorithms, while using the available knowledge of the environment to improve the quality of the final path. Zhu et al. [179] use the DH-Bug algorithm as the sensor based component, while an A* search over a grid is used to find subgoals that lead the DH-Bug through shorter paths.

Others use partial or global knowledge of the environment to construct a navigation function with only one local minima. Daily et al. [29] use a partial map to build a navigation function that solves the Dirichlet problem, where the navigation function has known value at the obstacles and goal, while satisfying the Laplace equation. Conner et al. [26] also solve the Dirichlet problem by decomposing the map into a fixed size grid.

Coleman et al. [25] proposed a path planner based on RRT that stores plans on a graph that can be queried in the future to reuse the experience.

2.2 Reinforcement

Reinforcement learning belongs to the field of machine learning. It lies between supervised and un-supervised learning. In supervised learning, the response to a set of examples is to be learned. The output for these examples is known in advance. In un-supervised learning, structure is to be recognized from a set of examples, without knowledge of the correct response or category for those examples. In reinforcement learning, an agent traverses a set of states performing actions. The correct actions to be performed in each state are not known, but need to be learned. Feedback, however, is only given to the agent on a reduced set of state or state-action pairs. The agent, then, has to figure out which state-action pairs led to positive feedback from experience. For a good review of reinforcement learning, see [155].

Reinforcement learning is applicable when the problem involves executing a sequence of actions to reach a desired state or configuration, where the set of correct state-action pairs is not known in advance. In other words, problems where one knows what is needed, but not how to get it.

2.2.1 Mathematical Framework

In reinforcement learning, we are going to consider an agent that is always in some state $s \in S$, where S is the set of all possible states. For the sake of simplicity, lets consider a grid navigation problem, where the agent is in a 3×3 grid, and has to achieve a certain goal position. Here, s is the currently occupied grid.

A set of possible actions A can be executed by the agent. Each action takes the agent from one state to another, according to a model function, as shown in Eq. 2.1. In the grid example, actions could be: go north, go south, go east, go west.

$$m : S \times A \to S :: m(s, a) = s' \tag{2.1}$$

The model can also be non-deterministic, assigning a probability p to the transition of state s to s', by action a, as shown in Eq. 2.2. In the grid example, this could mean that the agent might end up traveling east when actually trying to execute the "go north" action, with a certain probability.

$$m : S \times A \times S \to [0, 1] :: m(s, a, s') = p \tag{2.2}$$

A reward value is assigned to the state action pairs. The reward policy function r specifies these values, as shown in Eq. 2.3. In the grid example, all state-action pairs that land on the goal could be rewarding.

$$r : S \times A \to \Re :: r(s, a) = r \tag{2.3}$$

Then, the problem becomes finding a policy for the agent. A policy π is a function that will determine the action to execute when in each state, as shown in Eq. 2.4.

$$\pi : S \to A :: \pi(s) = a \tag{2.4}$$

2.2.2 Solutions to the Reinforcement Learning Problem

Three main approaches are available to find the optimum policy:

- dynamic programming
- Montecarlo methods
- temporal difference learning

Dynamic programming problems assume the learner has access to both the model m function and the reward policy r function. They also assume a finite number of states and actions. Under these hypothesis, the policy can be computed using dynamic programming techniques. This approach is not applicable to many situations, as the assumptions usually don't hold, and when they do, the computational cost can be prohibitive.

Montecarlo methods sample a sequence of state-actions pairs $(s, a)_1..(s, a)_N$ by exploration. Then, they update a structure reflecting the earned outcome of each pair. This method does not need access to the model or the reward policy, as it samples them directly from exploration. Its main disadvantage is that it fails to use the information learned so far to guide exploration, which can make it an unfeasible solution when the state-action space is too big.

Temporal difference learning (TD learning) algorithms maintain a table $Q(s, a)$ that reflects the suitability of a certain action in a certain state. This table is used to learn from every action taken, rather than waiting until the end of the batch, as done in Montecarlo methods. This makes this method more applicable to high-dimensional state-action space problems. Thus we are going to focus on this approach.

In order to select an action when in state s, a greedy approach would pick the action to perform according to (2.5).

$$a = argmax_a Q(s, a) \tag{2.5}$$

The way this table is updated varies, depending on the specific TD algorithm. We review two variations relevant to the dissertation below.

2.2.3 Temporal Difference Learning Variations

Two variations of TD learning are of interest for this dissertation: Q-Learning (QL) and actor-critic (AC). They differ in the way the update the $Q(s,a)$ table, and thus, in the way they learn.

After every action a is taken from state s, arriving to state s', QL updates the table according to Eqs. (2.6) and (2.7), where α and γ modulate the learning speed and r is the obtained reward signal.

$$\delta = r + \gamma max_{a'} Q(s',a') - Q(s,a) \tag{2.6}$$

$$Q(s,a) = Q(s,a) + \alpha * \delta \tag{2.7}$$

The actor-critic algorithm, on the other hand, maintains a separate value table $V(s)$. This table represents the reward expected to be obtained, when departing from state s. After performing an action a that takes the agent from state s to state s', the tables are updated as specified in Eqs. 2.8, 2.9 and 2.10. Here λ and α represent the same as in Eqs. 2.6 and 2.7.

$$\delta = r + \lambda V(s') - V(s) \tag{2.8}$$

$$V(s) = V(s) + \alpha\delta \tag{2.9}$$

$$Q(s,a) = Q(s,a) + \alpha * \delta \tag{2.10}$$

The maximization step performed in QL updates speed up learning. This is due to the fact that the value of expected reward is estimated as the maximum over possible actions, which promotes the learning of “shortcuts” by concatenating pieces of discovered solutions based on an optimistic prediction.

However, this same maximization makes learning about negative outcomes very slow, as it ignores the expected values of other actions. This can be catastrophic for non-greedy (exploring) agents. The actor-critic algorithm, on the other hand, maintains a separate value table that summarizes the expected reward of all actions. This makes it better to learn both positive (pursuit) and negative (avoid) outcomes.

In addition, the actor-critic has been widely adopted by the modeling community because instantiating it in the rodent brain is more straight-forward than for QL [155].

2.2.4 Continuous Space Reinforcement Learning

When applying RL to robotics navigation or when modeling rodent decision making as RL, the problem of continuous states arises. TD algorithms keep tables with entries for each state or state-action pair. This means that the states must be discrete for them to work. The next action to perform, however, usually depends on the location of the agent, which is a continuous multi-variate value. We present modifications to the TD framework that account for continuous state spaces.

The most intuitive approach to apply tabular RL to a continuous state problem is to make an arbitrary discretization of the continuous variables, in order to get a discrete set of possible states. This is the case with Nemec et al. [111] where a robot learns to play “ball in a cup”. The robot state consists of a coarse hand-made discretization of the cup position, velocity, ball angle and angular velocity. Benbrahim et al. [13] robot learns to balance a ball on a beam by rotating it perpendicular to the axis in which the ball moves. They discretize the ball position and velocity, as well as the beam angle and angular velocity to get just 180 discrete states, which are used in an Actor-Critic RL algorithm. Work by Kimura et al. [75] consists of learning to move forward on a 4 legged robot without prior knowledge of the dynamics of the robot. They discretize the angular position of each of the robot’s 8 motors into 8 discrete values, obtaining 256 possible states. Tokic et al. [162] make use of value iteration to teach a robot to crawl using passive wheels and a two joint arm. The angular position of both actuators in the arm is discretized into 5 possible values. Discretizing the state space arbitrarily consists of a valid approach, but involves expert knowledge to decide a good trade-off between learning speed, due to fewer states, and the final policy’s precision and smoothness, allowed by a higher number of states.

Another approach is to learn the discretization strategy from data itself. This can be described in general as applying unsupervised clustering techniques to some input information to obtain a resulting discretized state space. Asada et al. [6] use a clustering algorithm to discretize sensor data and apply it to a navigation problem of getting a robot to shoot a ball. Takahashi et al. [156] also make use of clustering, using a nearest neighborhood approach. They work in the domain of robot soccer, as well. It is interesting to note how their work uses the reinforcement signal to discretize the state space near the goal state, something that could be related to some findings on the interaction between reward signals and hippocampus learning [61]. This is also done by Piater et al. [124], in which a visual state space is adaptively segmented using the TD error from a RL algorithm. Toshikuyi et al. [176] use Bayesian discrimination to adaptively segment the state space into clusters and apply it to multi-robot environments.

Other approaches make use of relational logic to generate abstract policies that may apply to similar, unseen domains. Cocora et al. [99] use relational logic to learn high-level navigation policies on building floors segmented into rooms. Katz et al. [72] use relational logic to describe the nature of the objects that a robot is learning to interact with. Morales [99] use relational logic to express abstract features of the state of a navigational task, allowing an agent to learn general movement policies. These techniques represent a good approach to lower the dimensionality of the problem while introducing domain knowledge that can increase learning speed. However, they rely somewhat on the availability of an expert to design the high level features or relations.

Finally, another category of approaches consists of using function estimators to learn the value function over the continuous variables directly. This could be understood as using supervised machine learning to acquire a map from the continuous state to the expected return, where the reinforcement signal is used as the error signal. There are plenty of approaches in this direction and we will mention only some of them to convey its main ideas. Gasket et al. [51] use a neural network to learn both the value function and selected action from a given state, coded from visual and kinesthetic information. It learns to wander a hallway, moving smoothly and avoiding obstacles. Duan et al. [41] combine fuzzy neural networks with Q-Learning to learn the problem of navigating towards a goal avoiding obstacles. The Bellman error is used as a trainer to correct weights and membership function parameters using back-propagation. Riedmiller et al. [133] accomplish autonomous learning of several basic robot soccer skills such as dribbling, defending and speed control. It uses a multi-layer perceptron with feedback from the Q function of the successor state. These works have obtained good results and the work in this dissertation is complementary to them, since they usually do not impose any restriction on the codification of their inputs. However, the convergence warranties provided by RL do not apply to this kind of function approximators [76].

Singh et al. [138] theoretical work introduces soft-state aggregation or clusters as an intermediate layer between the continuous states and the RL algorithm. Each state cluster s_i defines a conditional probability distribution $p(s_i|c)$ of original (continuous) state c belonging to the cluster s_i. Then, the Q-Learning algorithm is applied to the cluster states. Singh et al. work's approach is similar to the one used in this dissertation, in the sense that it allows for several states to be active concurrently. However, the concept of having multiple scales of clusters was not considered in Singh et al. work. We adopted from this work the concept of *soft-state*, which we will explain more in detail in chapter 4. This concept matches very well with place cells as the input to the navigation system, as discussed in chapter 3.

CHAPTER 3

STATE OF THE ART IN RAT NAVIGATION AND MODELING

This chapter contains an in-depth overview of rat navigation and existing models. It is divided in three sections:

- Section 3.1, reviews the navigational capabilities of rodents and the experimental designs used to test them. This section serves two purposes, it analyzes the rodent navigation abilities and properties, and it establishes a framework by which computational models can be analyzed

- Section 3.2 reviews the physiological properties of the rodent navigation system, namely, the neural basis for the system.

- Section 3.3 analyzes computational models of the rodent navigation system by using the framework established in Section 3.1. It draws a correspondence between the topics presented in Section 3.1, and how the reviewed work models them.

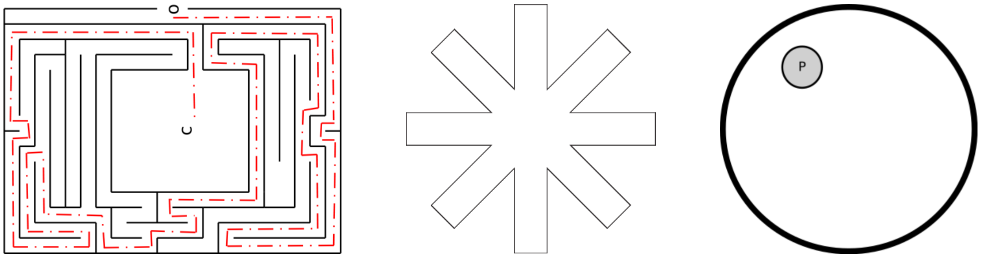
3.1 Rat Navigation

Many computational models of navigation have been proposed based on the nature of hippocampal cells and their involvement in navigation (see [168, 91, 139] for reviews). Many different approaches are used to explain a diversity of properties of the rodent navigation system, which makes their comparison difficult.

When modeling the rodent navigation system, it is important to understand its abilities and limitations. We include below a summary of some important properties of this system, considering it as a black box. This will also serve to establish a criterion to analyze the reviewed computational models later on.

3.1.1 The Need for Motivation

The need for a motivation factor to facilitate learning is of general consensus nowadays. This was already recognized in early experiments involving instrumental conditioning, in which the



(a) Small maze [147]. The dashed line shows one possible solution.

(b) Radial Arm Maze.

(c) Morris maze [101] diagram. The gray circle denotes the hidden platform.

Figure 3.1: Rat experimental mazes.

animals had to learn to perform a certain action to achieve a reward [149]. Neither hungry nor scared animals failed to explore the universe of possible actions, preventing them from discovering the ones that led to rewards. Moreover, if they did discover the rewarding actions, the lack of motivation would slow down the learning process.

3.1.2 Complex Decision Chain Learning

Besides being able to learn simple stimulus-action associations, as shown in Skinner's [145] experiments, rats are able to learn a chain of actions that will lead them to rewards. This can be used in navigation by learning the correct series of right and left turns that lead to a certain goal.

Small [147] designed the maze shown in Figure 3.1a. Rats were first familiarized with the environment overnight in a non-rewarded fashion. Later on, the rats were placed in the start position o while food was located in the center of the maze c. Small observed that rats could eventually learn the task, lowering the amount of errors made as trials went on. This discovery settled a baseline of what a rat can do navigation wise.

Small also noticed how learning required that the animal solved the problem first by pure trial and error, or by pure luck. The animal's learning abilities would then allow it to learn from that first successful experience. This is consistent with the learning approach of the later developed theory of reinforcement learning [155].

3.1.3 Behavior Stereotyping

Early studies had a strong focus on a rat's ability to stereotypy behaviors. Namely, rather than emphasizing the animal's ability to learn flexible behaviors, the attention was drawn to its ability to master a given task [118]. Small [147] observed in his experiments how rats would not only solve a given complex maze, but progressively perfect their navigation until they reached a state of "...a practically perfect knowledge of the maze, so that they can make the journey quickly and accurately..."

Later qualitative work presented by Gingerelli [55] showed how it was possible to over train rats in a chain of arbitrary turns, to the point when each rat "bumped his nose into the end of the blind alley with considerable violence" when the maze was modified.

Stolz and Lott [151] showed that rats over trained to follow a corridor to obtain one pellet of food tend to pass through a pile of pellets without paying attention to them. Furthermore, rats keep going to the end of the corridor even if the single pellet is removed.

3.1.4 Working Memory

Besides learning instrumental actions, and chains of navigation decisions, rats are capable of maintaining a record of recent events, i.e. working memory.

Olton and Samuelson [119] designed an experiment to test the rat's ability to remember choices made in the near past. A rat was placed to navigate in a maze consisting of a central circular base of approximately 30 cm diameter connected to 8 arms that extended radially, as shown in Figure 3.1b. All arms were baited only once in the distant end at the beginning of the trial. The rat had to avoid visiting any given arm more than once, as it would result in a non-rewarded choice.

Rats were indeed able to avoid repeated visits to the same arm. The average of correct decisions made within the first 8 visits was 7.6 after 5 days of training. Control experiments were carried out to discard odor, intra-maze cues and sequential strategies; thus attributing spatial working memory as the factor mediating the rat's success.

3.1.5 Reference Memory

Besides being able to remember a chain of choices and the most recent made ones, rats are able to remember (and navigate to) a certain place in the environment, which is called reference memory.

Olton and Samuelson's [119] maze was also used to test for reference memory in corridor-based environments [126]. Only a fixed subset of arms were baited at the beginning of each trial, and the rat's ability to learn and remember which ones were baited was assessed. This differs from learning a chain of actions, like in [147] experiments, because the animal has to learn how to get to the baited places from a single location (the center), rather than how to get to a single baited place through many choices. Rats were indeed capable of learning the baited arms, decreasing the amounts of errors made as trials went by.

Experiments carried out by Morris [101] showed that rats are able to learn a certain location, but this time in an open environment. That is, they were able to learn a reference frame, or map, anchored to distal cues, that allowed them to reach a certain patch in an open environment.

His maze consisted on a circular pool of 1 m diameter. The pool was filled with water and milk, to make it opaque. A platform was placed in the pool, with its top being a few centimeters below water. Thus, the platform was not visible to the animal. Figure 3.1c shows the layout of the maze. The rat was placed in different locations of the pool, from where they had to swim to the hidden platform. The water was set below room temperature, motivating the rat to escape from it.

Morris showed that the animals could learn the location of the hidden platform, decreasing the time to reach it from trial to trial. The average escape latency was reduced to 50% after 8 trials and to 20% after 32, when comparing the average latency of the first 8 trials.

3.1.6 Directional Navigation

Pearce [121] showed that rats are able to find a hidden platform that is at a fixed angle (in a global reference frame) of a landmark. Moreover, hippocampal-lesioned rats are also able to do so, implying that this mechanism differs to that of reference memory, which is usually considered to be linked to the hippocampus.

3.1.7 Shortcuts and Novel Routes

During the mid-twentieth century, a great discussion revolved around whether rats were able to navigate through novel routes upon changes in the environment.

Tolman et al. [163] tested pre-trained rats in a modified maze in which the original path to reward was blocked, whereas a lot of new alleys were laid out in different directions. Figure 3.2 shows this maze. They found that 36% of the rats took the alley that led in the direction of the reward during a single test trial. However, they also account for the possibility that the observed behavior was due to the rats being conditioned to follow the light bulb used, which was cuing the food location.

More recently, rats have been shown to take shortcuts consistently (97% of the sessions) when a new shortcut is opened in an M maze [1]. Furthermore, once the shortcut is discovered, it is taken most of the time by the animals (90% of the times). It can be argued that this implies the

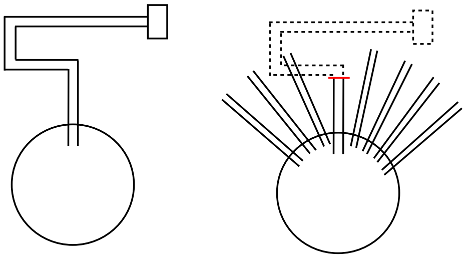


Figure 3.2: Tolman maze for the metric shortcut experiment.

existence of structures that allow the rat to compute the advantage of taking the new, shorter path. However, the urge of the animals to take the new route could be accounted to curiosity.

3.1.8 Sensory Modalities

Rodents have a wide variety of information sources available for navigation. They can be divided in two categories, proprioceptive and exteroceptive. Proprioceptive senses provide the animal with information about itself, such as the position of each joint, the movement currently being performed (kinesthetics) and motivational aspects (e.g. hunger). Exteroceptive senses provide the animal with information about the outside world, such as vision, hearing or tactile information. Experimenters usually divided exteroceptive into two categories: intra-maze proximal cues (e.g. walls, cue cards, floor texture) and extra-maze distal cues (e.g. posters on the lab walls, computers, light sources).

In early studies, proprioceptive information was assigned the most important role [118]. This might have been due to the fact that most tests were carried out after habitual learning (over training) had taken place. However, Small [147] recognizes, in a discussion about two blind subjects, that vision might have mediated the learning process.

An influential review by [131], based on a review by [109], settled some important conclusions on the topic:

- proprioceptive cues are enough for a rat to learn how to solve simple mazes, but are not sufficient for learning how to solve more complex ones,
- once a maze has been learned, proprioceptive cues alone allow the rat to solve it, and
- intra or extra maze cues (exteroceptive) can be scrambled one at a time, retarding but not preventing learning of a maze.

3.1.9 Teleological vs Habitual Actions

After observing the phenomena of behavior stereotyping and automation, researchers were faced with one controversial question: are the actions of animals ruled by a goal oriented decision processes (teleological) or do they belong to an innate/acquired habitual stimulus-response schema? In other words, is a rat thinking of the desired food at the end of a corridor when it runs through it? More explicitly, is it reasoning about the consequences of traversing the corridor, i.e. getting to the desired food? Or is it just that it has learned, throughout the learning trials, that running forward when in that corridor leads to a reward?

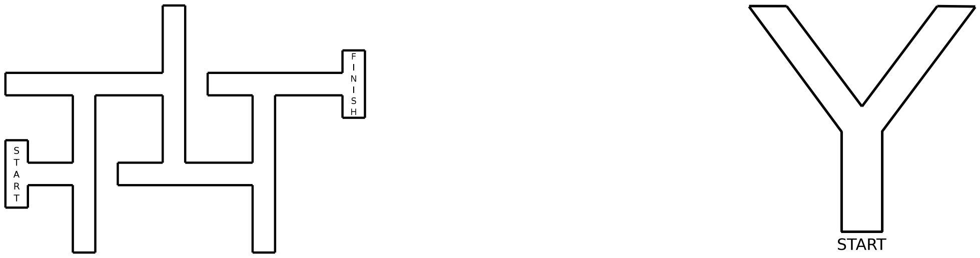
Dickinson [37] argued that rats are indeed capable of both types of behavior, and that distinguishing between them could be absolutely non-trivial. He used tools from instrumental conditioning to show that if a goal is devalued after learning, rats tend to decrease goal-seeking behaviors (extinction), suggesting they understand the consequences of their actions. Dickinson also argued that special control groups need to be tested to distinguish between both types of behavior, something that was not done in many of the latent learning experiments, described below.

3.1.10 Latent Learning

The concept of latent learning arouse as a consequence of the habitual vs teleological discussion. In order to disprove the stimulus-response theory of navigation, defenders of the teleological hypothesis (also named field-theorists) argued that if non-rewarding trials given prior learning could affect the learning speed, then it would be shown that rats are able to learn something besides stimulus-response associations that lead to reward [164].

The importance of maze habituation was already recognized by Small [147]. In his experiments, the rats were allowed to explore the environment overnight, providing knowledge of the maze prior to the rewarded trials.

An experiment by Blodgett [14] showed how the time to reach a goal in a 4T maze (Figure 3.3a) decreased more rapidly the more familiar rats were with the environment, prior to the rewarding trials. The experiments consisted in dividing the animals in different groups. A control group had food delivered at the end of the maze right from the first session. For other two groups, the experimental groups, food was available only after a number or habituation sessions. Rats in the experimental groups showed faster drop rates in the time to reach the food than control rats.



(a) Blodgett 4T maze.

(b) Y-maze.

Figure 3.3: Mazes used in latent learning experiments.

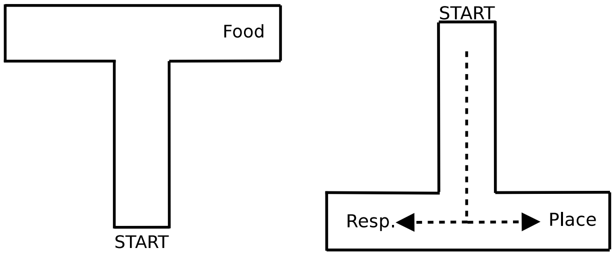
This implies that these groups where learning something about the maze during in the unrewarded sessions. Tolman [164] argued they were building an internal representation of the environment, the cognitive map.

Another experiment showed how rats are able to use prior knowledge of the environment to direct their behavior so as to satisfy their current needs [148]. Neither hungry nor thirsty rats were trained to become familiar with a Y maze (Figure 3.3b) that contained food in one arm and water in the other. Later on, half the rats were food deprived, while the other half where water deprived. Results show that thirsty rats went to the water arm and hungry rats to the food arm in the first run. This implies that the rats had been learning the location of the different resources while not hungry nor thirsty and were able to use this knowledge to satisfy their needs.

3.1.11 Place vs Response

The latent learning discussion had an underlying assumption that later became a strong topic of debate. For latent learning to express, the rat must be making navigation decisions based on those learned models of the environment, though this was not a consensus [163]. Some argued that animal's actions come in fact from space reasoning (place), while others argued that they come from stimulus conditioned actions (response). It is important to notice that here stimulus includes both exteroceptive ones (vision, olfactory, etc) and proprioceptive ones (kinesthetic, mental states).

This dilemma can be summarized in one experiment. Rats are put a T maze (Figure 3.4) and trained to always find food in the right arm. After training, a testing trial is performed in which the maze is rotated with respect to the room, which is supposed to be the frame of reference for the map in the rat's mind. If the rat makes a right turn, it means it has learned a response behavior



(a) T-maze. (b) T-maze.

Figure 3.4: Place vs response experiment.

(always turn right), whereas if it turns left, it means it has learned to always go to an absolute place in the frame of reference of the room.

Although this issue is tightly coupled with the previously presented topic of latent learning, they are not exactly the same. The discussed experiment by Blodgett [14] showed how rats learn the environment without any reinforcement, allowing them to learn a good policy faster in the reinforced trials. This shows latent learning, but the learned actions that take the rat to the reward could be the product of a conditioned stimulus-response that is just facilitated by the previously built model of the environment. Spence's [148] experiment, on the other hand, shows both latent learning and place decision-making. When the thirsty (hungry) rats are first put into the maze, they use their previously learned model of the environment (latent learning) to make a navigation plan that would take them to the water (food). The decision involves using the map to plan the consequences of actions (i.e. place decision-making).

In a good review work [131], Restle poses the problem as response vs. place domination. Namely, he shows with previous data that under different training conditions, response dominates place actions, while the reverse is true for different training conditions. In summary, visual cue rich environments tend to favor place behaviors, whereas monotonous environments tend to favor response ones. Placing trials together in time seems to favor place behaviors too.

Devan [33] performed alternation between visible and hidden platform trials in the same location. On the last day, they performed a competition trial in which the visible platform was placed opposite to where it was found during training. They showed that by lesioning different parts of the brain they could bias rats to either perform response or place behaviors.

In conclusion, the current consensus is that rats are able to learn both place and response behaviors, while the dominant one depends on the learning conditions.

3.1.12 Hypothesis Testing and Structured Exploration

Another interesting question relates to how rats make navigation decisions when they have no prior information of the environment, namely when they are exploring.

An experiment by Tolman [164] showed that when rats try to solve a 4 randomized binary decision runaway task (i.e. the solution changes from trial to trial), they display concrete “hypothesis testing” behaviors. Namely, they show strategies like always choosing right or left, or always selecting the doors that look alike. This experiment shed light on the fact that exploratory actions in the rat are not entirely ruled by chance, but by predefined strategies that are tested one by one [78].

In addition, rats tend to adopt different strategies when searching for the escape platform in a Morris arena. These strategies include swimming near the border, making turns into the center of the pool and swimming in circles [173]. This also shows there is a considerable difference between rat exploring behaviors and a random walk process.

3.1.13 One-Trial Learning

Steele and Morris [150] showed an interesting phenomenon called one-trial learning. The experiment consisted on a modified version of the Morris water pool, in which the hidden platform is located in a new position every session (4 trials). Thus, the rat was forced to remember the new position of the platform after the first trial and apply it to the following trials in the same session. They showed that after some pre-training, rats reached the platform in significantly less time in the second trial than in the first one. This effectively shows that the rat is able to encode the position of the platform in a single trial, challenging reinforcement learning approaches (model-free), as they are considered not able to reproduce this kind of results.

3.1.14 Path Integration and Homing

Another interesting phenomena linked to both sensory modalities and the cognitive map is the ability of animals to return to a home location upon navigating in the dark. Mittelstaedt [96] showed that mice could return to their home from a suckling search task using “a rather straight course”. They also showed that if the maze is rotated, the error made by the animals is equal to the rotation amount. This was not the case for when the animal was rotated, in which they return home successfully. Maaswinkel and Whishaw [90] designed an experiment in which olfactory cues are made irrelevant by rotating the outer part of a circular maze while the blindfolded rat is in the middle. They showed that upon rotation, the error of homing increases, suggesting that olfactory cues play a role, at least in the final portion of the path (see [90] Fig. 5). It is also worth noticing that their protocol has pre-training trials, in which the rat might learn a suitable strategy to solve the problem.

3.1.15 Putting It All Together

Rodents are able to solve complex navigational tasks, including decision chain mazes, remembering recent explored options or navigating to a hidden place. To do so, they use proprioceptive (e.g. self-motion) and exteroceptive (e.g. visual) cues. Usually depriving the rodent form a subset of these cues slows down, but does not prevent learning. Meaning, as a specific case, that they can solve many of these tasks in complete darkness.

The observed rodent behavior might be due to teleological planning that takes into account what has been learned about the environment, or it might be a consequence of habitual behavior, acquired through repetition (teleological vs. habitual). Orthogonally, behavior might be due to responding to immediately available stimuli or trying to reach a desired place (place vs. response). The differences between these are subtle. A rat might be performing goal-seeking behaviors but learned a strategy based on simple responses to stimuli. The rat could stop responding once the sought goal is no longer desired (e.g. not thirsty anymore), thus confirming it as a goal seeking behavior. Moreover, one could even contemplate the existence of habitual like behaviors based on internal stimuli related to the rat location, supported by biological structures discussed in the next section.

The "teleological vs. habitual" and "place vs. response" discussions have one point in common, the transfer of control from one to the other as training continues. Meaning that rats move from teleological or place decision making to habitual or response behaviors with the pass of trials. The evolution of behavior throughout the learning process was recognized early by Small [146, 147]. He pointed out how rats make naive choices during the first trials. In later ones, they recognize key decision points by hesitating and sometimes making small runs in each direction (place and teleological). Finally, during the latest trials they make automatic, habitual like decisions (habitual and response). This relates to the findings of [134] and [62], where they discover that a place behavior (they call it place disposition) is developed during early trials, while a response behavior starts to dominate with successive training.

An interesting example involves the experiments designed by Tolman [165], where he showed the results of a variation of the discriminating box maze, originally designed to prove place behaviors. Rats were trained to choose a certain door that led to a corridor with a box with food at the end, whereas the other door lead to an electrical shock room. After the animals were over trained, they were put in the food box and given electric shocks. To Tolman's surprise, rats did not show aversion to following the corridor in the subsequent run, as he was trying to show. However, when shocked in the food box after going through the corridor, the animals showed subsequent aversion of following the said path. As Tolman claims, this evidence favors the response theory, as the rats were not able to use the learned map to compute the consequences of following the corridor. They were rather following a usual learned stimulus response sequence. It is true that over training might have played a role in the final results, though, as rats might have become too "fixated" (habituated) in their task without taking the time to plan their actions.

The issue of 'place vs response' can also be related to the distinction of reference and working memory. In the context of a radial arm maze, reference memory means remembering the right arms (to avoid going to unbaited arms), while working memory means remembering the recently visited ones (to avoid repetition). In this context, reference memory can be posed as hippocampal independent [35, 34, 175], if enough training is provided. If the observations seen in the transfer of control from place to response in the T-maze hold, it would be expected for reference memory in this context to be hippocampal dependent at first, but not anymore after retraining. At the same time, the Morris maze, used here as an example of reference memory, can be posed as working memory for protocols that switch the platform position one or few trials before testing [150, 34].

Some other captivating properties of the rodent navigation system includes the use of structured “hypothesis learning” when in an unknown environment and the capability to consolidate spatial memory after a single trial (one trial learning).

This black box analysis of the rodent navigation system poses many questions: what are the structures implementing this system? How is the map encoded? How is the rat itself encoded? How is the target represented? How is the "cognitive map" used to plan new routes? What are the advantages of ignoring it towards habitual or response behaviors? How are decisions transmitted

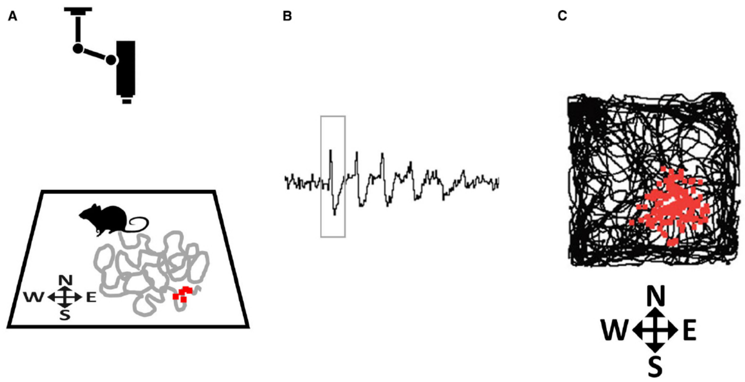


Figure 3.5: How place cells are recorded and the resulting place field. Image reproduced from [66], Creative Commons.

to the action selection center to be performed? How is the map or the route to the goal learned through experience? The following sections will try to answer them by reviewing current knowledge about rodent physiology and computational models of navigation.

3.2 Rat Brain Physiology

Extensive research has been devoted to study the biological basis of navigation. Some structures in the rat brain that have been linked to navigation are reviewed below.

3.2.1 Hippocampus and Place Cells

The hippocampus (HPC) is a structure at the base of the temporal lobes. Experiments carried out by Morris [100] showed that lesions in the hippocampus impair the rodent's ability to learn the Morris maze.

Later, O'Keefe [115] showed that there are cells in this brain structure that fire in correlation with the rodent's position. These cells, called place cells, fire whenever the rat is inside a certain region of the environment, named place field in analogy to sensory fields. Figure 3.5 shows a schema of the recording process and its results.

Interestingly, place fields are fixed to a global frame of reference, e.g. the room. This means that the rodent brain is able to integrate information from a local frame of reference (observation and self-movement cues) to derive its localization in a global frame of reference, much like robotic SLAM systems do [161].

The existence of these cells provides a new angle to the locale vs. response behavior, as discussed in the previous section. The fact that the rodent has information of its location in a global frame of reference at all times creates the possibility that some place behaviors are actually learned associations between the activity of these cells and a certain action, namely response behaviors rather than "place reasoning" ones.

Lastly, it is important to say that much research has been devoted to place cells and they are more complex than the simple summary described here. To name some important features (important to this dissertation):

- place cells have different scales of representation, namely some of them fire in very specific locations of the environment, whereas others can fire in region as large as half the environment [53, 87, 70, 154, 47, 32, 104]
- sometimes place cells are modulated by the heading direction of the rat, although this has been argued to happen only in corridor like environments [107, 137, 18, 93]
- place cells have been reported to fire only in a sub-phase of the task when it has well delimited sub-phases, which suggests that the hippocampus might maintain several maps (multiple map hypothesis, [129])
- place cells have been reported to be anchored to frames of reference other than the room, e.g. the goal location, the home location or the maze [140]
- place cells fire in coordination with theta cycles[1] and their peak firing rate advances in the theta cycle as the rat moves through the place field [144], generating a wave of place field firing from the place fields behind the rat to the ones in front of it
- under certain conditions, the firing of place cells is no longer correlated with the current position, but rather to where it has been (re-play), or where it is going to be (pre-play and spiking sweeps). This happens both in awake and sleep states [136, 169, 36, 68, 123].

3.2.2 Papez Circuit and Head-Direction Cells

An interesting type of cells are found in several parts of the rat brain, called head direction cells (HD cells). These cells fire when the rat's head is oriented in a certain angle, with respect to a global frame of reference [157]. Figure 3.6c shows an idealized response curve of the firing rate of a cell with respect to the rodent's head orientation.

[1] A pattern of activity observed in the hippocampus during locomotion and other active behaviors. It is characterized as an oscillation of approximately 8hz.

HD cells are found in several areas of what is called the Papez circuit. They are found in subiculum (Sub), thalamic nucleus (TN), mammillary nucleus (MN), retrosplenial cortex (RC) and entorhinal cortex (EC) [158].

The existence of head-direction cells further proves that rodents have localization information readily available for behavioral control. It is important to note how this global localization information is derived from local information only, the same way place cells firing is derived.

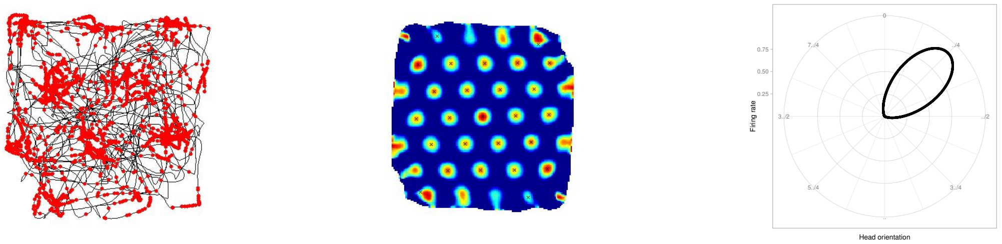
Head-direction cells show some interesting properties:

- they present different scales of representation or angular specificity [158]
- they tend to shift preferred orientations when salient landmarks are rotated in the environment [159]
- they keep firing even under visual stimulus deprivation (e.g. lights off), although the preferred location can drift [57, 157]
- they can be controlled by other sources of external stimulus, such as odor [57] and optic flow [3]

3.2.3 Entorhinal Cortex and Grid Cells

The entorhinal cortex receives input from both Sub and HPC, while projecting back to the hippocampal formation. This gives this region a suitable context for its participation in navigational information processing.

A type of cells called grid cells have been discovered in this region. These cells fire when the rat is located in any vertex of a regular grid fixed to a global frame of reference [60]. Figure 3.6a shows the spiking behavior of a grid cell, recorded in the same way as explained in Figure 3.5. Auto correlation figures (autocorrelograms), as the one shown in Figure 3.6b, are built in a two-step process. First, a rate map is built by dividing the environment in a 2D grid and counting the amount of spikes produced by the cell while the head position was in each cell. Then, the rate map is shifted by an (x, y) vector and a correlation coefficient between both images is computed. The (x, y) pixel of the autocorrelogram represents this computed coefficient. Then, if the cell present spatial symmetric bumps of activity, the autocorrelogram should show periodic peaks in the auto correlation coefficient.



(a) (b) (c)

Figure 3.6: Grid cells. (a) Recording of a grid cell. The black line shows the rat's path and each red dot correspond to a spike. (b) Auto correlation plot for a grid cell. (c) An idealized head-direction cell response curve. φ is the head-direction and r the firing rate. Images (a) and (b) reproduced from [60], Creative Commons.

Grid cells have been attributed as the metric path integrator in the rat, as they allow the rat to detect relative displacement as it navigates through those vertices.

These cells show interesting properties, similar to those of place cells and HD cells:

- they show different scales of representation or specificity [60, 17]
- the regular grid is often anchored to external landmarks [102]
- they keep firing upon removal of salient landmarks [60]

3.2.4 Basal Ganglia and Actions

The basal ganglia (BG) is a convergence center, gathering input from many parts of the brain. In turn, they influence the upper motor neurons in the cortex that ultimately control movement. Thus, the BG is suitable for action selection and learning. Any kind of information like visual, olfactory or place information can be associated with a proper action to perform in the basal ganglia [128]. The BG works by receiving sensory input in a proper structure named striatum. The striatum then projects to another structure called the globus pallidus (GP). The GP spontaneously inhibits the motor cortex, preventing movements from being made. When striatal cells (spiny neurons) spike, they inhibit cells in the GP, thus "liberating" a certain movement or higher level behavior.

The reinforcement learning theory attributes the BG as the neural implementation of the temporal difference algorithm [155, 15, 132]. The striatum receives input from dopaminergic centers, the ventral tegmental area (VTA) and the substantia nigra (SN). The dopaminergic neurons phasic firing has been attributed to be an error signal between the expected and the encountered reward [155]. The theory states that this error signal is able to modulate the synapse strength between stimulus neurons and the appropriate motor response, promoting rewarding behaviors in the long run.

This structure has been also involved in the teleological vs. habitual behavior dilemma, as the striatum has been linked to both types of behaviors. When the dorso-lateral portion of the striatum is lesioned, habitual behaviors are inhibited. However, when the dorso-medial portion is lesioned, teleological ones are inhibited [120]. Despite this, it is important to keep in mind that it is often tricky to distinguish teleological from habitual behaviors. Specially when trying to discern model-free teleological behaviors, which are still sensitive to reward depreciation but with slow relearn rate, from habitual ones, not affected by reward-depreciation.

It is interesting that this decision-making structure receives input from the hippocampus through the subiculum [58]. This allows for the implementation of place behaviors, or response to place stimuli, where the animal's response is a function of the current place, grid and head-direction cell activity. Devan [33] showed that lesions to the medial part of the striatum had similar effects to lesions to the fimbria-fornix (an input structure to the hippocampus), in that rats were impaired in place behavior learning and showed a bias towards response behaviors.

3.2.5 Ventral Tegmental Area and Reward

The ventral tegmental area (VTA) is one of the two dopaminergic centers of the brain, the other one being the substantia nigra. It is directly connected with the striatum, which receives input from the hippocampus [48, 85]. This connectivity pattern creates a suitable mechanism for error signal computation. Houk and collaborators [63] proposed a model of the interaction between cortical structures, the striatum in the basal ganglia and the dopaminergic neurons in the VTA and SN. Cortical inputs excite the striatum, providing context dependent activity. The VTA receives input from the lateral hypothalamus, providing it with information on primary reinforcements (wired satisfactory feelings, such as eating). A reciprocal connection between VTA and striatum, including an indirect path through the sub-thalamic nucleus (ST), allows the system to learn to inhibit the response to the primary reinforcement and to propagate the dopaminergic signal back in time. Thus, the system learns to respond to stimuli that lead to reward, rather than the reward

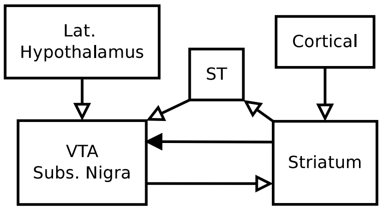


Figure 3.7: Actor critic model of the basal ganglia. Black arrows denote inhibition, while white ones denote excitation, keeping the original style [63].

itself. This facilitates learning what actions lead to rewards in the long term. Figure 3.7 summarizes the model.

3.2.6 Pre-frontal Cortex, Ventral Striatum and Vicarious Trial and Error

When rats are solving mazes, they are known to perform hesitation-like movements, called vicarious trial and error (VTE; [164]). The pre-frontal cortex (PC), as well as the ventral striatum, have been linked to value estimation functions during these events [153]. Recording from both structures show cells that change their firing upon stimuli that predicts reward. However, there are differences in their timing, suggesting that ventral striatum value estimation plays a role in decision making while the prefrontal cortex plays a role in decision evaluation [152].

3.2.7 Putting It All Together

Figure 3.8 shows the structures discussed so far. The EC-HPC-Sub loop would provide the system with spatial location information, providing metrical, place and orientation information, respectively. The subiculum acts as a relay for these centers. State information is communicated directly to the striatum, where it is used both for action selection (through the GP and MC) and for value estimation. The value estimation circuit includes the hypothalamus for primary reinforcement signals, the dopaminergic centers VTA and SN, the striatum and ST for value estimation learning. The PC-striatum connection plays a role in value estimation also, and both structures are involved in value estimation during VTE events.

3.3 Models of Rodent Navigational Abilities

Many models of the rat navigation system have been proposed; we review here a selection of them. We restrict to models using the sources of information known to be present in the brain, such as place cells, grid cells, HD cells.

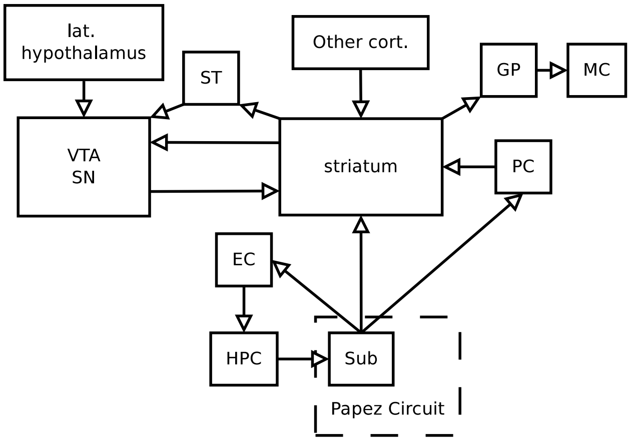


Figure 3.8: The structures and their connectivity of the rodent navigation system. Arrows represent directed connections and no distinction is made between excitatory and inhibitory connections. The lateral hypothalamus (lat. hypothalamus) connects to dopaminergic centers, the ventral tegmental area (VTA) and substantia nigra (SN). A bidirectional connection exists between these centers and the Striatum, including an indirect loop involving the SubThalamic Nucleus (ST). The Entorhinal Cortex (EC), the Hippocampus (HPC) and Subiculum (Sub) form a closed loop (HPC-EC arrow not shown). The subiculum forms part of the Papez circuit, which is related to head-direction cells. The subiculum also acts as an output relay for this closed loop, communicating information to the striatum and prefrontal cortex (PC). The PC, as well as other cortical areas, connects in turn to the striatum. The striatum outputs to the globus pallidus (GP), which in turns connects to the motor cortex (MC) to generate motion.

Each model addresses different aspects of the rodent navigation system and its capabilities. For example, some address the problem of how to learn to reach a platform as in Morris experiments, others focus on the integration of different memory systems. In this review, we group models with respect to the problems they address, keeping a correspondence to those outlined in section 3.1.

3.3.1 Motivation

Few models contemplate the role of motivation. Some models implement motivation modules that mediate both the level of reward obtained upon satisfaction (the more hungry, the more rewarding food becomes, [10, 59, 12, 11]) and the action selection process (if hungry, go towards food and not water, [59]). Gaussier [52] recognized the need for a motivation drive when modeling rodent behavior. When a goal is reached, a link between a cell tuned to the current need (e.g. thirsty cell) and the currently active place cells is formed by hebbian learning. Then, when the need arises again, this link can be used to define the goal in the stored map. Some other models considers the current goal the animat is focused on to learn and select actions, modulating the recall process [4].

3.3.2 Complex Decision Chain Learning

Few models cast the problem of navigation as a set of discrete decisions. Models that determine their state based on affordances [12, 10, 11, 59] might be seen as casting the problem as a series of decisions to be made only when affordances change. It seems that it is not clear where or how rodents would store the recent made decisions and how that would affect the following ones.

3.3.3 Behavior Stereotyping

Related to the previous subsection, none of the reviewed models includes the full transfer of control to automatic like behaviors, or the stereotyping of them. This might be due to the fact that kinesthetic cues are not used in the decision process directly by any model, although they are used to derive secondary forms of information, such as path integration.

Dollé [39] models the transfer between place to response behaviors, where the response ones also learn over time. This could facilitate the perfection and stabilization of behavior over time, but it would always depend on visual cues. Other models that learn mappings from place cells to actions [4, 16, 20, 19, 86] tend to stabilize paths into stereotyped routes even though they do not include proprioceptive cues into account in their action selection process. However, it could be argued that behavior stereotyping needs for the rodent to become "fully disconnected" from exteroceptive cues for the automation to work, providing a tighter (faster) loop of control.

3.3.4 Working Memory

None of the reviewed models approach the problem of working memory, as this seems to be a field yet to be developed. A few work with more than one goal and keep track of the last visited ones [4], although they do so without modeling how would this process be implemented in the rat.

3.3.5 Reference Memory

This is one of the most studied problems. Most models make use of place cell (and sometimes grid cells or HD cells) to decide which action to perform to reach the desired place. We review here the main mechanisms used to map places to actions.

3.3.5.1 Place Cells Driving Motor Units

Brown and Sharp’s model [16] uses place cell output to inhibit different subpopulations of motor units in the nucleus accumbens. Each place cell totally inhibits all but two cells in the nucleus accumbens, one for turning right and one for turning left. This tie is then broken by the head-direction cells. The reference memory is then built by adjusting the synaptic efficacy of these connections upon getting a reward.

3.3.5.2 Place Cells as Radial Basis Functions

Many models use place cell firing as radial-basis functions (RBF, [20, 19, 4, 86]). It consists of considering the firing of a cell i as $\Phi(x - c_i)$, where x is the animat position and c_i the preferred center of cell i. Then, the radial basis functions can be used to approximate any function as explained in Eq. 3.1, where the weights λ_i can be learned. The function $f(x)$ could, for example, approximate the proper action to take to get to a certain goal.

$$f(x) = \sum_i \lambda_i \Phi(x - c_i) \tag{3.1}$$

These types of models lack an explicit representation of a cognitive map and their actions are a response behavior, where the stimulus is the firing of the place cell population. Their behavior is not teleological in the sense they do not usually account for the consequences of their actions (model-free), although they would show sensitivity to goal devaluation as they are always learning.

3.3.5.3 Place Cells as Localization System

Some models use place cell activation to infer the localization of the animal. Then, they assume high-level mechanisms of navigation to reach a desired goal, based on that localization. Touretzky's [167] model explains existing experimental data as a function of the simulated behavior of their place, head-direction and map integration modules.

These models trade-off the modeling detail of the underlying structures (e.g. place cells) for a less accurate behavior model. The model place cell phenomena that is not found in other models, but do not focus on how they explain behavior and how other components, such as taxic behaviors, could explain the data. They also assume a highly informed teleological animal.

3.3.5.4 Place Cells as the Cognitive Map

Some models instantiate hippocampal pyramidal cells as places in the cognitive map. They usually cast the hippocampus as the system implementing the process of planning over that map.

Gaussier's [52] model considers CA3 as a transition-learning layer. Actions are selected by propagating activity from the goal backwards through transitions, until the current node is reached. Then, the action that was associated with the last transition is executed.

3.3.5.5 Place Cells to World Graph

A structure called the World Graph is often used as the implementation of the cognitive map. It consists of a graph which nodes represent locations, or local views, and edges represent the navigability between them. This graph can be used in two different ways. It can be used to learn the proper action to execute when in each node, in a model-free fashion. It can also be used as a model of the environment for planning purposes, e.g. find the shortest path to the goal. Some models make use of a world graph to implement their locale behaviors in a model-free fashion [10, 59]. Dollé's [39], on the other hand, makes use of a world graph to perform pure teleological path planning.

Milford and coworkers' [95, 94] model uses a concept similar to the world graph called the experience map, which they later on use for planning.

3.3.5.6 Place Cells as Simulators

Chersi and Pezzulo's model [23] relies on the "simulation" capabilities of the hippocampus to plan ahead, selecting the action that maximized expected reward during simulation. Erdem and Hasselmo [46, 45] use the "spiking sweeps" detected at choice points [68] to probe the path to follow. Place cells associated with goal locations allow for the detection of a successful probe. In addition, place cells serve as a value map for the environment. Johnson and Redish's model [67] simulates experiences during sleep to increase the amount of training of a model-free reinforcement learning algorithm. Koene's model [77] associate the value of a place by linking place cells to amygdala cells. Then, simulations of potential paths in the hippocampus allow the animal to recognize routes to desirable places.

3.3.6 Directional Navigation

Many models assume that rodents have the ability to navigate to a certain direction expressed in an extra maze reference frame. Dollé's model [39] implements a taxon expert that is configurable to navigate using allocentric or egocentric angles. Other models learn allothetic directions of navigation and assume the animal is able to execute those commands [4, 10].

Few others, however, make use of egocentric actions to solve navigation [16, 86, 52]. This poses a harder problem, as there is now a new dimension in the state space. Moreover, Brown and Sharp [16] showed that the problem of which direction to turn when in a given place becomes linearly inseparable if disjoint place and direction variables are used as input. This means that a task like that of the Morris hidden platform cannot be solved by a one-layer feed-forward neural network with place and head-direction cells as the input. Brown and Sharp [16] add that the inclusion of a hidden layer would solve this problem in detriment of the generalization capabilities of the learning system, and that generalization of both place and direction is not possible. Llofriu et al. [86] model combines the information in this way (in a "hidden layer"), but makes use of multiple scales of representation in both place and head-direction cells to achieve generalization capabilities in the model. Gaussier's [52] model detects transitions between place cells and learns what egocentric action to perform in each transition.

3.3.7 Shortcuts and Novel Routes

One simple mechanism that generalizes actions by construction is the use of larger scales of representation [4, 86, 16], as the actions are learned over place cells that cover a big portion of the environment. This allows the model to know what action to perform when in a novel place.

Other models use the hippocampus as an online "simulator" for the outcome of potential actions [23, 45, 46]. This allows the model to find shortcuts when conditions in the environment change. However, it is not clear how information about changes in the environment (e.g. placing an obstacle) would change the dynamics of the simulations.

Erdem and Hasselmo's model [45] is applied to a set of known rat experiments with interesting results. The Tolman maze described in subsection “Shortcuts and Novel Routes” is solved by the model, as well as a shortcut experimental platform called “Hairpin maze” [1]. The use of the underlying metrical structure, grid cells, for the linear look-ups facilitate the discovery of new shortcuts when obstacles are removed. It is not clear however how the perception of the open or closed spaces would modify the dynamics of the "spiking sweeps" used by the model in real rats.

3.3.8 Sensory Modalities in Learning

Different modalities are used by different models. Some models use distance and bearing to a landmark to influence place cell firing [20, 167, 59, 39, 10, 16]. Others use the distance to a certain wall [19, 167, 59], accounting for the fact that changing the shape of the environment changes their firing. The visual distance, or angle, between two landmarks as a way to distinguish a local view is used as well [167]. Arleo's model [4] uses a Gabor filter to perform feature detection over quasi-panoramic images.

When implementing a working model, one is faced with the problem of how to chose a set of potential actions to perform. It would be irrational for a model to consider the possible outcome of executing an impossible action, like going forward through a wall. Guazelli [59] formalized this concept as affordances. They implement perceptual schemas [2] that process the visual and tactile information to compute a set of possible actions to perform, e.g. go forward, turn, eat. Affordances could be considered an abstraction of sensory information, or a meta-sensor, in this way. In Barrera et al. model [10, 12, 11], affordances play a role in the animat localization, as they influence creation and recognition of nodes in the world graph.

3.3.9 Teleological vs. Habitual Actions

Based on Dickinson's [37] hypothesis, rodents are capable of both types of responses, teleological and habitual. None of the reviewed work models the generation of habitual behaviors, namely ones that do not respond to reward depreciation. Some, however, implement model-free reinforcement learning [155], which would show slower extinction times. Other more teleological approaches use model-based reinforcement learning. Finally, the most teleological ones use models of the world to perform deliberative planning on how to get to the goal, which we call planning models.

3.3.9.1 Reinforcement-modulated Hebbian Learning

Brown and Sharp's model [16] casts the problem of learning to the instrumental learning paradigm. Under this paradigm, subjects learn to solve a problem by increasing the likelihood of repeating actions or behaviors that led to reward. They implemented this by increasing the synaptic efficacy between hippocampal and nucleus accumbens cells upon receiving a reward. Anticipating the reinforcement learning models that arose in the following years, they also incorporated a notion of cumulative synaptic activity, which was equivalent to the use of eligibility traces.

3.3.9.2 Model Free Reinforcement Learning

Regardless of the underlying structure, many models make use of a model-free reinforcement learning schema [155] to model the adapting part of the system. This lies on the edge between teleological and habitual actions, as the system would eventually be able to relearn upon reward depreciation. However, it would not do it immediately due to the lack of planning. The learning times of these systems tend to be slow, which resembles more habit generation than flexible place behavior learning.

Burgess et al. [20, 19] use a reinforcement signal to activate synapses between subicular cells and “goal cells” that allow the animal to compute the heading to the goal later on. Guazzelli and collaborators’ model [59] uses two actor critic modules to learn the proper action to execute, one using visual stimuli and the other based on a cognitive map. Other models perform reinforcement learning directly over the place cell output, changing the weight between them and actor units as a function of reward [4, 20, 19, 86]. Barrera et al. model [10] uses an actor critic architecture attributed to the HPC-VTA-Striatum interaction.

These models focus on the idea that the hippocampus provides nothing more than localization information, rather than a complete map where planing can take place. As stated by Guazelli [59]: *“Nevertheless, there is still no evidence that the hippocampus proper (dentate gyrus and Ammon’s horn) can simultaneously encode the rat’s current location and the goal of current navigation. In other words, the hippocampus may provide the "you are here" function of a map but not "this is where you are going" and "this is how to get there" functions, which thus must depend on a larger system of which the hippocampus proper is but one part.”*

Adding to this, eligibility traces significantly speed up learning [155]. Many models make use of eligibility traces when updating synapses in a reinforcement learning fashion [59, 10, 86, 12, 11, 4].

3.3.9.3 Model-Based Reinforcement Learning

Chersi and Pezzullo [23] proposed a navigation system that uses the hippocampal forward sweeps to simulate each possible path in a T-maze. Each sweep stimulates cells in the nucleus accumbens that provide a notion of learned value of the given place. Then, a total value for each possible path can be estimated, and the path with the greater value can be chosen. Johnson and Redish’s [67] work updates inter-hippocampal synapse strength as a result of simultaneous firing. The resulting connectivity is then used to simulate replay events and apply reinforcement learning, in a model-based approach.

Model-based RL paradigms explain how a subject could react to a distant goal depreciation in one trial. The depreciation would be "detected" while planning using the built model, or various sweeps of learning could facilitate faster consolidation.

3.3.9.4 Planning

Guassier’s model [52] performs purely teleological actions, where a plan to a goal is found by searching in a learned topological cognitive map. Similarly, Milford’s [94] model constructs a map of “experiences”, which link places with their odometric displacement, and uses it to perform teleological planning. Their model will persistently try to follow a planned path until an occluded link is unlearned (highly deliberative).

Koene [77] proposed a model in which the hippocampus is capable of recognizing distant places. Then, the amygdala responds to these places by retrieving their value, within a place conditioning framework. Finally, the retrieved value is used to select the best alternative in a pure teleological way.

Erdem and Hasselmo’s [45] model performs linear look-ups in different directions to assess each one’s potential to lead to the goal. In addition, a reward diffusion mechanism helps the rat in situations in which the goal is not within the reach of the linear look-ups, much like value functions work in reinforcement learning. Erdem and Hasselmo’s model [46] builds on this previous one by including different scales of representation. Larger scales significantly increase the reach of the look-up process.

3.3.9.5 Combination of Paradigms

Dollé et al. model [39] combines reinforcement learning taxon behaviors with teleological cognitive-map like behaviors. They show how they cooperate and compete, depending on the situation.

3.3.10 Latent Learning

Few models contemplate latent learning. Some authors interpret latent learning as the stabilization of the place and head-direction cell system [20, 19, 4]. By settling these signals, the animat gains a representation of the hidden state, making learning a goal-oriented task easier. However, latent learning experiments are carried out during intervals much longer than place cell stabilization delays. Thus, this would only explain a portion of the latent learning phenomena.

Other models generate topological maps of the environment during exploration, which are used for planing later on [39, 52].

Similarly, some models learn transition probabilities between places, which they later use to simulate the route to take [23], or new routes to apply learning to [67].

3.3.11 Place vs. Response

Many models ignore the response aspects of behavior and focus on modeling place behaviors [23, 52, 4, 12, 10, 11, 45, 46, 16].

Other models place their focus on the interaction between place and response behaviors. Guazelli [59] proposed a model that reproduced T-maze reversal results, where the transition between the learned turn and the new turn occurs gradually for animals with hippocampal input deactivated [113]. They combined a response (taxon) and a locale module (world graph layer) in a cooperative way, and showed how their model reproduces existing results [113]. Dollé's model [39] also implements two different modules, the taxon and place "experts", which are switched by a reinforcement-learning driven gateway. They apply the model to Pearce's experiments [121], showing how inhibiting the place expert improved performance at the beginning of each session but prevented intra-session improvements. The model is also applied to Devan's experiments [33], obtaining similar results in the competition trials.

Some models account for the taxic component in the tasks they simulate, but in less detail [86, 77]. They implement cooperative response and place behaviors, but only the place component is able to learn during trials.

It is interesting how Brown and Sharp [16] cast the problem of the Morris arena as a Stimulus-Response (S-R) problem, or instrumental learning. Thus, they implement a response system, but where the stimulus consists of place cell output. They show that place-like behaviors, such as showing preference for the training quadrant in Morris probe trials, can be explained by simple learned responses to place cell activities. From now on these types of models will be referred as reactive place models, in contraposition with deliberative (more teleological) place behaviors.

3.3.12 Hypothesis Testing and Structured Exploration

The implementation of a working learning model requires a process of exploration. Some models implemented a curiosity level associated with each node of a world graph [59, 10]. This assigned an innate drive to visit nodes not yet visited. However, this solution does not scale to scenarios where possible actions are many or infinite, or when the animal does not have a model of the consequences of executing each action (model free). Many other models use regular reinforcement learning exploration schemas, where actions are randomly picked [4, 86, 16, 45, 46].

In Arleo's model [4], however, they implement an initial exploratory phase where the animal is aware of its path integration error, returning to the home location to correct it. Arleo's model [4] achieves an animal-like exploratory behavior that performs small routes around the home location at first, to engage on larger exploratory journeys later. None of the other reviewed work models the presence of innate structured exploratory behaviors and the advantage they pose to the rodent navigation system.

Milford and collaborators [94] use the map to guide exploration, so as to maximize the coverage of the exploration process.

In models that combine taxon behaviors with their place ones [39, 77, 59], exploration is partially visually guided.

Burgess et al. [20, 19] uses random exploration, but biases forward motions by constraining the next angle to an interval centered in the current one.

3.3.13 One-Trial Learning

One-Trial learning is one of the least explained phenomena in the reviewed models. Burgess et al. [20, 19] has a “one-shot learning” property that allows it to reach a goal after finding it once. However, the way they update their synapses would not allow for the learning of a new location.

Dollé’s model [39] applied to Pearce’s [121] experiments showed how their place expert was able to significantly improve performance within the span of one session (4 trials), although the task consisted on a cued version of the Morris maze. More interestingly, they showed how an allocentric angle taxon expert is responsible for inter-session improvement, as the rat learns how to approach the platform near the landmark (taxon expert) and when to do so (switching gateway). When translated to the non-cued Morris task, this could suggest the existence of a component in the rat navigation system that is able to progressively learn how to reach a certain recently remembered location. This would explain why rats improve their time to escape with sessions, even though the information from previous sessions is useless in the new ones.

One-Trial learning could also be explained by models that learn transitions in space [23], as learning the reward location for the first time would allow the animal to recognize the value of that place in later simulations. Jhonson’s model [67] could also explain one-trial learning if the hippocampal connectivity is changed enough so new simulations during rest resemble only the new scenario.

The forward linear look-up models [46, 45] could also explain the one-trial learning phenomena, as they associate places with goals the first time they encounter them.

3.3.14 Path Integration and Homing

Arleo's model [4] was one of the first of the reviewed models to include path integration into the navigation decision process. It did it in two different ways, on one hand place cells were partially driven by path integration cell populations, which in turn drove navigation decisions. In addition, when in new environments, the animat used the path integrator to find its way back home on a regular basis to ensure a consistent map was built. Likewise, Milford's model [94] uses odometry to move the activity bump of a continuous attractor implemented by the hippocampus, which could drive place cell activity in complete darkness. They also use this map to build an "experience map" [95], which could implement homing behaviors, although this was not tested. Barrera's model [10] implements a "dynamic remapping module", attributed to the posterior parietal cortex, that keeps track of the rat displacement from the point of entry. This information was in turn used to maintain a path integration module, which contributed to place cell firing.

Although they do not explicitly test this, Erdem and Hasselmo [46, 45] model would explain homing behaviors in the dark, as both grid and place cell firing patterns would remain stable and look-ups would allow the rat to find its way home.

3.3.15 Putting It All Together

All of the reviewed models focus on the reference memory aspect of navigation. Three main approaches are used: model-free reinforcement learning, model-based reinforcement learning and planning. Model-based and planning approaches can explain more sophisticated abilities of rodents, like one-trial learning, faster sensitivity to goal devaluation and teleological behaviors. However, model-free approaches usually explain the decision making process in greater detail and rely on fewer assumptions. None of the reviewed work combines these two, possibly parallel, systems of navigation. The combination of the two could shed light in the "teleological vs habitual" dichotomy.

Most models make use of exteroceptive cues and path integration. None of the reviewed models uses kinesthetic cues directly in the decision making process, so they are not able to fully explain phenomena like behavior stereotyping or complex maze solution in the darkness. No model focuses on the working memory aspect of navigation either.

Exploration is another aspect that can be improved. Many models rely on random walk processes to explore, which is not what normal rodents do. Few models focus on the interaction of parallel place and response (taxic) behaviors, which is something that could improve initial exploratory behaviors.

Tables 3.1 and 3.2 summarize the reviewed models' main features.

Table 3.1: Main characteristics of the reviewed models. Abbreviations: Not Applicable (N/A), Action Selection (AS), Reward Modulation (RM), Multiple Maps (MM), Landmark (LM), Distance (D), Bearing (B), Affordances (Aff), Path Integration (PI), Wall Distance (WD).

Model	Motivation	Working Mem.	Reference Mem.	Directional Nav.	Shortcuts	Modalities
Burgess et al. [20, 19]	N/A	N/A	PC as RBF	Allocentric	N/A	LM D&B, WD
Brown and Sharp[16]	N/A	N/A	PC to Motor	Egocentric	N/A	LM D&B
Touretzky and Redish[167]	N/A	N/A	PC as Localization	Allocentric	N/A	LM D&B, WD
Guazelli et al. [59]	AS & RM	N/A	PC to World Graph	Allocentric	N/A	LM D&B, Aff.
Gaussier et al. [52]	AS	N/A	PC as Cogn. Map	Egocentric	N/A	LM B
Arleo et al. [4]	N/A	MM	PC as RBF	Allocentric	Larger Scales	Gabor Filter
Johnson and Redish [67]	N/A	N/A	PC as Simulator	Allocentric	N/A	N/A
Milford et al. [95, 94]	N/A	N/A	PC to World Graph	Allocentric	N/A	Histograms
Barrera et al. [12, 10, 11]	RM	N/A	PC to World Graph	Allocentric	N/A	LM D&B, Aff
Koene and Prescott [77]	N/A	N/A	PC as Simulator	Allocentric	N/A	LM Angles, Touch
Dollé et al. [39]	N/A	N/A	PC to World Graph	Both	N/A	LM D&B
Chersi and Pezzulo [23]	N/A	N/A	PC as Simulator	Egocentric	Look-ups	N/A
Erdem et al. [45, 46]	N/A	N/A	PC as Simulator	Allocentric	Look-ups	Only PI
Llofriu et al. [86]	N/A	MM	PC as RBF	Egocentric	Larger Scales	LM D&B

Table 3.2: Main characteristics of the reviewed models (cont). Abbreviations: Model-free (MF), Model-based (MB), Reinforcement learning (RL), Planning (Pl), Place Only (PO), Response (Rsp), Biased Random (BR), Random Exploration (RE), Curiosity Based Exploration (CBE), Taxon Guided (TG), Path Integration Guided (PIG), Place Guided (PG), Place Cell Firing (PCF), Exploration (Expl), Homing (Hm), Grid Cell Firing (GCF).

Model	Teleo./Habitual	Place/Resp.	Exploration	One-trial Learn.	PI and Homing
Burgess et al. [20, 19]	MF RL	PO	BR	One shot learning	None
Brown and Sharp[16]	MF RL	PO	RE	No	None
Touretzky and Redish[167]	N/A	PO	N/A	N/A	None
Guazelli et al. [59]	MF RL	Place + Rsp	CBE	No	None
Gaussier et al. [52]	Planning	PO	RE	By Planning	None
Arleo et al. [4]	MF RL	PO	PIG	No	PCF & Expl
Johnson and Redish [67]	MB RL	PO	RE	By Simulation	PCF
Milford et al. [95, 94]	Planning	PO	PG	By Planning	PCF
Barrera et al. [12, 10, 11]	MF RL	PO	CBE	No	PCF
Koene and Prescott [77]	Planning	Place + Static Rsp	TG	By Planning	PCF & Hm
Dollé et al. [39]	MF RL + Pl	Place + Rsp	RE	In visible platform	PCF
Chersi and Pezzulo [23]	MB RL	PO	RE	By simulation	PCF
Erdem et al. [45, 46]	Pl	PO	RE	Using look-ups	PCF
Llofriu et al. [86]	MF RL	Place + Static Rsp	RE	No	PCF

CHAPTER 4

MULTI-SCALE FRAMEWORK

4.1 Introduction

This chapter[1] introduces the main concepts involved in this dissertation. The ideas presented here are shared by the models and algorithms introduced in the next four chapters.

Based on chapters 3 and 2, this chapter presents how the problem of navigation is posed, which branch of planning this dissertation focuses on and how learning is modeled. Further chapters will build on this general framework, introducing particular details of the models or algorithms being described.

4.2 Path Planning and Learning

There are many ways an agent can learn to improve its path planning efficiency (see chapter 2). This dissertation focuses on agents that have limited information about the environment, but good localization. Thus, it will deal with sensor driven path planners, e.g. Artificial Potential Fields and Bug-like algorithms.

Sensor driven path planners usually model navigation as a function that maps the locations of the agent and goal, and the sensors information, to the next action to perform $a \in A$, as shown in Eq. 4.1. Where A is the set of possible actions, m is the dimension of the state space, n is the dimension of the sensor information, l_{agent} is the location of the agent and l_{goal} is the location of the goal.

$$a : \Re^m \times \Re^m \times \Re^n \rightarrow A :: a(l_{agent}, l_{goal}, s) = a \tag{4.1}$$

Many learning algorithms can learn such function, under different assumption. This dissertation mainly focuses on the use of reinforcement learning, its main advantage being it allows learning

[1] Portions of this chapter have been previously published in Neural Networks, 2015, 72: 62-74, and have been reproduced with permission from Elsevier

with spurious feedback. The last chapter, though, uses roadmap concepts to improve the route by building a graph of the environment.

When applying reinforcement learning to a navigation problem, part of the state space becomes the possible locations of the agent. This makes the state space continuous, introducing a gap between the problem and tabular RL solutions. The next sections describe the approach used to bridge this gap, taking inspiration from (and modeling) place cells.

4.3 Place Cell Driven Learning

Place cells are a good source of information for localization [174]. We adapted the classical RL algorithm in order to use a multi-scale space representation as we describe in greater detail next.

4.3.1 Place Cells as the RL State

When a rat is in a specific location within a known environment, a set of place cells fire signaling that the animal is within their place fields. Since place fields overlap [116], several cells might be firing at any given moment.

Thus, in the animal's brain, the location is encoded as the activity of an ensemble of cells. This contrasts with the intuitive representation of the state as vector holding position and orientation information. Thus, our RL algorithm input is comprised of a set of place cell activities that encode the current location [4, 22, 52, 11].

When dealing with a continuous state space, some RL solutions resort to discretization of the environment, due to its simplicity of implementation [76]. Figure 4.1 illustrates the difference between an environment discretization and the use of place cell like states. Instead of discretizing the environment into a fixed grid, each place cell is laid out over the environment, with their place fields overlapping. Then, the place cells ensemble activity will encode the location of the robot at any point in time. This activity can then be used as the current state in RL algorithms [4].

More formally, we are considering the problem of dealing with continuous state RL, where the continuum corresponds to the position and orientation of the robot. Let the continuous state be as defined in Eq. 4.2.

$$c = (c_1, ..., c_n) \in \mathbb{R}^n \tag{4.2}$$

Note that variables c_i have a continuous domain. For example, in our case, c is composed of the position and orientation of a robot in a $2D$ plane, i.e. $c = (x, y, \theta)$.

Singh et al. [143] describe work on RL over soft states and allows for the use of the place cell ensemble activity directly as the state. The main idea behind Singh et al. soft-clusters, or soft-states, approach is to consider a new discrete set of states $S = \{s_1 \dots s_m\}$, where the states s_i are soft-clusters on the space of c, $\mathbb{R}^n$. Soft-clusters are defined with a conditional probability, where every possible value c will belong to a cluster s_j with a certain probability $p(s_j|c)$. We call the probabilities $p(s_i|c)$ the *activation value* of state s_i to emphasize the place cell metaphor and denote it as $A(s_i, c)$. Figure 4.1 shows an environment where 5 soft states are laid out. Given the robot position x, two of them show an activity $A(s_i, x)$ greater than zero, while the other three are inactive.

Notice that when the robot is in a specific continuous state c, more than one cluster may be active simultaneously, the same way as more than one place cell might be firing simultaneously.

Summarizing the notation:

- variable c represents the state in its original continuous nature, e.g. the position of a mobile robot in our navigation problem,
- variable s_i represents a state in the multi-scale framework, it corresponds to the place cell that fires in a subregion of the environment or the soft states from the work by Singh et al. [142]
- $A(s_i, c)$ is the activation of the soft-cluster s_i when the continuous state is c.

With the soft state framework in mind, Figure 4.1 can be reinterpreted as a set of soft states, s_j, laid out over the environment and their level of blue represents their activation, or $p(s_j|location)$.

The clusters will always have a decreasing activation as c gets away from a preferred value, or center, of cluster s_i. That is, the activation will be decreasing with the distance of c to the cluster center. The activation will range from 0, when c is far away from this preferred value; to 1, when c and s_i preferred value are the same. This correlates also with the way place cells fire.

In summary, we have applied the concept of place cells as a way to represent the RL state, which is made possible by Singh et al. [142] algorithm for QL using soft states. This allows us to perform QL over the activity of an ensemble of place cells or soft-states.

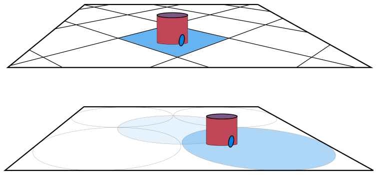


Figure 4.1: Environment discretization vs place cell states. (top) vs place cell states (bottom). The trapezoid represents the environment and the robot is shown in magenta. Each state activation is shown in blue, the darker the color, the more active the state is. In a usual discretization, only one state is fully active given the position of the robot (top), whereas activation is shared among many place cell states in the other case (bottom), two for this figure.

4.3.2 Reinforcement Learning Equations on Place Cells

The canonical Q-Learning algorithm maintains a table of the expected reward $Q(s, a)$ of performing an action a when being in a state s. In order to select an action when in state s, a greedy approach would pick the action to perform according to Eq. 4.3.

$$a = argmax_a Q(s, a) \tag{4.3}$$

In the soft state framework, the action selection becomes as described in Eq. 4.4. Note that canonical QL could be understood as having only one active soft state at a time, thus $A(s_i, x)$ would be 1 for only one s_i. Then the canonical QL action selection equation becomes a special case of Eq. 4.4. This resembles the use of place cells as radial basis functions proposed by Burgess et al. [20].

$$a = argmax_a \sum_S Q(s_i, a) * A(s_i, c) \tag{4.4}$$

Figure 4.2 contrasts the action selection process when using a fixed discretization and when using soft states.

After every action a is taken from state s and arriving to state s', tabular QL algorithms update this table according to Eq. 4.5, where α and γ correspond to learning parameters [155] and r is the obtained reward signal.

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma max_{a'} Q(s', a') - Q(s, a)) \tag{4.5}$$

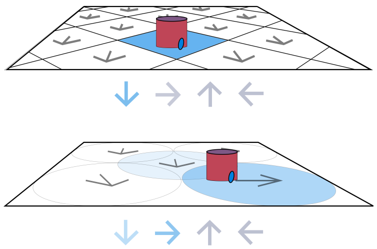


Figure 4.2: The difference in the action selection process for an environment discretization and soft states. . Each state has a preferred action in both cases. In the discretization, the only active state's preferred action is picked. With soft states, each active state contributes with its preferred action in proportion to their activation value.

In the soft state framework, all soft states s_i must be updated according to their activation value, see Eqs. 4.6 and 4.7. The maximal expected return from the successor state is replaced with a sum over all soft state's Q value for each possible action, where c' is the successor state in the continuous space. The value is normalized over the total activation to keep δ (Eq. 4.7) bounded.

$$Q(s_i, a) = A(s_i, c) * (Q(s_i, a) + \delta) + (1 - A(s_i, c)) * Q(s_i, a) \tag{4.6}$$

$$\delta = \alpha * \left(r + \gamma max_{a'} \sum_{s'} \frac{Q(s_i, a') * A(s_i, c')}{\sum_{s'} A(s_i, c')} - Q(s_i, a) \right) \tag{4.7}$$

The canonical QL update equation is a special case of this one. Since, for the canonical QL, $A(s_i, c)$ would be different from 0 for only one s_i, only the first term would apply to the active state and the second to all other states, leaving the value unmodified. The maximization over the successor state also degenerates to the canonical formula when only one state is active.

For the actor-critic algorithm used in chapter 7, the equations for the multi-state update are shown in Eqs. 4.8 - 4.10. Here the agent moves from location c to c', performing action a.

$$V(c) = \sum_{s_i} \frac{A\left(s_i, c\right) V\left(s_i\right)}{\sum_{s_i} A(s_i)} \tag{4.8}$$

$$\delta = r + V(c') - V(c) \tag{4.9}$$

$$Q(s_i, a) = Q(s_i, a) + \alpha A(s_i, c)\delta \tag{4.10}$$

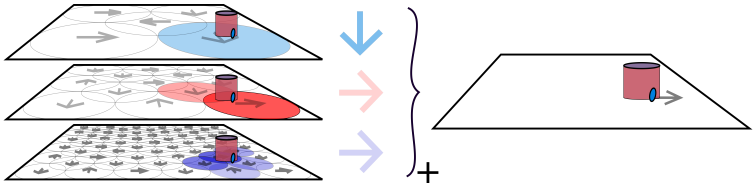


Figure 4.3: The action selection process for the multi-scale case. The active place cells on every layer contribute a desired action weighted by their activation value. All values are added together and the action with the maximum action is performed.

4.4 Multiple Scales of Place Cells

Place fields from different parts of the hippocampus have different sizes [70, 87, 73]. Septal (dorsal) place fields are smaller with higher spatial specificity, providing a fine grained spatial representation. In contrast, temporal (ventral) place fields are larger and have consequently lower spatial specificity, providing a coarse grained source of spatial information.

Because these cells fire simultaneously, they provide a redundant multi-scaled encoding of the animal location.

Figure 4.3 illustrates the multi-scale soft states representation and the action selection process.

The key of the multi-scale concept is to have soft-clusters with different degrees of selectivity, as there are different scales of place cells. Some soft states would be active in a confined range of the continuous state space while others would be active in larger regions of the continuous state space. For example, in a navigational task, some soft states would be active only within a radius of $0.1m$ of the cluster center, whereas others would be active anywhere within $1m$ of the cluster center.

At any given moment, many soft states could be active at the same time. Some of them would be more selective states, covering a small neighborhood near the current continuous state c, whereas other active states would be less selective. After an action is performed, the learning rule is applied to all soft states. Thus, the outcome of that action would be learned for states that are going to modify the behavior only locally in the future and for states that influence the behavior in larger region of the continuous state space. Thus, the agent will learn fine grained policies and coarse grained policies at the same time, combining them into a single policy when performing the action selection.

4.4.1 Asymmetric Contributions

In one of the presented models, an asymmetric contribution of different scales to learning is proposed. In the framework of the actor-critic algorithm, there are two quantities to be determined as a function of the location l: the value $V(l)$ and the action values $q(l,a)$. Based on empirical biological evidence, we propose that different scales of representation contribute with different weights to the computation of each of these quantities. Larger (coarser) scales of representation are proposed to contribute more heavily to value estimation, whereas smaller (finer) scales of representation contribute more heavily to action selection.

4.4.1.1 Algorithmic Fundamentals

From the algorithmic point of view, we propose that this distribution makes more sense than the opposite alternative and the uniform one. As discussed in further chapters, larger scales of representation contribute to generalization. We argue that the value function $V(l)$ for an optimal policy changes less across space than the action values $q(l,a)$ or, equivalently, the navigation function introduced in Eq. 4.1. Thus, it makes more sense to assign it to states that generalize more across space. An optimal action selection policy, on the other hand, might change abruptly in smaller spaces, due to the presence of obstacles that must be circumvented. Thus, it makes more sense to assign them to smaller scales of representation.

4.4.1.2 Biological Fundamentals

From a modeling point of view, it can be argued that anatomical evidence in the hippocampus suggests that dorsal and ventral levels project onto each other and exchange information internally [154]. These space representations are then projected to multiple structures including the ventral striatum and the ventral tegmental area, structures involved in reward and decision making. There is a well-known functional loop structure between the hippocampus and the ventral tegmental area (dopamine center), which could support the type of reinforcement learning used in our model [85]. Our algorithm takes advantage of the ability for large place fields to provide a global view of the environment. This scenario matches the common conception regarding the ventral striatal functionality of driving behavior on the basis of the motivational value of the environment [82]. It is also consistent with the fact that ventral hippocampus projects more strongly to ventral striatum [5, 50].

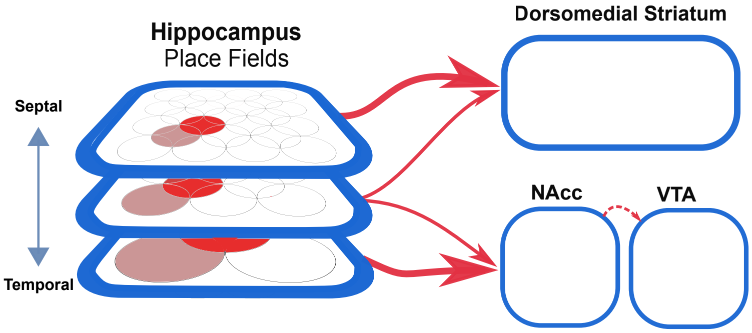


Figure 4.4: Differential connectivity from the hippocampus to the downstream structures.

When looking at this HPC-BG loop, connections from CA3-CA1-Sub remain organized in somewhat isolated circuits [54]. The subiculum projects to both ventral and dorsal striatum [58]. A distinction between dorsal an ventral striatum has been suggested in the framework of reinforcement learning and the actor critic implementation, where dorsal is associated with stimulus-response learning (actor) and ventral to value learning (critic) [7]. Moreover, the projections from the hippocampus to the dorsal striatum are to its medial portion, which has been associated with place strategies, as opposed to cue based ones in the lateral portions [33]. Figure 4.4 shows an schematic of this connectivity.

4.5 Flow of Events

Figure 4.5 shows the flow of events of a single cycle of the models and learning algorithms introduced in this dissertation. A cycle starts with the agent reasoning about its next motion and ends after it has moved and learned about the outcome of that last move. The steps are as follows:

1. The agent location is acquired. In the case of simulation, this information is provided by the simulator itself. In the case of experiments with the physical robot, this data is provided by an implemented Fast-SLAM [98, 97] system, a modified version of ORB-SLAM [110] or by a ceiling camera connected to the RoboCup SSL vision system [181].

2. The location information, in the form of (x, y, θ), is used to compute the firings of place cells and head direction cells.

3. The PC and HDC information is combined and fed to the Q-Learning/actor-critic action selection.

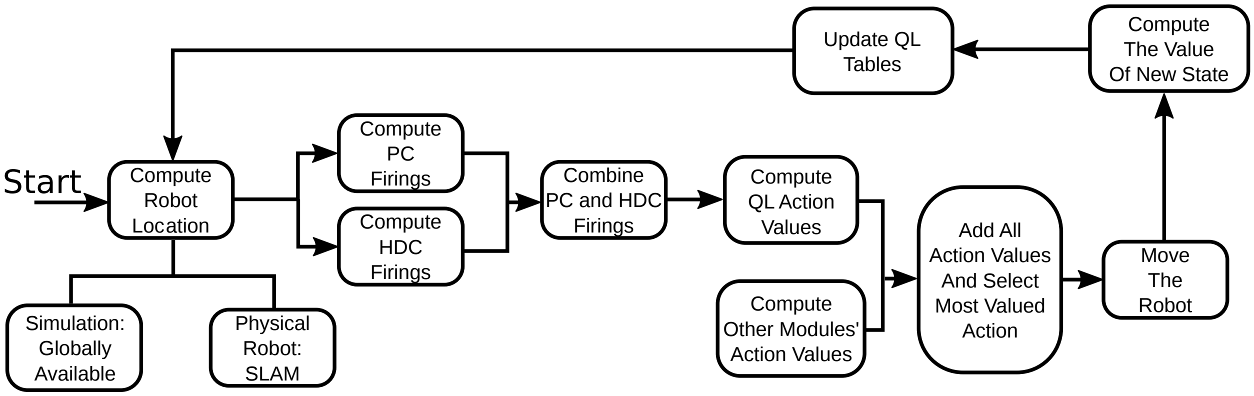

Start
Compute Robot Location
Simulation: Globally Available
Physical Robot: SLAM
Compute PC Firings
Compute HDC Firings
Combine PC and HDC Firings
Compute QL Action Values
Compute Other Modules' Action Values
Add All Action Values And Select Most Valued Action
Move The Robot
Compute The Value Of New State
Update QL Tables

Figure 4.5: Flow of events for the navigation model.

4. In parallel with 3, all other possible behaviors (e.g. sensor based behaviors, exploration behaviors) are executed. The relevant information is fed to them and they assign a value for each possible action.

5. All action values are added and the most valued action is selected.

6. The agent is moved using the selected action.

7. The arrived state is observed and QL/V tables are updated, according to Eq. 4.6 or 4.8 - 4.10. Notice that the computation of the arrived state involves observing the location, computing PC and HDC firings and combining the information again. This has been left out of the diagram for the sake of simplicity.

CHAPTER 5

MULTI-SCALE OPEN MAZE MODEL

5.1 Introduction

This chapter[1] introduces a first approach to modeling the effect that multiple scales of representation have on learning.

The model consists of three scales of place cells that serve as input to a reinforcement learning algorithm. It is tested in a Morris like experiment, with an additional cued component.

The following sections describe the model, the performed experiments and the obtained results. Then, results are discussed and conclusions are drawn.

5.2 The Model

This spatial cognition model is comprised of six main modules, described below and shown in Figure 5.1.

It has been proposed that navigation involves the interaction of four components: place cells, head direction cells, local view and path integration [130, 166]. We consider our path integration and local view components as solved. Namely, place cell firing values are derived from sources of location information directly, rather than computing them from path integration and visual information, as will be explained in the Experiments section. Thus, we focus in this work on the place cell and head direction cells components and their contribution to learning using multiple scales. Our model uses this multi-scale representation as the information source for a reward driven learning system [79].

5.2.1 Modules

5.2.1.1 Place Cell Module

This module calculates the firing of a population artificial place cells. They take the current position x of the robot as input and calculate the firing rate as Eq. 5.1, where f_i is the firing rate

[1] Portions of this chapter have been previously published in Neural Networks, 2015, 72: 62-74, and have been reproduced with permission from Elsevier

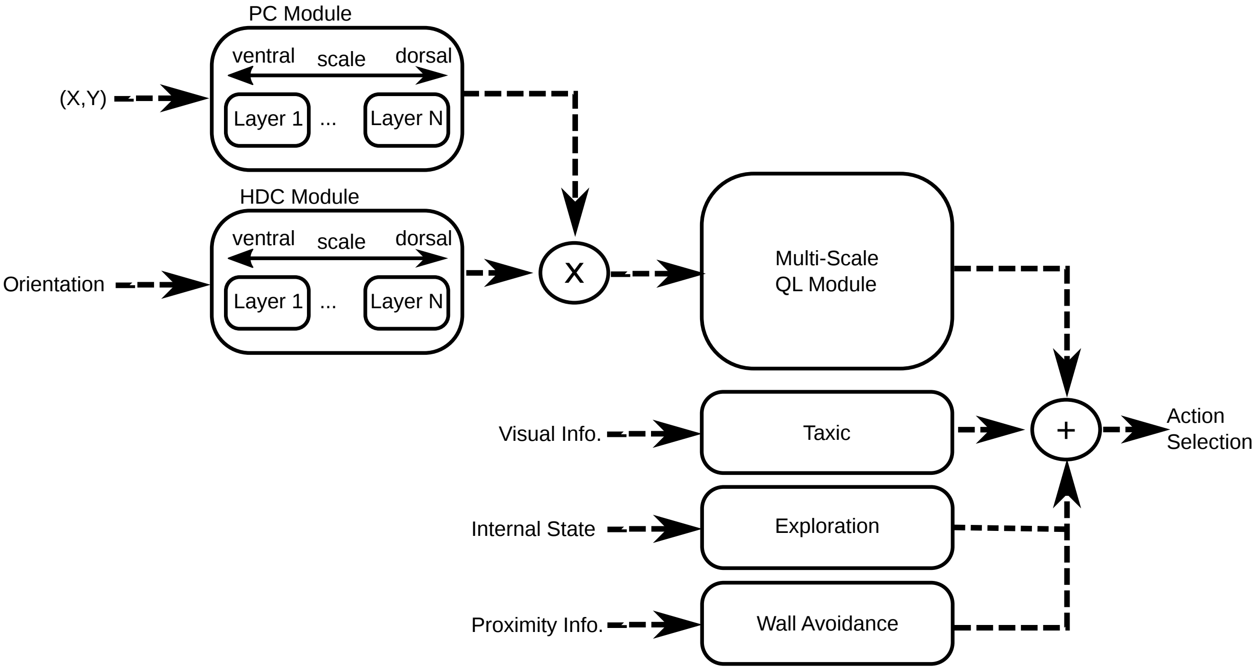


Figure 5.1: The system modules and the flow of information. The dotted arrows represent flow of information. The Place Cell Module (PC) receives location information. The Head Direction Cell Module (HDC) receives orientation information. PC and HDC information are combined and sent to the Multi-Scale Q-Learning Module, which outputs action selection values. The Taxic Module receives visual information, the Exploration Module incorporates internal state information, the Wall Avoidance Module processes proximity information.

of cell i, c_i its preferred location and Σ_i its covariance matrix. Namely, each cell fires according to a $2D$ Gaussian function with a center on each place cell preferred position, as modeled by O'Keefe and Burgess [114].

$$f_i = exp\left(-\frac{(x-c_i)^T\Sigma_i^{-1}(x-c_i)}{2}\right) \tag{5.1}$$

The key of this work involves the use of different scales of place cells, which we map to choosing different Σ. The covariances matrix are always of the form $\sigma^2 I$, where σ^2 models the specificity and I is the identity matrix.

5.2.1.2 Head Direction Module

This module computes the firing of a population of artificial head direction cells. This module takes the current heading θ of the robot and computes the firing rate of each cell as Eq. 5.2, where f_i is the firing rate of the i^{th} head direction cell, σ^2 its variance and θ_i its preferred orientation. Thus, this cells are also computed as a Gaussian function with the peak in the cell's preferred value.

The variance of head direction cells is also varied to obtain multiple scales of representation of the current heading.

$$f_i = exp\left(-\frac{(\theta - \theta_i)^2}{2\sigma^2}\right) \tag{5.2}$$

5.2.1.3 Multi-Scale QL Module

This module performs Q-Learning on the information provided by the place cells and head direction cells, as explained in chapter 4. Place and orientation information is obtained by selecting all possible pairs from both sets and computing the resulting activity as the product of both the place cell and head direction cell. This combined source of information is passed onto the QL module with a symmetric connectivity to value estimation and action selection, since QL does not separate the two.

5.2.1.4 Taxic Behavior Module

This behavior moves towards a visible goal. It works cooperatively with the QL learning module by assigning a fixed value to the action that will take the robot to the goal. In the framework proposed by Guazzelli et al. [59], this module corresponds to the execution of the affordance of going to a visible goal.

5.2.1.5 Exploration Behavior

This module promotes exploration in early phases of an experiment. The exploration value is calculated as shown in Eq. 5.3, where episode is the episode number, *maxReward* is the maximum reward possible given to the robot, and β is a given parameter that models how fast the exploration value decays. Higher values of β mean slower decay, and thus, a higher exploration gain.

$$expval = maxReward * 0.5 * exp(-episode/\beta) \tag{5.3}$$

5.2.1.6 Wall Avoidance Behavior

This module also works cooperatively with the QL learning module. It prevents the robot from bumping into walls by assigning negative values to the actions that would lead to them.

5.2.1.7 Action Selection

In order to select an action, a linear combination of each action value provided by the different modules (QL, Taxic, Exploration, Wall Avoidance) is performed. The action with the greatest value is chosen as the next action to execute in a winner-take-all fashion, as explained in chapter 4.

5.2.2 Model Implementation Details

The model was implemented using the Mobile Internet Robotics (MIRO) [172] simulator and the Neural Simulation Language (NSL) [171]. The same model implementation, with the exact same parameters, was used for both the continuous simulated environment and the real robot one. The only difference was whether the model moved a simulated robot or the real one after each decision.

The robot possible actions consisted on: go forward 0.05 meters, rotate $\frac{\pi}{8}$ to the left and rotate $\frac{\pi}{8}$ to the right. Note that the actions are relative to the robot.

The state for this algorithm was comprised of location and orientation information. Orientation information was needed due to the fact that rotations were coded in the robot's frame, so it needed to be aware of its orientation to make the right choice.

Location was encoded using Gaussian place cells fields, as explained earlier. Three layers were used of 100 uniformly distributed place cells each. Place field diameter was varied from 0.10 to 0.6 meters throughout these layers, corresponding to data reported on dorsal and medial hippocampus [92]. The activation function was nullified if it was lower than 0.2 for computational reasons.

Orientation was also encoded using Gaussian head direction cells. Four layers of orientation functions were used and the selectivity varied from π to $\pi/16$. The number of functions per layer varied depending on the selectivity in this case.

Soft-states were computed by combining all possible location and orientation cells activity levels. The resulting activity was computed by multiplying the activity of the individual cells as shown in Eq. (5.4), where x represents a 2D location, θ is the robot orientation, x_s is s preferred location, θ_s is s preferred orientation, σ variables are the specificities of s for location and orientation and g represents a non-normalized Gaussian function.

$$A(s, x, \theta) = g(x, x_s, \sigma_{s,x}).g(\theta, \theta_s, \sigma_{s,\theta}) \tag{5.4}$$

Table 5.1: Model parameters and their default value.

Parameter	Default Value
Learning rate α	0.1
Exploration decay rate β	1
Number of place cell layers	3
Place cell diameter	$(0.1, 0.3, 0.6)$
Place cells per layer	100
Number of head direction cell layers	4
Maximum head direction cell width	π
Minimum head direction cell width	$\frac{\pi}{16}$
Step size	$0.05m$
Step angle	$\frac{\pi}{8}$

Table 5.1 summarizes the default parameter values for the model.

5.3 Experiments

5.3.1 The Goal-Oriented Navigational Task

We chose a dry version of the Morris [101] water maze as our test bed for the navigation model. In this task, the robot navigates an $2m \times 2m$ square environment to go from the initial position to a fixed location goal. Once the robot reaches that interest point, an episode is considered finished. Many episodes are needed for the robot to learn a suitable navigation policy that will take it to the goal faster.

The robot is able to make three types of movements: turn right, go forward or turn left. After a turn, a forward motion is followed. In the presence of obstacles, subsequent turns are made until a forward motion can be carried out.

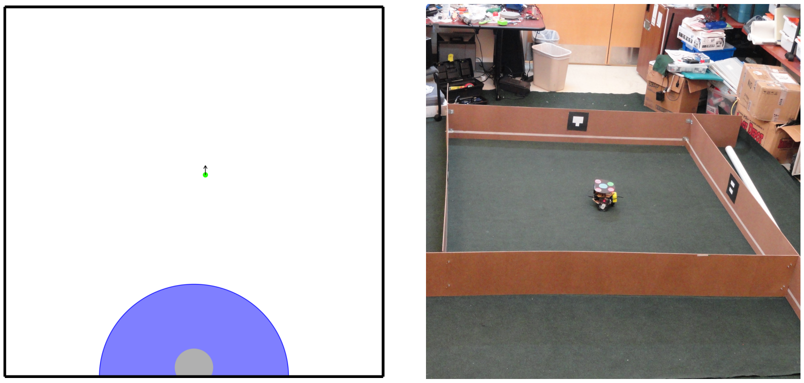
For our experiments, the individual was always put in the middle of the field, facing in the opposite direction to the goal.

The goal position is visible to the animal within a $0.4m$ radius.

Figure 5.2 illustrates the experiment setup.

5.3.2 The Experiments

We first tested the algorithm on a continuous and stochastic simulated environment to evaluate the multi-scale algorithm's performance in comparison to a single layer model. Then, we applied



(a) (b)

Figure 5.2: Experiment 1 and 2 setup. Experiment 1 (a): Morris square dry maze. The grey circle at the bottom represents the goal, the blue semicircle the region where the goal is visible, within a 0.4m radius, and the green dot and arrow the initial position and orientation. Experiment 2 environment (b): the robot is in the initial position and orientation with a patch on top for the SSL vision system to recognize. The black and white squares are the ARToolkit markers used by the SLAM system. The white line stripes at a fixed height used for wall detection is also shown.

Table 5.2: Experiment summary.

Experiment	Domain	Action Outcome	Simulated/Physical	Location Information
1	Continuous	Stochastic	Simulated	Global Vision
2	Continuous	Stochastic	Physical	Local Vision

the policies learned during the simulation to a physical robot navigation task, to validate the system performance under real environment conditions.

A more detailed description of each experiment is included next, while table 5.2 summarizes the main characteristics for all experiments. A description of the implemented robot programs for the experiments is included at the end of this section.

5.3.2.1 Experiment 1 - Simulated Robot Task

In this experiment we compared the performance of the proposed learning algorithm using three scales at once and using each of those scales of representation separately. The single scale systems were implemented using the soft state QL with only one scale for all place cells.

Table 5.3 summarizes the groups used in this experiments with their place field diameters.

Noise was added to the outcome of each movement performed by the robot. The added noise was sampled from a uniform distribution in the interval $[0, .2 * m]$ where m is the magnitude of

Table 5.3: Experiment 1 (left) and 2 (right) groups.

Group	Place Field Diameter
Mulit-Scale	$(0.1, 0.3, 0.6)$
Small Scale	0.1
Medium Scale	0.3
Large Scale	0.6

Group	Description
Learned	Policy learned through simulation and executed in the physical robot.
Naive	The physical robot with no policy learned.

the movement, i.e. meters traveled or radians turned. Namely, a $0 - 20\%$ noise was added to each movement. This was done in order to simulate physical robot conditions more accurately.

No noise was added to the position information provided to the robot.

Figure 5.2 shows the experiment setup. The robot started always from the same position and the goal position was fixed. An experiment consisted on a number of episodes that ended when the robot was able to reach the goal. One hundred different individuals were simulated, with 25 episodes each.

5.3.2.2 Experiment 2 - Physical Robot Tests

This experiment consisted of performing the same goal reaching task using a physical robot.

Figure 5.2 shows the testing environment. It consisted of a $2 \times 2m$ side square with small walls. In each side, an artificial marker was placed for the robot to use as landmark in a small Fast-SLAM system. In the physical robot experiment, position information was derived from local sensory information, as opposed to using global information like it was done in the simulation tests.

Each wall also included a white line stripe at a known height to allow the robot to derive the distance to them using monocular vision.

A differential robot powered by AX-12 motors was used. It used a BeagleBone Black single-board computer and a web camera as its only sensor. All sensory information was processed on-board, including landmark and wall detection. Piloting algorithms and self-motion computations were run on-board as well. Finally, the implemented visual Fast-SLAM system was also run on-board. The robot program with the MSQL algorithm, however, was run off-board on a personal computer, connected to the robot using a Bluetooth network (PAN).

A global camera recorded the robot position at each iteration using the Small Size League vision software [181]. This information, however, was not made available to the robot program during the decision making process.

A policy was learned during 25 simulated iterations. Then, the policy was loaded into the robot and one episode was carried out.

As a comparison, a robot without knowledge was put to perform the same task.

Table 5.3 shows the groups used in this experiment.

Each individual started at the center of the maze, as shown in Figure 5.2. Before the beginning of the experiment, an initial routine of twelve 90 degree rotations was performed. This allowed the SLAM system to build a stable map in a similar coordinate system as the global camera, because the coordinate frame is not determined by the initial position of the robot, but by the map of the surviving particles in the Fast-SLAM algorithm. The episode ended when the robot was within a radius of $0.4m$ of the goal.

The SLAM system position was fed to the place cell layer to compute the firing values at each iteration. Namely, the location information used to determine the firing values of the artificial place cells was derived from local sensory information only.

Six different individuals were used for each group.

5.4 Results

5.4.1 Experiment 1

Figure 5.3 shows the average number of steps needed to reach the intended goal as a function of the episode number. The default model parameters were used for this experiment, namely $\beta = 1$ (exploration decay, Eq. 5.3) and $\alpha = .9$ (learning rate, Eq. 4.5). Standard deviation per repetition is also shown.

An ANOVA test was done for each group to compare the completion times for episodes 1 and 100. A statistical difference was found in all cases ($p < 0.05$).

In order to assess the impact of the exploration decay parameter on the robot performance we tested different configurations. Figure 5.4 shows the time to reach the goal for all groups over 100 individuals, for a fixed α of 0.9 and different β values.

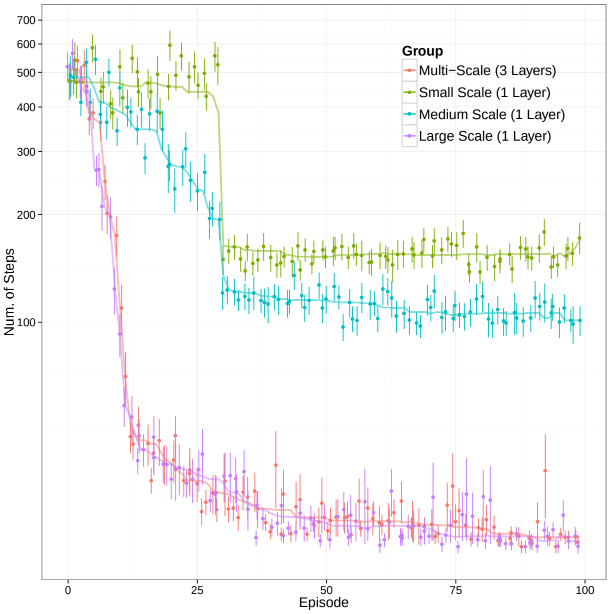


Figure 5.3: Number of steps to reach the goal as a function of the episode number for all groups of Experiment 1.

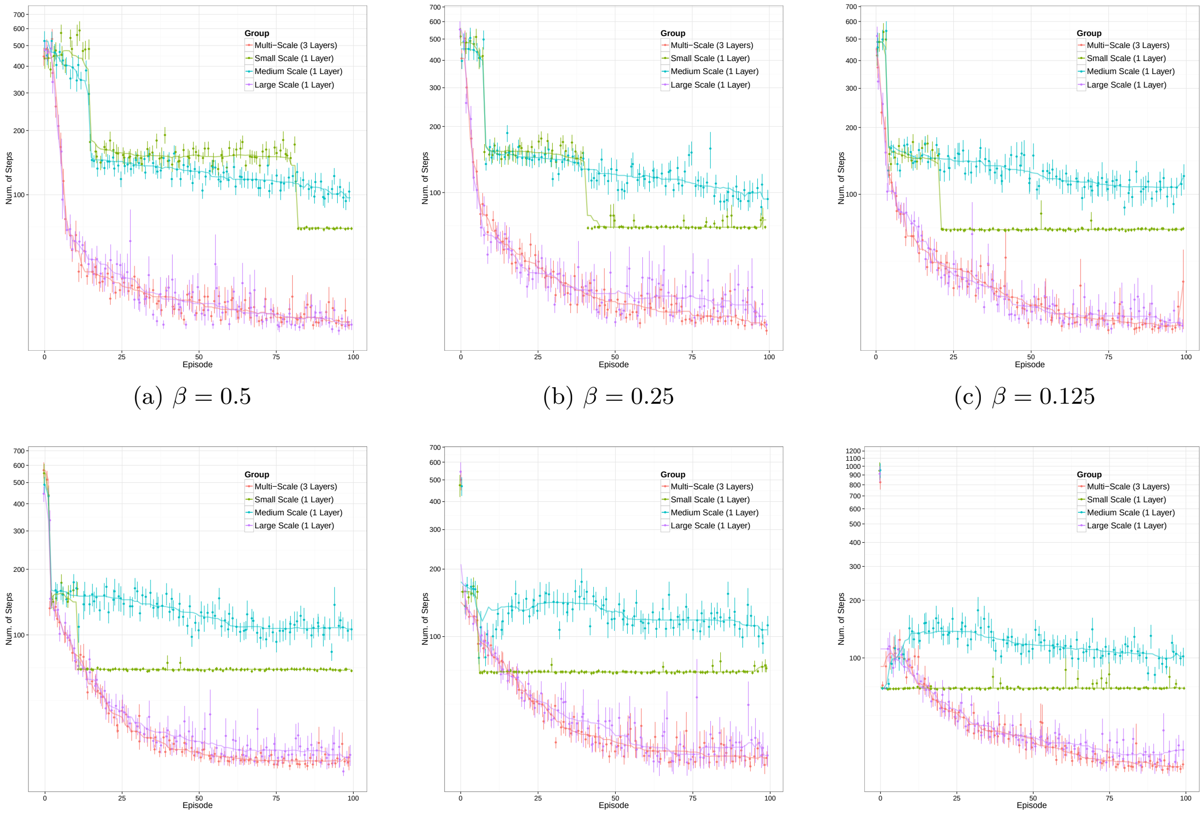
The groups Multi-Scale and Large Scale behaved similarly, so we include a Table 5.4 with results of an ANOVA test and Tukey HSD post-hoc for each β value. The tests were carried out using the completion times for the second half of the episode, i.e. the last 50 trials.

Adding to this, we include sample paths from different values of β, for different groups, at different episodes, shown in Figure 5.5.

Figure 5.6 shows policies acquired after the 20^{th} repetition for the Multi-Scale group for exploration values of $\beta = 0$ and $\beta = 1$. In order to plot the policy, a grid of sampling points was laid out over the environment. Then, the simulated robot was placed in every point and the orientation was varied. For each orientation, the value of the most valued action was taken. The orientation that gave the maximum expected value was plotted in each sampling point.

We also executed the experiment for different values of the learning rate parameter α. Figure 5.7 includes the finishing times for each episode, averaged over 100 individuals, for all groups.

Figure 5.8 shows two sample paths corresponding to episode 50 for one individual of the Multi-Scale group, for α values of 0.4 and 0.9.



(a) $\beta = 0.5$

(b) $\beta = 0.25$

(c) $\beta = 0.125$

(d) $\beta = 0.06125$

(e) $\beta = 0.03125$

(f) $\beta = 0$

Figure 5.4: Task completion times for 100 individuals per group for different values of the exploration decay factor β. The learning rate α was fixed at 0.9.

5.4.2 Experiment 2

Figure 5.9 shows the number of steps needed to reach the goal for the simulated learned policy tested on the robot. The Naive group had no learned policy, whereas the Learned group used the policy learned through 25 simulated repetitions.

A t-test was run on this data to check for significant difference of means. The groups means were found to be significant ($p < 0.05$).

Figure 5.10 shows two sample paths, one from each group.

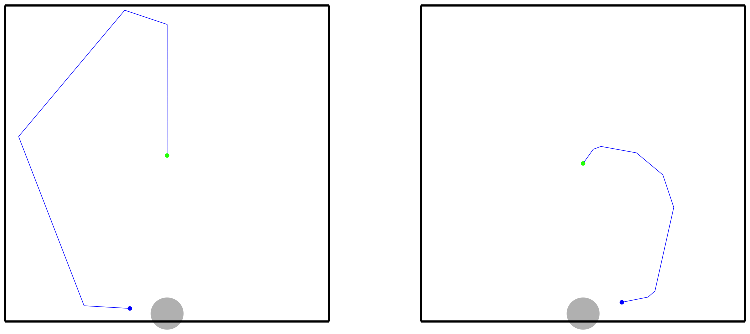
5.5 Discussion

5.5.1 Experiment 1

An initial experiment using the model default parameters showed faster learning and a better asymptotic solution for the Large Scale and Multi-Scale groups than for other groups.

Table 5.4: ANOVA test and Tukey HSD post-hoc results for the Multi-Scale and Large Scale groups, for all experiments varying β.

β	Mean Difference (Large - Multi)	p value
1	-0.65	0.997
0.5	8.56	0.00057
0.25	2.34	0.608
0.125	5.75	0.017
0.0625	6.51	0.0033
0.03125	3.82	0.078
0	4.31	0.05

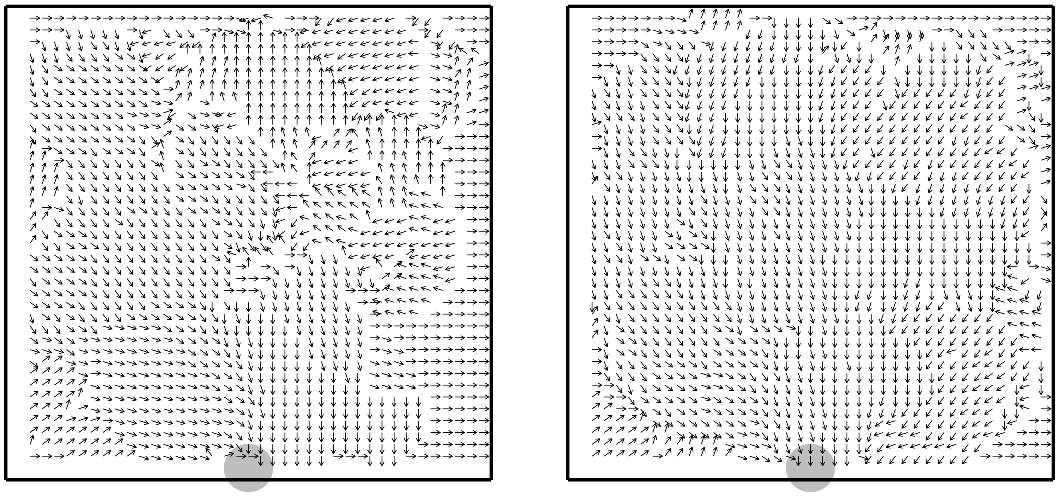


(a) $\beta = 0$, Small Scale, Episode 100 (b) $\beta = 1$, Multi-Scale, Episode 96

Figure 5.5: Sample paths from individuals from different groups, for different values of the exploration decay factor β.

Tests were carried out to assess the impact of the exploration parameter β on the completion time and several interesting phenomena were observed. A sudden drop in completion times can be observed in almost all cases, see Figure 5.4. This drop is more noticeable for the Small and Medium Scale groups. This drop seems to occur earlier as the exploration decay velocity is increased. We attribute this to the fact that exploration is done by assigning a decaying value to a random action. Then, when learned action values meet this decaying value, the learned policy takes control and no more time is wasted in exploration.

In the case of the Small Scale and Medium Scale groups, there was a second drop in completion times late in the experiment. Our model assigns an infinitesimal value to forward motions to favor



(a) $\beta = 0$

(b) $\beta = 1$

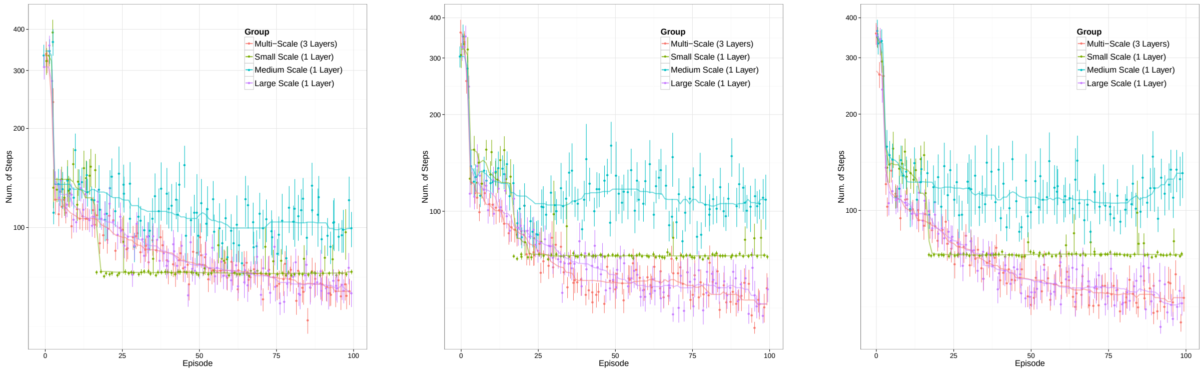
Figure 5.6: Policy of the a sample individual of the Multi-Scale group after the 20^{th} repetition.

them over rotations in case of a complete tie. When the exploration value decays below this value, forward motions are favored over rotations. Then, the strategy turns into a route navigation [129] one, in which the simulated robot went forward until it reached a wall, turned and repeated the action, as seen in Figure 5.5a. For a given number of place fields, the Small and Middle layers are not able to cover as much of the environment as the other two groups. The Small and Medium groups might therefore not be able to learn across the whole field and perform actions that would take them out of this suboptimal route navigation strategy. This is supported by the fact that the Medium Scale group shows an increasing tendency to default to this behavior, showing a second decay in completion times, as exploration decays faster.

The Large Scale and Multi-Scale groups, on the other hand, showed more optimized learned paths, such as the one shown in Figure 5.5b.

Differences in the learned policies for different values of exploration can be appreciated. Slower exploration decay allowed the simulated robot to explore more of the environment, learning the proper action to perform at every possible point. This resulted in more coherent policies, see Figure 5.6.

Additionally, we observed that the Large Scale and Multi-Scale groups behaved similarly for all exploration parameters, see Figure 5.4. They showed a faster initial learning (first 10 episodes)

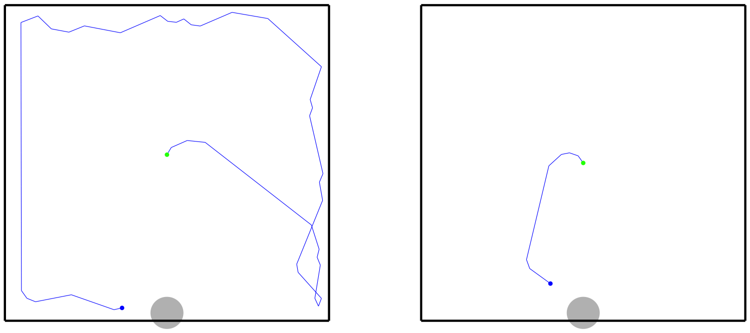


(a) $\alpha = 0.4$

(b) $\alpha = .8$

(c) $\alpha = 0.9$

Figure 5.7: Task completion times for 100 individuals per group for different values of the learning rate parameter α. The exploration decay parameter β was fixed at .125.



(a) $\alpha = 0.4$

(b) $\alpha = 0.9$

Figure 5.8: Sample paths from one Multi-Scale individual at episode 50 for different values of the learning rate α.

than the other two groups for all exploration parameter values. However, this relation was inverted once the other groups fell into praxic strategies. Statistical analysis showed some significant, but slight, differences in the average completion time for some exploration parameter values, favoring the Multi-Scale group.

One interesting phenomena with regards to exploration could be observed. It would seem that by decreasing exploration, learning speed is increased. Specially in the case of the small scale system, which reaches a stable suboptimal solution sooner, as the exploration energy is reduced. However, we observe that the navigation problem has a suboptimal but simple praxic solution, as the one shown in figure 5.5. Namely, the problem can be solved by always going forward and turning at the walls always in the same direction. This solution could be learned fast by the model,

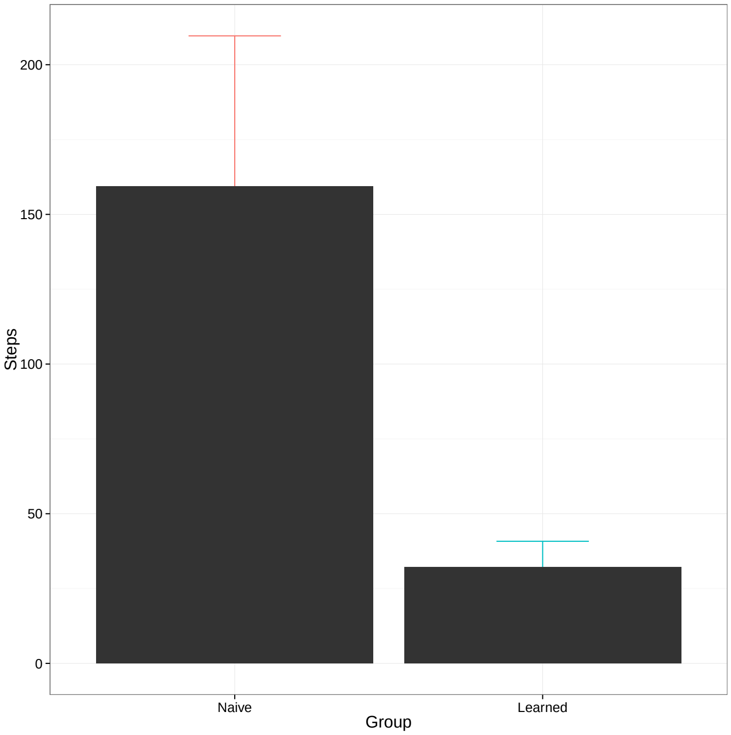


Figure 5.9: Experiment completion time for the real robot trials for the Naive and Learned groups.

as it only needed to learn the advantages of going forward and acquire a turning bias. Then, when exploration energy was low and no better solution had been learned, the individuals tended to use this solution, which despite being suboptimal, it allowed the animat to reach the platform quite fast.

Variations in the learning rate parameter showed the greater learning potential of the Large and Multiple Scale groups. At higher values of α, these groups were able to continue learning until they improved completion times with respect to those achieved by the Small and Medium Scale groups.

Summarizing, this experiment has shown that large and multi-scale representations serve as a better source of information for this navigation task. We attribute this to the fact that they provide a better coverage per place field and that they allow for faster generalization of the learned value of a certain region. This goes in line with the fact that the ventral portion of the hippocampus has more projection to value estimation regions, such as the ventral tegmental area [50, 5]. Namely, the presence of large place fields allows for a faster propagation of the reward values to the rest of the environment. The fact that the less accurate cells are a better source of information for

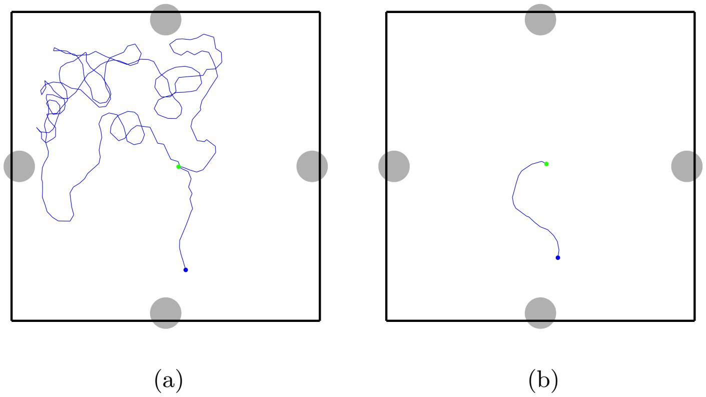


Figure 5.10: Sample paths for the robot showing the position reported by the global camera system, for the Naive group (a) and Learned group (b).The additional gray circles (north, east and west) indicate the position of the landmarks used by the SLAM system.

navigational decision making could seem counter intuitive. However, the fact that many fields overlap in a given place and that the value of an action is computed as a linear combination seems to compensate for the lack of precision of each individual field. The finding that the Multi-Scale group is significantly better for most exploration parameter values indicates that the presence of small cells is also beneficial.

5.5.2 Experiment 2

Experiment 2 was challenging to the algorithm for various reasons. This experiment involved the use of local sensory information to derive the robot location. Thus, the algorithm had to cope with noise in the reported position when making action selection decisions. What is more, there could be noise in the learning process when pose correction events occurred in the underlying SLAM system. After performing a single motion, the robot could find itself suddenly far away from its original position, due to re-localization in the SLAM system. Then, the algorithm would erroneously update the value of the performed action in the previous state.

The levels of motion noise of the real robot were also greater than those used in the simulator.

Despite this, the physical robot was able to execute a policy learned in simulation and significantly reduce completion times.

The paths reported by the global camera system show how the learned robot was able to reach the target faster (Fig. 5.10).

CHAPTER 6

MULTI-SCALE ROBOT NAVIGATION LEARNING ALGORITHM

6.1 Introduction

This chapter uses what was learned from the previous chapter to propose an algorithm for robot navigation learning. It focuses on applying a Multi-Scale RL algorithm to the robot navigation domain. First, we implemented a proof of concept test scenario inspired in the taxi problem introduced by Dietterich [38]. By doing so, we are able to compare it to standard learning algorithms that work on discrete spaces. Then, we applied the algorithm to a real continuous navigation problem, based on the same taxi scenario.

This chapter incorporates tests against a widely used learning algorithm, a canonical Q-Learning. It also incorporates the presence of inner obstacles in the environment, which the model in the previous chapter does not account for. This demands for modifications in the model, as explained below.

The following sections introduce the task, the algorithm, the experiment results and discussion.

6.2 The Task

The task develops in a discrete, grid-like, world and consists of learning to navigate to a certain position to pick a passenger, and then navigate to another position to drop the passenger. The passenger and drop-off sites are always picked from four fixed interest points (Y, R, G, B) [38].

For the first experiment, we kept the discrete state hypothesis to test the algorithm's behavior and be able to visualize certain aspects of its dynamics more easily. We enlarged the taxi problem arena to be 20 squares sided and kept the original 4 interest points (Y, R, G, B). Figure 6.1 shows the problem map and the robot in its starting position. This experiment also allowed us to compare the algorithm against a canonical QL implementation to test the hypothesis of faster learning times.

Then, for the second experiment, we implemented a continuous state simulated version of the same taxi problem to test the algorithms real potential over a continuous environment.

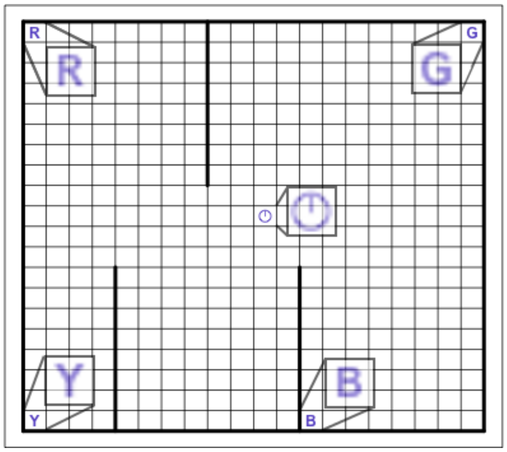


Figure 6.1: The discrete taxi problem enlarged to a 20 × 20 grid. The robot is in the center cell heading upwards. The letters Y, R, G and B denote the interest locations. Zoomed sections were included on the regions of interest to facilitate visualization.

6.2.1 Discrete Taxi Problem Experiment

The task we wanted our robot to learn consisted of the first half of the taxi problem. Namely, we wanted our robot to go from the initial position to a randomly picked interest point. Once the robot reached that interest point, the episode was considered over. The robot had knowledge of which point it wanted to reach but no prior knowledge of where those points where.

The robot location was fully described by the Cartesian coordinates of the grid cell it was in, as well as the robot orientation. The latter could take values multiple of $\frac{\pi}{2}$.

At every iteration, the robot performed one of four possible actions, going North, South, West or East. After going in any direction, the heading of the robot was updated to match that direction.

We ran 100 episodes for 100 different agents executing the canonical QL agent and 100 agents executing the Multi-Scale QL agent.

6.2.2 Continuous Taxi Problem Experiment

We also implemented a simulated continuous version of the taxi problem using the Mobile Internet Robotics (MIRO) [172] simulator and the Neural Simulation Language (NSL) [171].

In this version the field side was $2m$. Figure 6.2 shows the environment and the robot initial starting position.

The robot possible actions consisted on: go forward 0.05 meters, rotate $\frac{\pi}{8}$ to the left and rotate $\frac{\pi}{8}$ to the right. Note that the actions are relative to the agent now. Additionally, uniformly distributed noise of up to 5% of the motion's magnitude was added at each step, both to linear

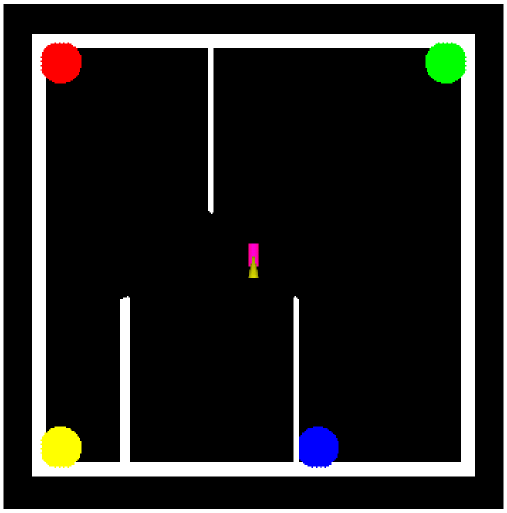


Figure 6.2: The continuous taxi problem. The pink box with a yellow triangle represent the robot, which is heading downwards. The colored circles represent the interesting places.

and rotational movements. We consider the addition of noise to be important, as it reflects the kinematics of a real mobile robot more accurately. It also prevents the agent from always being in a discrete finite subset of possible states due to the motion commands being too regular.

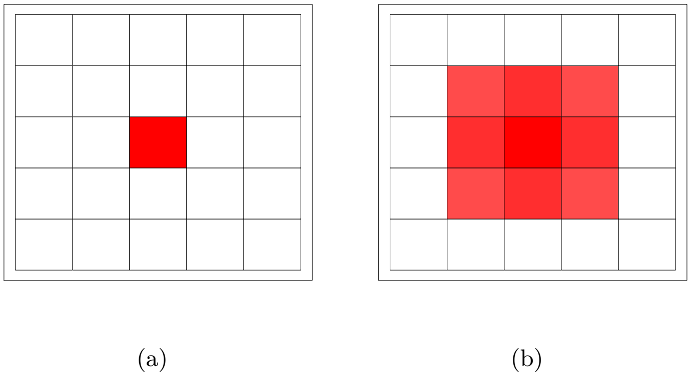
We ran 100 episodes for 64 different agents executing the continuous version of the algorithm. To simplify analysis, the goal was kept fixed to the blue goal for this experiment.

6.3 The Algorithm

6.3.1 Discrete Algorithm

Two different agents were put to solve the task. First, a canonical QL was implemented. The state was comprised by three nominal values: two for the Cartesian coordinates and one that coded the intended place to reach.

The Multi-Scale agent was implemented by coding the state with two layers of soft-states. The first layer was equivalent to the usual grid states, as the states were active only when the robot was in the preferred cell. The second layer was implemented using less selective soft-states, which activity was determined by Eq. 6.1, where x is a cell in the grid, x_s is state s preferred cell and $manh(a, b)$ is the Manhattan distance between cells a and b. Figure 6.3 shows the activation of a cell with preferred location at the center of the grid, for the first and second layer.



(a) (b)

Figure 6.3: Cell activation pattern for the Multi-Scale agent. The cell's preferred position is at the center of the grid. The level of red shows the activation value when the agent is at that cell of the grid. The closer to the cell's preferred position, the greater the activation. The first layer (a) only fires in the preferred location, whereas the second layer (b) fires in the 9-neighborhood of the preferred location.

$$A(s,x) = \begin{cases} 1 & x = x_s \\ .8 & manh(x, x_s) = 1 \\ .7 & manh(x, x_s) = 2 \end{cases} \tag{6.1}$$

Both algorithms were working cooperatively with a greedy taxon behavior [59], which consisted of approaching the goal when it was visible to the agent. The goal was considered visible if it was in the agent 180 degree visual field, no wall occluded the goal and it was at most 4 squares away from the agent. The taxon behavior assigned a fixed value to the action that moved towards the goal. We believe the incorporation of these sensor oriented behaviors are a good way to fill the gap between theoretical RL and its application to robotics.

Additionally, a cooperative exploring behavior assigned a fixed value to a random action. No ϵ-greedy approach was implemented in the RL algorithms. We believe that by keeping the exploration component separated from the RL algorithm, more sophisticated exploration algorithms can be incorporated. This algorithms may rely on data that is not available to the RL component or may be to complex too get in the RL loop.

Finally, the values for every action from all cooperative behaviors were added, and the action with the greatest value was picked in each iteration.

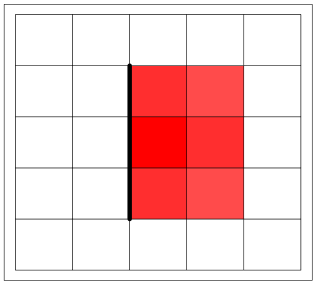


Figure 6.4: Multi-scale cell activation in the presence of a wall. The cell's preferred position is at the center of the grid. The level of red shows the activation value when the agent is at that cell of the grid. The closer to the cell's preferred position, the greater the activation. If a wall is between the agent and the cell's preferred location, the cell will not be active (left of the wall).

Table 6.1 shows the parameters used for each agent. Both algorithms were calibrated by trial and error, trying to keep both configurations as similar as possible. The value for γ was initially set to 1 for both algorithms, as there was no real need for a discount factor due to the episodic nature. However, it was observed that this parameter helped the multi-scale version stay out of local minima. The robot was given one of two different values of rewards. If it reached the goal, the Goal Reward was given, whereas the Non-Goal Reward was given otherwise.

Another important aspect to mention is that soft-states with activity at both sides of any wall was deactivated completely. In other words, if a less selective soft-state had its preferred value in one side of a wall, but was active when the agent was in the other side too, the state was disabled. Figure 6.4 illustrates this. This was necessary to avoid the conformation of absorbing local minima that occurred after the update of these less selective states with an action that led the agent towards that wall.

6.3.2 Continuous Algorithm

The state for this algorithm was comprised of location and orientation information, as well as the intended place to reach. Orientation information was needed due to the fact that rotations were coded in the robot's frame, so it needed to be aware of its orientation to take the right choice.

Location was encoded using Gaussian activation functions of the distance to a preferred location. At first, single scale experiments were carried out, testing cell radius of 0.2 to 0.4. Then, multi-layer experiments were carried out. Two layers of 20000 uniformly distributed functions each were used.

Table 6.1: QL algorithms parameter values.

	Canonical QL	Multi-scale QL
α	0.8	0.8
γ	1	0.9
Goal Reward	1000	1000
Non-Goal Reward	-5	-5
Taxon Value	50	50
Exploration Value	25	25

The selectivity, or variance, of the Gaussian function was varied from 0.2 to 0.35 meters throughout these layers. The activation function was nullified if it was lower than 0.2 for computational reasons.

Orientation was also encoded using Gaussian activation functions of the distance to a preferred orientation value. Four layers of orientation functions were used and the selectivity varied from $\pi/2$ to $\pi/16$. The number of functions per layer varied depending on the selectivity in this case.

Soft-states were computed by combining all possible location functions, orientation functions and intended place. The resulting activity was computed by multiplying the activity of the individual functions, where the intended place was coded with an indicator function, as shown in (6.2).

$$A(s, x, \theta, p) = g(x, x_s, \sigma_{s,x}).g(\theta, \theta_s, \sigma_{s,\theta}).\mathbf{1}_{p=s_p} \tag{6.2}$$

Variable x represents a 2D location, θ is the robot orientation, p is the goal place, x_s is s preferred location, θ_s is s preferred orientation, p_s is s goal place, σ variables are the specificities of s for location and orientation and g represents a non-normalized Gaussian function.

As with the discrete algorithm, cells did not activate across walls.

The continuous algorithm was implemented using an actor-critic architecture, instead of a Q-Learning one, in order to allow to test asymmetric contributions to value and action-value estimations, as discussed in 4.4.1.

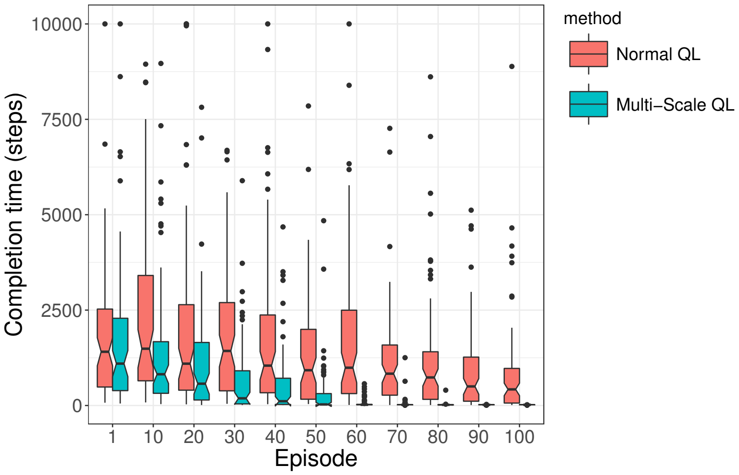


Figure 6.5: Average number of steps to reach the goal as a function of the episode number for the discrete problem.

6.4 Results

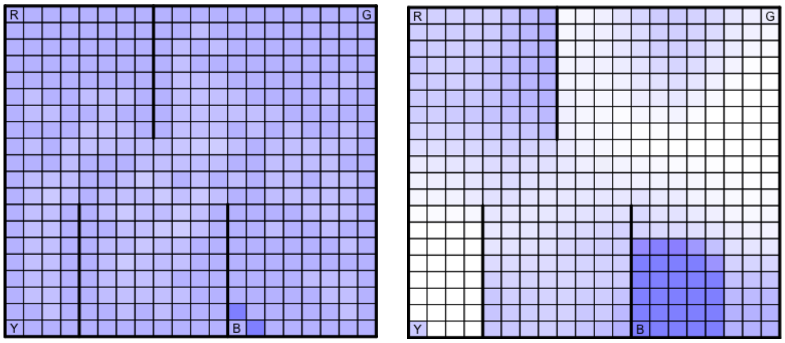
6.4.1 Discrete Taxi Problem Experiment

Figure 6.5 shows the average number of steps to reach the goal as a function of the episode number taken over all 100 agents.

In addition, we performed an analysis of variance (ANOVA) test on the number of steps as a function of episode number and algorithm. This was done to determine if there was any range of episodes in which there was statistical significant difference on the number of steps it took to reach the goal between one method and the other. We found that the number of steps was significantly different from episodes 29 to 84. There were other episodes with significant difference, but the mentioned segment was the largest range found with uninterrupted significant difference.

In order to illustrate the workings of the algorithm, we included Figure 6.6, in which the Q value of each cell is plotted for a random individual at the start of the third episode of going to goal B. The value shown in Eq. 6.3 is taken for every cell and plotted as a heat-map, where dark blue corresponds to one time the exploration value and white corresponds to 0. We compared to the exploration value because at this point, the RL algorithm starts driving the agent as it overrides the exploration behavior.

$$\frac{max_a Q(s,a)}{explorationValue} \tag{6.3}$$



(a) (b)

Figure 6.6: Plotted Q value for each cell for Canonical QL (a) and Multi-Scale QL (b).

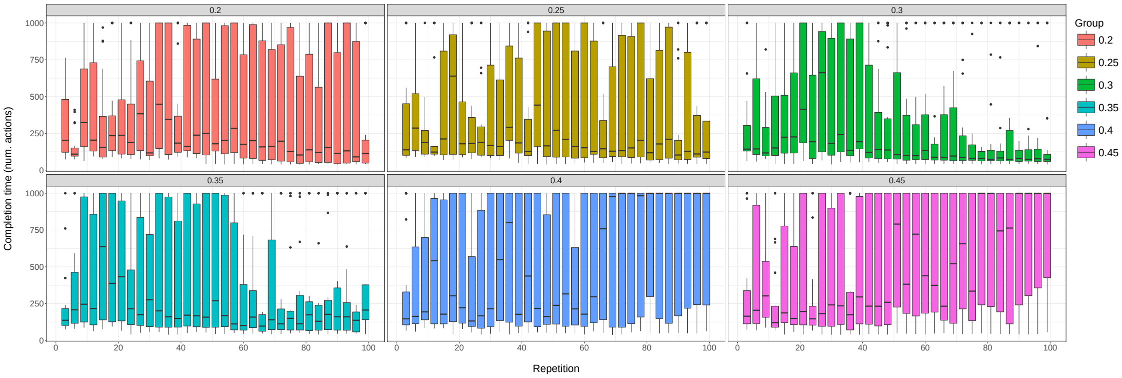


Figure 6.7: Runtimes in simulation steps for the goal B, for different cell sizes (single layer).

6.4.2 Continuous Taxi Problem Experiment

Figure 6.7 shows the runtime in simulation steps for a single layer going to goal B exclusively, for different cell sizes. Cell sizes of 0.3 and 0.35 are able to learn the task, converging to a lower runtime.

Figure 6.8 shows the runtime for goal B, for the multi-scale system. Two different ways of using the place cell output were tested (see 4.4.1):

- Using both layers to estimate the value function $V(s)$ and action values $Q(s,a)$ indistinctly (Symmetric group)
- Using the larger cell layer to estimate the value function $V(s)$ only, and the smaller cell layer to estimate the action values $Q(s,a)$ only (Asymmetric group)

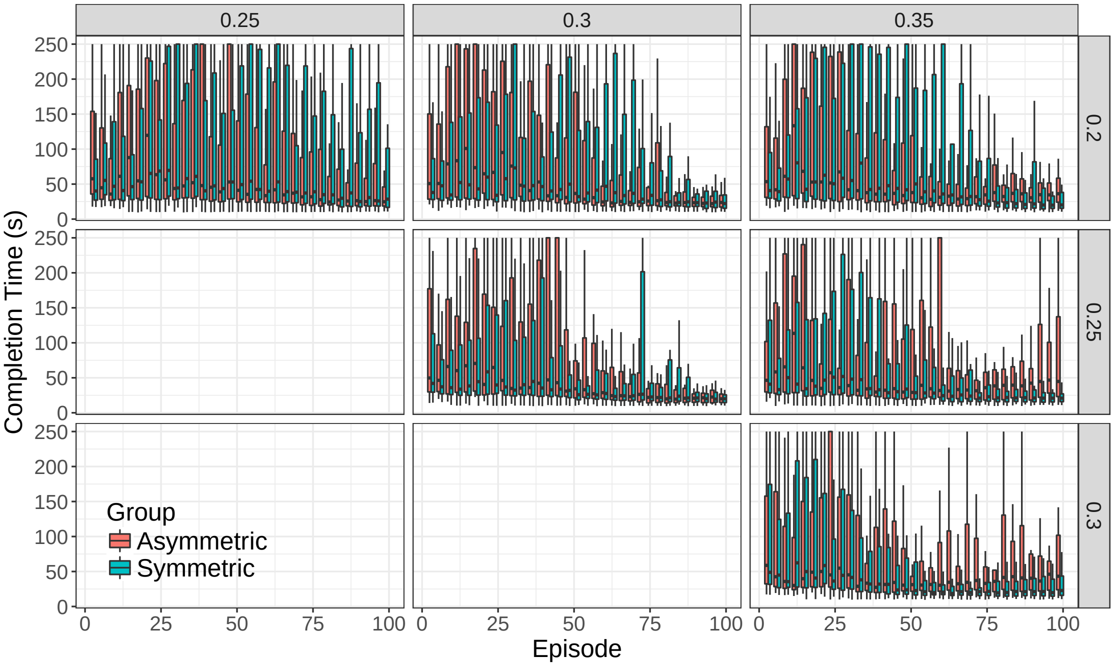


Figure 6.8: Runtimes in simulation steps for different small (y axis) and large (x axis) cell sizes, for symmetric and asymmetric contributions. Outliers are not shown.

6.5 Discussion

The experimental outcomes of the first experiment have met our working hypothesis predictions. This experiment, performed in a discrete domain has shown that the algorithm converged faster than a canonical implementation, for the chosen, very similar, sets of parameters. We attribute this phenomena to the less selective states, which change the Q values of larger regions upon an update event.

Figure 6.6 qualitatively shows how the value function surface changes radically between algorithms after just 2 episodes.

Quantitatively, the results thrown by the significance test have shown that the multi-scale algorithm converges faster to lower latencies than the canonical one for these sets of parameters. The canonical QL algorithm does not seem to converge to a steady value, and is certainly far from optimal (observe that the optimal has to be less than twice the scenario length). Thus, we would have to carry out further tests before being able to conclude on the quality of the obtained policies.

The second experiment tested the algorithm in a continuous environment, opposite to enforcing a discrete set of states. The test over cell radius shows that large cells of 0.3 to 0.35 m radius are needed to properly learn the task in 100 episodes. Agents with lower cell radius failed to learn a policy that reaches the goal. Inspection of their paths show that they timeout in later episodes because they have learned negative values associated to the effort of moving towards the goal, but the value of reaching it could not propagate backwards fast enough. Consequently, they avoid the goal area whatsoever. Individuals with larger cell sizes failed for a different reason, they learned incorrect policies, which kept them doing rotation motions in specific places of the environment. In order for a RL algorithm to converge, all states must be visited with probability $p > 0$ in the future [155], which does not hold in this setup due to exploration being bounded for practical reasons.

It is worth noticing that a radius of $0.3m$ is large for a place cell in a $2m \times 2m$ environment. Given that the activation of the final units is a multiplication of both place and head direction modulation (Eq. 6.2), the effective range of the cell is reduced, which explains the need of larger cells than those found in nature.

Taking values around the effective cell sizes, symmetric and asymmetric contribution to value and action-value estimation were tested. All tested values seem to have learned a policy that lowers the runtime. Comparing them, however, is difficult because it depends on the criteria being used (e.g. lower median or lower spread). Despite this, some general conclusions might be drawn:

- Whan small cell radius is small, the asymmetrical system is more stable. The asymmetric group uses these to estimate action-values only, where as the symmetric group uses half of them to estimate value and half to estimate action-values. This implies that small cells are not appropriate to estimate value. This is consistent with subsection 4.4.1.

- When small cell radius is large ($0.25m$ and $0.3m$), the symmetrical system seems to outperform the asymmetrical one, with lower and more stable runtimes in later episodes. This implies the learning algorithm benefits from using large cells for action-value estimation. This behavior was not predicted by our working hypothesis stated in subsection 4.4.1. However, it seems reasonable that provided there is enough size for value propagation to occur, larger cells can benefit action selection, even in the presence of obstacles.

CHAPTER 7

MULTI-SCALE MODEL FOR A NOVEL TASK

7.1 Introduction

This chapter introduces the application of multi-scale learning to a task that is also a contribution of this dissertation (in collaboration with Fellous Lab).

The task was designed with complexity in mind: the original hypothesis was that the larger scales of representation would be needed more as the complexity of the task increased. This could explain why larger scales are not essential in a simple Morris maze task, but are in more complicated tasks (see Discussion).

Thus, the task involves multiple goals, cued and delayed cued goals, semi-dynamic environments and hippocampus deactivation.

The task was run concurrently on agents executing the model (simulated and robots) and in real rats (Fellous Lab). Thus, we were able to compare results and attempt to provide explanations to the observed phenomena in the context of this newly designed task.

The next sections present the task and the model, the obtained results and a discussion with an emphasis on modeling the biological system (the rat).

7.2 The Task

The task consisted of training an individual animat (i.e. artificial animal) to learn a goal-oriented navigation task and then performing a recall session on a modified environment, where obstacles are introduced into the environment. Eight feeders were laid over a 1m diameter circular open field maze. The feeders' angular position was distributed uniformly. Feeders could also be cued, i.e. drawing the attention of the animat/animal towards it. In the real rat version, A LED was placed above each feeder to use as queue when needed (see below). In the animat version, information on which feeder was being cued was provided to the model. Figure 7.1 shows the layout of the maze.

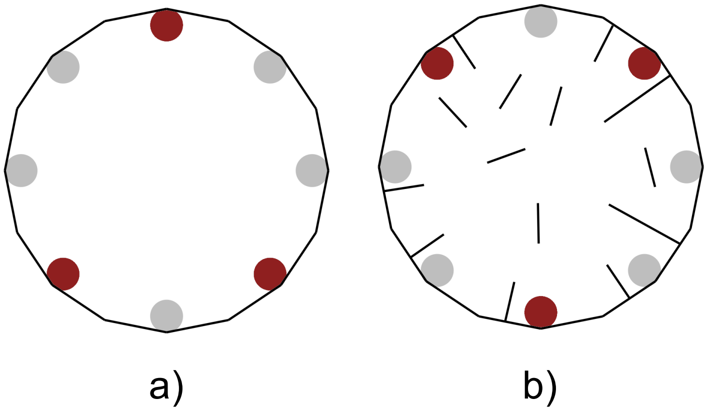


Figure 7.1: The experiment layout. (a) The maze layout. The circles represent the feeders, with the set represented in red. Black lines show the walls in the environment (b) A sample disposition of obstacles for the recall phase. The interior black lines represent the placed obstacles.

For each experiment, a subset of three feeders, known as the set, were selected to give rewards, whereas the other ones did not have food at any time [69]. The set of 3 feeders was fixed throughout the experiment although rewards were not always offered to the animat, as explained below.

The experiment consisted on three different phases:

1. First, a *non-delayed cue* phase was executed. A feeder was picked from the set and a light on top of it was flashed until the animat fed from it. This was repeated 100 times.

2. A *delayed cue* phase was then executed. In this phase, no feeder flashed initially. The animat was allowed to go through the feeders freely and without flashing cues. If enough time passed without the animat feeding from one of the 3-feeder set, one of them was flashed until the animat reached a correct feeder. This phase was executed until the animat consecutively reached 15 feeders from the set, without making any mistake (wrong feeder) and only reaching two flashing feeders in those 15.

3. The last phase, called *delayed cue with obstacles*, only differed from the previous one in that a set of small obstacles was placed in the environment. The obstacles consisted on 12.5 cm wide barriers. Some of them were put against the maze wall, whereas the rest were placed towards the middle of the maze. Some of the barriers near the wall were placed together to form a bigger (25 cm) barrier. Figure 1b shows a normal barrier layout.

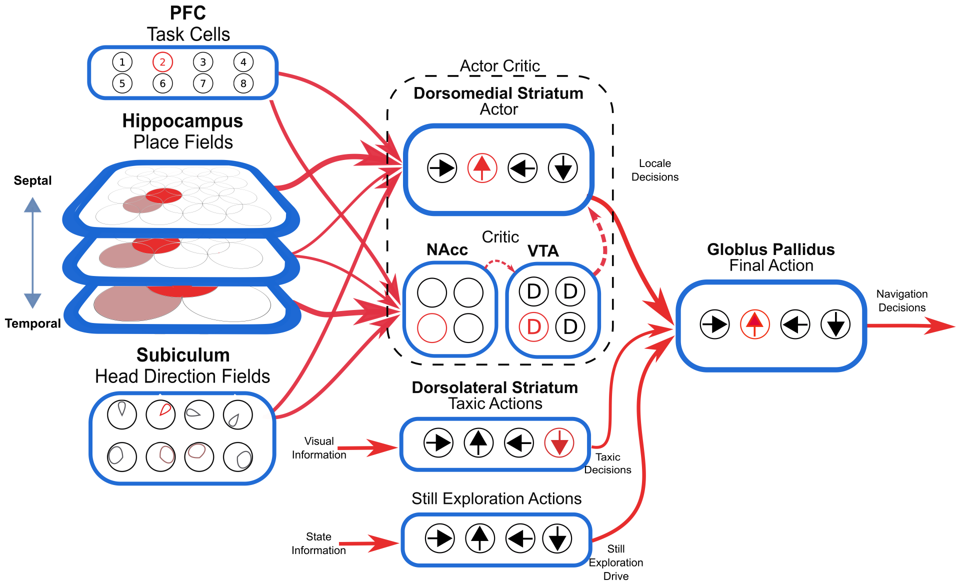


Figure 7.2: The model of spatial navigation. Hippocampal place cells are arranged in three layers along the longitudinal axis. The layers output is differentially projected to a value estimating complex, where information is input to the nucleus accumbens (Nacc) and relayed to the dopaminergic ventral tegmental area (VTA), and to action selection structures, composed by the dorsomedial striatum. The striatum also receives input from the PFC indicating the current state of the task (8 possible states, one per feeder) and from subicular head direction cells. Head direction cells have also different scales of representation (the response is shown curve inside each unit). Dopaminergic error signals are projected to the dorsomedial striatum, where they are used to learn the association between situations (stimulus) and actions (response). Additionally, visual information drives a taxic behavior module (dorsolateral striatum), and a still exploration module. All action selection information (circle with arrows represent actions) converges to a common structure for final action selection (Globus Pallidus), made in a winner take all fashion.

7.3 The Model

The model of navigation shown in Figure 7.2 is based on a reinforcement learning paradigm that uses information provided by the hippocampus (HPC), subiculum (SUB) and prefrontal cortex (PFC). The output of these regions is fed to a learning module comprised by the ventral tegmental area (VTA), dorso-medial striatum and ventral striatum (nucleus accumbens - NA).

7.3.1 Place Cells

The location of the animat is used to derive the firing rate of a set of place cells. The firing rate is determined by a Gaussian function of the distance to the cell's preferred location. Equation 7.1 shows the activation of cell pc, where l is the animat's current location and l_{pref} is the cell's preferred location. The parameter σ models the location specificity of the cell.

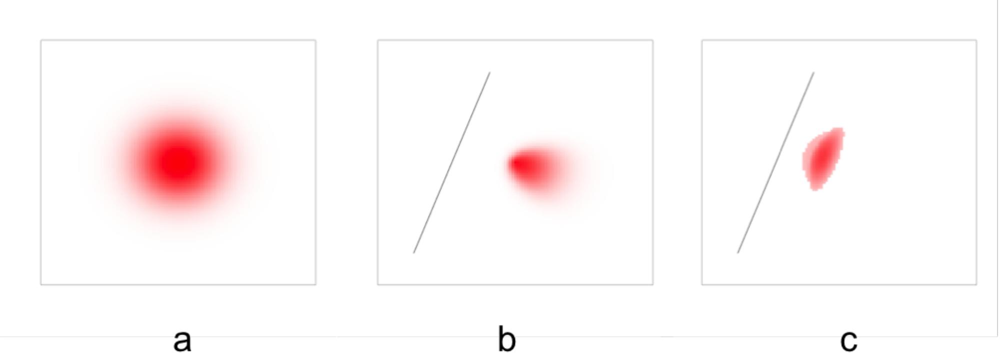


Figure 7.3: Interaction between place fields and obstacles. (a) A normal place field. (b) A normal place field after an obstacle has been introduced. (c) A wall cell place field after an obstacle has been introduced.

$$A(pc) = e^{-\frac{dist(l,l_{pref})}{\sigma}} \tag{7.1}$$

7.3.1.1 Interaction With Obstacles

This task involves the introduction of obstacles to a previously learned environment. It has been shown that obstacles interact with place cell firing patterns in many ways. It has been observed that cells can be silenced when obstacles are put within the cell's field [108]. Placing obstacles also creates “wall” or “obstacle” cells that fire only when the obstacle is introduced [135], which may be explained by boundary vector cell input [84]. It has also been shown that cells which fire when an obstacle is placed near its preferred location, usually only fire on one side of the obstacle [135].

We model these findings by assigning the place units to two categories, normal cells and obstacle cells. Figure 7.3 shows the interaction of place fields and obstacles for both types of cells. We describe the firing equations below.

When an obstacle is next to a normal cell's field, its firing is negatively modulated by a sigmoid function of the distance to the obstacle, as shown in Eqs. 7.2 - 7.4, where *dtco* is the distance to the closest obstacle, and r_{cell} and c_{cell} are the cell's field radius and center respectively and l is the animat current location. The distance to the obstacle is normalized by the place field radius to obtain d. Then, it is compared to the distance to the center to obtain d'. Finally, a sigmoid function of d' defines the modulation factor m that will have a multiplicative influence in the cell's firing. A linear function of d' is used, with constants a and b.

$$d = \frac{dcto}{r_{cell}} \tag{7.2}$$

$$d = max\left(0, d - \frac{dist\left(l, c_{cell}\right)}{r_{cell}}\right) \tag{7.3}$$

$$m = \frac{1}{e^{a*(d'-b)} + 1} \tag{7.4}$$

Equation 7.5 shows the modulation of obstacle cells. Their firing is positively modulated by the presence of nearby obstacles (right factor) and inhibited if too far from them (left factor). For the inhibitory factor, a different constant b' is used in the linear mapping of d'.

$$m = \left(1 - \frac{1}{e^{-a*(d-b)} + 1}\right)\left(\frac{1}{e^{-a*(d'-b')} + 1}\right) \tag{7.5}$$

7.3.2 Head Direction Cells

Head direction cell firing is modeled as a Gaussian function of the angular difference to the cell's preferred direction. Equation 7.6 shows the activation value of a head direction cell hc, where θ is the animat's current heading, θ_{pref} is the cell's current heading and σ is the cell angular specificity.

The model includes different scales of representation for angular information as well. The angular specificity σ for each cell is drawn from the interval controlled by two system constants $[\sigma_{min}, \sigma_{max}]$, the minimum and maximum angular specificity.

$$A(hc) = e^{\frac{dist\left(\theta, \theta_{pref}\right)}{\sigma}} \tag{7.6}$$

7.3.3 Task Units

Task cells signal the currently pursued sub-goal in the task. This information is needed given the multi-goal nature of the task. Namely, since there is more than one goal, the navigational decisions to be performed in a certain place depend on the currently pursued goal. The model includes eight task units, one per feeder, to represent this information. Relatedly, the multiple map hypothesis [129] proposes that some place cell activity depends not only on the location but on the current sub-task being carried out. Although it might be true that some place cells are

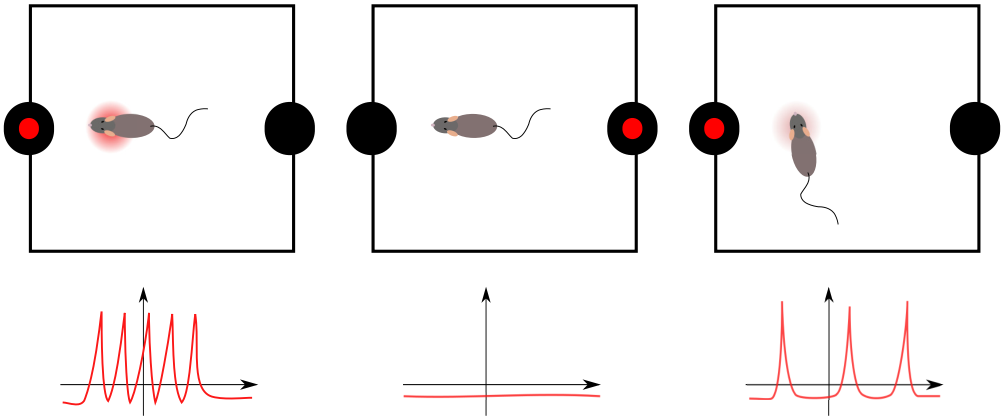


Figure 7.4: Striatal cell response in a square environment, where the rat has to visit two feeders (black circles). The currently pursued feeder is shown with a red circle on top. The sketches on the bottom show the activity of the striatal cell (the same cell for all three situations). Left: the striatal cell is firing maximally, due to the rat being in the preferred position, heading and pursuing the preferred feeder. Middle: the same striatal cell does not fire at all because the other feeder is being pursued. Right: the same striatal cell fires with lower rate because the heading of the rat is slightly off the cell's preferred heading.

strongly tuned to the current sub-task after training, we model this information as coming from the PFC and meeting spatial information in the striatum. This would also produce a multiple map hypothesis, but the multiple maps would be first found in the striatum, upon the convergence of the place and state information.

7.3.4 Striatal Cells

Cells in the striatum, both dorsal and ventral, receive inputs from place cells (HPC), head direction cells (SUB) and task cells (PFC). We build on previous work in which place and heading information met for the first time in the NA [16]. Each cell in the striatum is tuned to respond to one cell of each input source. Figure 7.4 shows the response of one of these cells in a two-goal task. The cell is tuned to the pursue of a particular goal, a particular head direction and a particular place.

The resulting activation of each striatal cell is computed as the product of the corresponding place, head direction and task cells' activities.

7.3.5 Taxic Modules

Reinforcement learning algorithms usually devote a lot of time to randomly exploring the state-action space. In a navigational task, this would correspond to a rat that moves randomly with no directionality at all, until it learns a reasonable policy.

Rats, however, show great directionality in their movements while performing these types of tasks. Thus, three taxic modules were added to guide the animat to promising visual stimuli. These modules assume that the visual sensory information is fed to the dorsolateral striatum, which has been associated with cue responses and habitual learning [33]. Namely, our model's taxic behaviors are implemented as already learned habitual stimulus-response associations. However, the sensory information is also fed to the ventral striatum and affects value estimation, as explained below.

The first two modules guide the animat to flashing and non-flashing feeders. They are treated different to reflect the prior knowledge the animal has about flashing feeders. The third one guides the rat to obstacles' endpoints, providing a way to navigate through a maze of obstacles when no feeder is visible, nor previous knowledge of where to go is available.

All three modules make votes on each possible movement action depending on the expected reward of getting to the given visual stimulus. In addition, the votes are inversely proportional to the number of steps it would take the animat to reach the feeder or wall. Equation 7.7 summarizes this, where vr is a system parameter representing the expected reward associated with the visual stimulus, sr is the small negative reward given after each step to account for the motion effort and sn is the number of steps needed to reach that stimulus, computed from the angular and linear distances.

$$vote_a = vr + sr * sn \tag{7.7}$$

In addition to voting for each action, the feeder related modules contribute to the value estimation of a given place or situation. Namely, if the animat is seeing a flashing feeder, the value of that location will be increased by the expected value of going to the flashing feeder. Value is also estimated using a constant expectancy and the number of steps it would take to reach the interest point.

This allows for the animat to learn the positive outcome of an action that takes it to a "promising place" where a feeder is first observed, before receiving the actual reward.

In addition, this allows to detect the negative outcome of trying to eat from a feeder without success. It is the contrast between the high value estimated by the taxic module and the zero outcome what produces a high error signal (or decay in dopamine release) upon failure.

7.3.6 Exploration

In contrast to what is usually done with RL algorithms, there is not a continuous exploration drive built in into the model; the animat learns the proper actions by navigating using the taxic strategies and observing their outcomes.

However, some situations arise in which the system as a whole cannot propose any action. This may happen due to the lack of visual stimulus, or due to a negative value estimation of all action's outcomes by the actor critic or by a combination of both (the system only chooses positive valued actions).

In these cases, after the animat has been still for a certain number of simulation steps, an exploratory module takes over for a fixed and small number of steps, where the animat executes random actions.

7.3.7 Action Selection

Action selection is performed in a collaborative fashion through a voting mechanism, which instantiates the action selection performed by the globus pallidus.

Each module makes a weighted vote on the expected outcome of performing each action, as shown for the locale module in chapter 4. Then, all votes are added and the action with the most votes wins, in a winner take all fashion.

7.3.8 Place Cell Inactivation

The experiments with real rats involve inactivating the rat's hippocampus during the task. This was modeled by negatively modulating (partially inactivating) the activity of each cell in the layer corresponding to the inactivated region (septal or temporal).

Each cell is modulated by a value inversely proportional to the volume of cells between the injection site and the cell [160], as shown in Eq. 7.8. The quantity r corresponds to the distance between the cell and the injection site (the radius of the sphere).

$$modulation = 1 - \frac{c}{\frac{3}{4}\pi r^3} \tag{7.8}$$

The constant c was set to ensure $1mm$ radius of total inactivation [160].

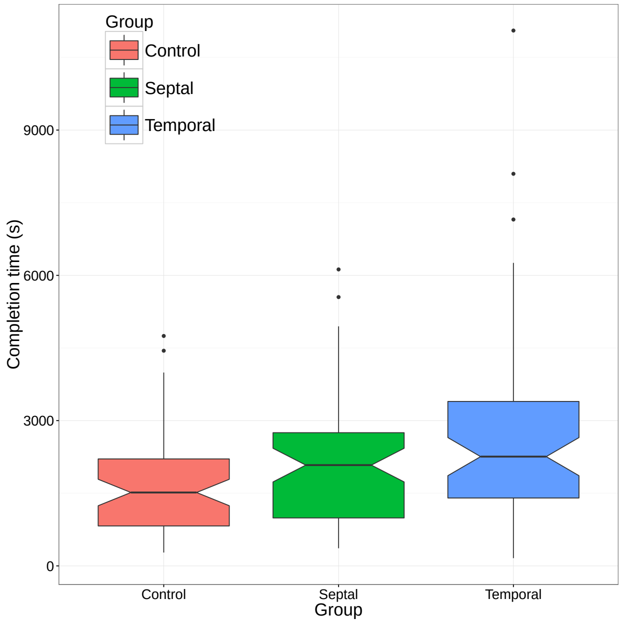


Figure 7.5: Completion times for the delayed cue with small obstacles for all three groups over 64 individuals. The septal portion of the HPC was inactivated in the Septal group and the temporal portion in the Temporal. The plot shows boxplots using the 1.5 IQ outlier criteria.

7.4 Results

Figure 7.5 shows a boxplot [81] of the completion times in simulated seconds for the delayed cued phase with small obstacles. The Temporal and Septal groups correspond to the temporal and septal portions deactivation, respectively.

A Kruskal-Wallis test [80] was performed over the data and significant differences were found $(p < 0.0001)$.

A Dunn test [42] post-hoc pairwise comparison was made. Significant difference was found between groups Control and Septal $(p < 0.05)$ and Control and Temporal $(p < 0.001)$. No significant difference was found between Septal and Temporal completion times.

Figure 7.6 shows sample paths traveled by the animat. The delayed cue paths were chosen from the animat with the median completion time among its group.

In order to assess the importance of some of the system parameters we performed calibration experiments, running the model with different possible values in the parameter's space.

Figure 7.7 shows the evolution of completion times for different values of the eligibility traces decay system parameter. Each panel shows the resulting completion times for a given value of the

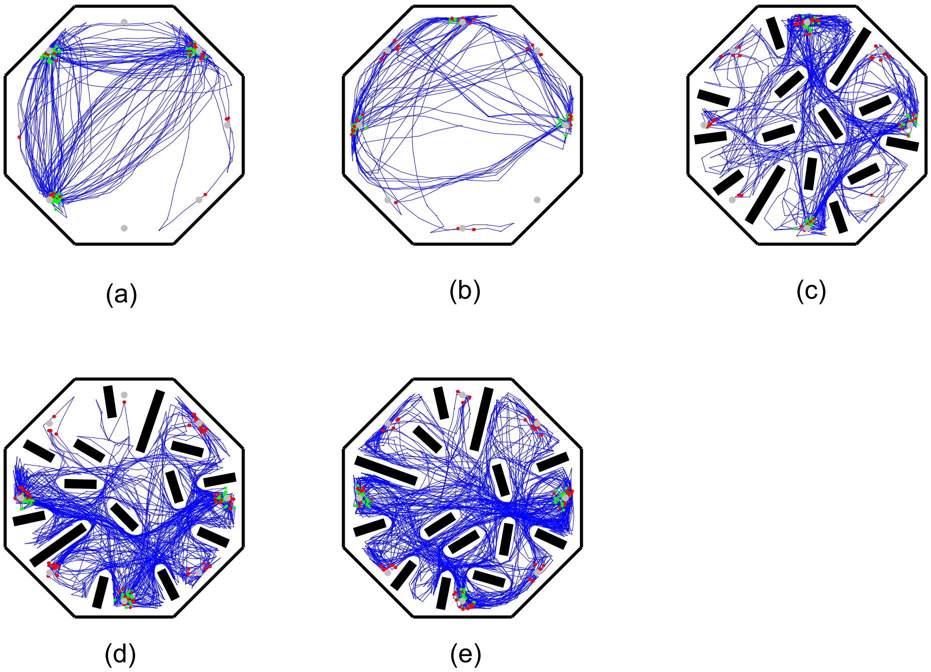


Figure 7.6: Sample paths followed by the rats for different task phases and groups. The green dots signal a successful eat attempt, whereas the red ones signal an unsuccessful one. (a) Shows a typical training (non-delayed cue – see The Task subsection) session and (b) a typical delayed cue one. (c) a delayed cue with obstacles session for the control group, (d) for the septal group and (e) for the temporal group. Delayed cue paths were chosen from the individual with the performance closest to its group median.

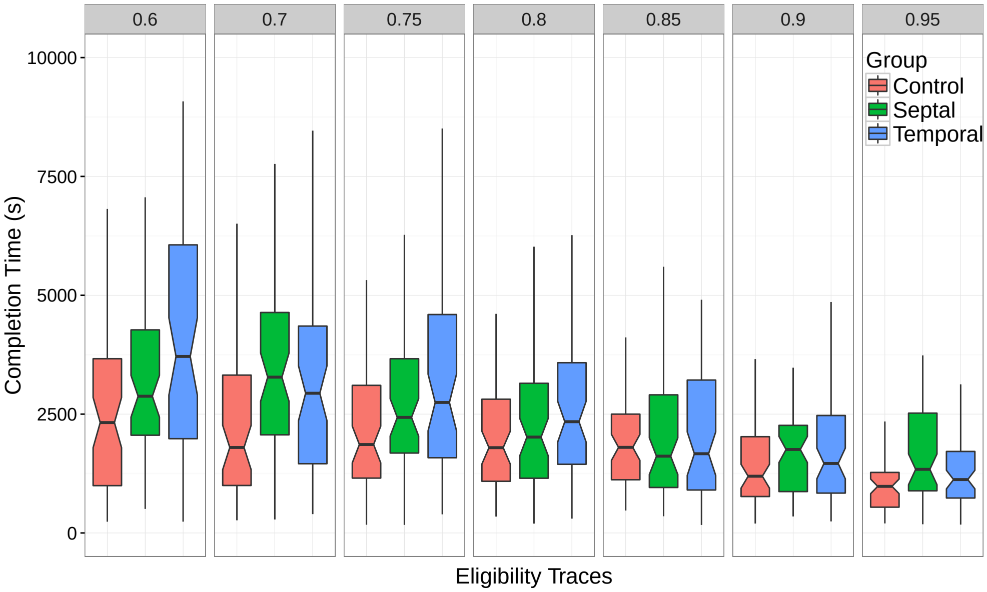


Figure 7.7: The completion times in seconds for the delayed cue phase with small obstacles while varying the eligibility traces decay parameter (top x axis).

parameter. As eligibility traces decay faster (lower values), all groups performance decreases. In addition, as the traces decay faster, the difference between groups becomes more apparent. The ventral group is notably more impaired under fast decaying traces conditions.

Figure 7.8 show the completion times for the control group in the delayed cue phase (both with and without obstacles). The model was tested with obstacle and place field interaction (Wopfi) and with no obstacle place-field interaction (Nopfi). In the no-obstacles condition, performance was not impaired. Both Wopfi and Nopfi groups could accomplish the task, with the exception of one individual in the Wopfi that timed out the task. In the presence of obstacles, however, the groups differentiated. The Nopfi group had many individuals that were able to complete the task, but also had a great number of individuals that timed out.

7.5 Discussion

This model shows how a differentiation of functions along the longitudinal axis of the hippocampus could explain the differences in performance seen after transient inactivation in this task. The septal portion of the hippocampus is attributed the function of action selection while the temporal portion is attributed the function of mapping places to value.

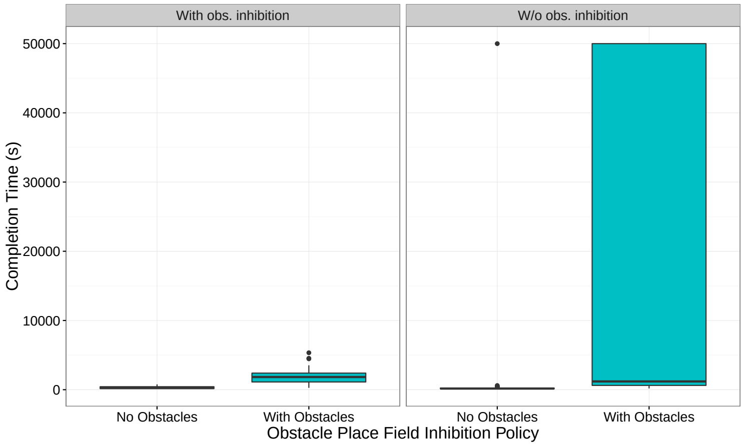


Figure 7.8: Completion times for the delayed cue with small obstacles with and without place field and obstacle interaction.

Obtained results are consistent with those seen in rats using the same protocol [27, 28], where both septal and temporal inactivation impair the animal's performance, being the latter group the most impaired.

7.5.1 Temporal Hippocampus and Spatial Learning

The general consensus has been that septal hippocampus is essential for spatial learning, whereas temporal is not [103, 104]. However de Hoz et al. [30] used a slightly modified version of the water maze protocol in which learning was spaced out to more days, each one containing fewer trials per day than usual (4 trials/day vs 8 trials/day). As a result, they saw no difference in performance between septal and temporal spare groups, suggesting that temporal involvement may depend on specific features of the learning protocol. This hypothesis, nonetheless, contradicts the findings of Bannerman et al. [8, 9], who carried a similar spaced protocol and arrived at the conclusion that temporal hippocampus was still not involved in navigation. Additionally, septal, and not temporal, hippocampus lesioned rats have shown to differentiate from control ones in a radial arm maze with a spaced out learning protocol (1 trial/day) [125]. These different and contradicting results have not been explained yet.

Prior research has shown that temporal HPC inactivation impairs inter-trial learning when trials are given within lapses of minutes [127]. Additionally, it has been shown that unilateral lidocaine inactivation of the temporal hippocampus affects both reference memory (going to non-baited arms) and working memory (going twice to the same arm) in a radial maze [126]. However, this contradicts the NMDA based injury results that report no difference between control and temporally injured groups in the same task [125]. Although reversible inactivation results provide the flexibility of controlling the phase in which the structures are not active, results can sometimes be difficult to interpret due to the fact that they also inactivate passing fibers [104].

Experiments carried out by de Hoz et al. [31] showed that the temporal portion involvement is more apparent when the rat has to relearn the location of a hidden platform. Septal lesions made after learning do not impair the rats' ability to learn a new location of the platform, implying that after learning has been acquired, the temporal hippocampus alone can trigger the retrieval and use of those spatial memories.

It is also worth noticing that some evidence shows that both the septal and temporal portions of the hippocampus are required to retrieve previously learned spatial memories [88]. Although these results are also based on lidocaine inactivation, the authors show in the same work how septal, but not temporal, training inactivation of the hippocampus impairs learning, consistent with Moser and Moser [104] results. Thus, they arrived to the conclusion that septal hippocampus is more important for task acquisition, while both portions are important for memory retrieval when the task has been learned with an intact hippocampus [88].

Consider the value of a place to be the expected reward to be obtained by the rat, if starting its route from that place. Then, if a correct "value map" is learned for a certain task, the animal can learn from each individual action, not only from the reward related ones. This is in fact, the basic principle of reward propagation. For example, from the unconditioned stimulus to the conditioned one, or from the rewarding place to the previous locations in the path to it. The presented model attributes the learning of this value map to the temporal region of the hippocampus. Then, its inactivation will prevent the learning of new maps upon changes in the environment, interfering with flexible re-learning locale behaviors as reported by Poucet et al. [127] and Hoz et al. [31]. If these assumptions are correct, we would expect that the inactivation of the ventral striatum, the nucleus accumbens, would be equivalent to the inactivation of the temporal hippocampus, with respect to performance degradation in these tasks.

To consider the consequences of septal but not temporal involvement evidence, one must first think about parallel learning processes. The evidence of temporal involvement discussed so far is focused on tasks that need fast relearning of the relation between the environment and the action to perform. Septal involvement evidence, however, involves more general setups, which include sleep or more trials per session (if comparing Moser et al. 1995 with de Hoz et al. 2003). It has been shown that rats learn while asleep, and that the disruption of sleep sharp-wave ripples impair navigation performance [44, 56, 169]. This means that rodents learn all these tasks using at least two mechanisms, i.e. learning while performing the task (awake) and while resting (asleep). Additionally, more density of trials per day might facilitate the use of temporary information acquired that is not going to be consolidated to the following day.

These alternative learning processes could compensate for the value estimation provided by the projection of the temporal hippocampus to the nucleus accumbens in a normal Morris maze, explaining many of the obtained results [103, 104]. However, when fast flexible re-learning is required [28, 31, 127], the large scale generalization power provided by the temporal place cells is needed. The modeled task falls into this category, as the recall phase involves re-learning the acquired policy to account for the introduced obstacles.

A consequence of this view would be the prediction that activity in the dopaminergic centers during learning would propagate faster through space in the presence of an intact temporal hippocampus. Reinforcement learning has been used in many models as a framework to explain the learning process [4, 20, 39, 45, 52, 86, 21]. However, little has been discussed related to the implications of casting this problem as reinforcement learning. These types of learning need for an error signal, which is usually attributed to dopamine phasic activity. In the context of normal conditioning, this signal would shift in time from the rewarding stimuli to the conditioned one as learning goes on. In the navigational context, this signal would spatially move from the rewarding location backwards to distant places. Then, we would expect this propagation to be slower or null if the temporal hippocampus is inhibited.

All but one of the protocol discussed (Pucet et al. [127]) so far use training throughout different days. The protocol used in this work marginalizes this out, by doing learning and recall within the same day, without any significant resting period [27, 28]. As with Poucet et al. [127], in experiments by Contreras et al. [27, 28] the temporal hippocampus was shown to be as important as the septal portion. Additionally, it is consistent with the results of Loureiro et al. [88], as inactivation occurs after learning. In our model, we cast the problem as a reinforcement learning, splitting the roles of action selection and value estimation. We account for the effects seen by Contreras et al. [27, 28]

and we would expect our model to account for Poucet et al. [127], though we do not include the latter experiment in this work. For the other experiments, our model would have to be enhanced with offline sleep-driven learning to explain the observations, and we leave that as our main line of future work.

7.5.2 Eligibility Traces

The analysis of the impact of eligibility traces shows degradation in performance as their learning power decreases. Eligibility traces are explained by means of increasing the effectiveness of updates. They are equivalent to reactivating all previously active states upon reaching a reward, consequently updating a larger portion of the place-value map [155]. Connected to this, disruption of replay events during sharp-wave ripples has shown to impair navigation performance [44, 56]. Also, backwards replay events are found upon reaching rewarding sites [49], which resembles eligibility traces in an even greater degree.

More interestingly, a closer look to the results of decreasing eligibility traces strength shows increasing involvement in the temporal portion in the task. That is, as eligibility traces are diminished the temporal portion becomes more important. This can be interpreted in two different ways.

One way is that there are two complementary learning processes taking place. One process, independent of traces, relies on the temporal hippocampus to propagate reward (in the form of value) through space. Another process learns relying on traces. When the power of eligibility traces is diminished, the first process becomes more important, thus making the temporal inactivation effect more pronounced. This hypothesis could explain why dorsally spared animals are still able to learn simple navigational tasks, even under the assumption that temporal hippocampus alone drives value estimation structures: backward replay events that trigger upon getting to a reward [49] and sleep replay events would reproduce the entire route in a time span suitable for synaptic plasticity. Thus, no value propagation would be needed while the traces are intact. Under this interpretation, this model would predict that SWR disruption would increase the involvement of the temporal hippocampus in the learning process.

Another interpretation involves eligibility traces learning efficacy being mediated or facilitated by the temporal portion of the hippocampus. Then, as eligibility traces loose their power, the loss of temporal hippocampus activity has a greater associated effect. This could involve the dopaminergic structures activating during replay events [169] facilitating faster value propagation through space. This hypothesis is consistent with the fact that temporal hippocampus becomes more involved as the protocol spaces out trials, maybe giving offline consolidation a greater role [30]. This could also be due to a coordination between the longitudinal axis, allowing a greater reach of replay events [46], which we do not include in the present model.

7.5.3 Obstacle Place Cell Inhibition

After taking out obstacle based place field inhibition, the performance of the delayed cue with obstacles phase drops. This is done without changing the performance of delayed cue without obstacles phase, which means no general degradation in performance was introduced by the changes made to the model.

Obstacle based place field inhibition provides two navigational advantages. In the first place, the disappearance of previous normal cells and the appearance of new obstacle cells disrupt the learned policy. This avoids the rat trying to execute a policy that is no longer consistent with the environment, because an obstacle is now there. Secondly, by allowing cells to fire only one side of the obstacle, previously learned value does not propagate to regions that are no longer closer to the place cell center. The introduction of the obstacle changes the distance the animal must travel from one point to the other, and it is useful that the value mapping changes as well.

CHAPTER 8

BIO-INSPIRED GRAPH-BASED NAVIGATION ALGORITHM

8.1 Introduction

This dissertation has focused on model-free reinforcement learning models and algorithms for navigation. Although larger scales of representation have shown to speed-up learning, model-free algorithms can be still too slow for practical purposes. This chapter focuses on exploring a model-based algorithm inspired in the way rodents represent their current location, build a topological model of the world and use it to plan future paths. It does not explore the topic of multiple scales of representation, focusing instead on other orthogonal concepts developed in chapter 3.

The algorithm consists on the creation of a multi-query sensor-based motion planning algorithm, which builds upon algorithms like the bug algorithms, e.g. Bug1 and Bug2 [24]. The first path would be carried out by the underlying bug algorithm, while building a sparse representation of the the environment in the form of a connectivity graph. This graph would then be used in subsequent queries to improve the generated path, while keeping the convergence properties.

We call the algorithm "Experience Roadmap", as it consists of building a roadmap from experience, using only local sensor information and a localization system.

8.2 Experience Roadmap

The proposed method consists of building upon robust algorithms, like the bug algorithms, to turn them into multi-query algorithms while keeping memory and processing footprints low.

In summary, the method consists of building a graph that captures the navigability directly from experience. This graph can be used then to optimize further trajectories. The step of building a dense map (e.g. an occupancy grid) is skipped.

The rest of this section explains the inspiration of the algorithm from biology, the graph creation process and how the algorithm uses the graph to optimize the second query.

8.2.1 Biological Basis for the Algorithm

When facing a new task, rodents try a set of uninformed strategies that allow them to discover possible solutions. Later on, the control is shifted to a system that uses the acquired knowledge of the environment to make more informed choices [164].

This system can be implemented as a model-free decision making one [4, 86, 16, 39, 22], or a model-based one [45, 67, 11, 10, 12].

Model-based decision making has gained some traction in the past years, due to the recent discoveries that support its use in the rodent brain [23, 67, 52, 94, 77, 45, 46]. It has been shown that place cells, previously considered to be just localization units, connect to each other as a consequence of experience [40]. Then, a topological graph of the environment is stored in the rat's hippocampus, much like Tolman predicted [164].

Furthermore, it has been shown that when a rat is at rest, the hippocampus shows periods of preplay [43, 122, 68]. In these episodes, cells fire as if a signal was being propagated from the current node through the topological graph, towards the desired goal. Recent results have shown that there is a correlation between these simulated paths and the paths actually taken by the rodent afterwards [123].

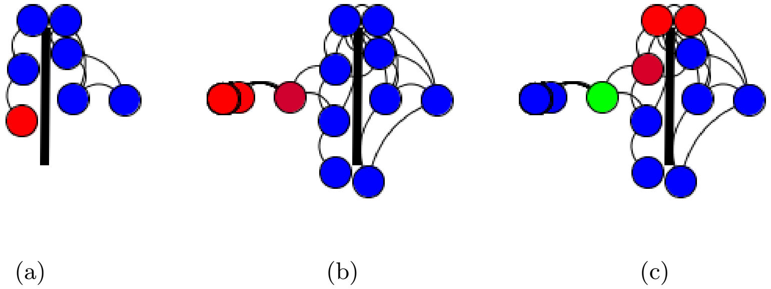
We draw inspiration from this hypothesized mechanism to improve robot navigation under the assumption of a known localization and a lack of a dense map. We recruit nodes on the graph following previous work on place cell recruitment models, we connect them based on experience and use graph search to find improved paths upon reiterative queries.

8.2.2 Algorithm Overview

Algorithm 1 shows the pseudo code of a single iteration of the Experience Roadmap system.

First, the set of active nodes are found. Each node has a preferred location, and its activation depend on whether the agent is close enough to it. The sensor readings, e.g. distance sensors, are also used to create a local map. A node is only active if there is no obstacle between the agent and the node's preferred location. This is important because it prevents links from forming across walls, which would not reflect the navigability of the environment.

Then, all nodes that have mutual visibility are connected to each other. This step is very important, to allow two different paths to connect to each other, even if the the agent does not



(a) (b) (c)

Figure 8.1: Snapshots of the graph built by the algorithm. (a) A partially built graph of the environment. The currently active nodes are shown in red. (b) The whole graph after the agent arrives to the goal (c) The graph being used for navigation: red nodes represent the currently active ones, while the green node is the currently pursued one.

navigate from one to the other. This is one of the main improvements over the World Graph Layer discussed in chapter 2.

If no node is active, a new node is generated with the current position as it's preferred one.

Figure 8.1a shows the graph being built while the agent traverses an environment with one wall. The nodes marked in red are the currently active ones.

Finally, the underlying navigation algorithm is executed. If there is a valid path in the graph from the currently active nodes to the goal, the next node's position in the path is used as a sub-goal for the algorithm. The next node is defined as the first node in the path that is not currently active. If no such path is available (e.g. the graph is not yet complete), the goal position is passed through to the underlying navigation algorithm. Figure 8.1c shows the graph being used for navigation. The green node's position is passed to the underlying bug algorithm as an intermediate goal to guide the agent towards the goal.

8.3 Experiments

We tested the algorithm in two types of environments. The first type consists of well-known scenarios that are used to test motion algorithms for efficiency and completeness. Figure 8.2 shows all the maps used. The second type consists of maps with randomly placed obstacles. These maps allow to test the algorithm's robustness under conditions not known during it's design and implementation. Figure 8.3 shows some example random maps.

```
function navigate(currentPose, goalPose, sensorReadings, graph):
        nodes = getActiveNodes(currentPose, sensorReadings)
        graph.link(nodes)

        if nodes.isEmpty():
                n = createNode(currentPose)
                nodes.add(n)

        if (graph.canReachGoal(nodes, goalPose)):
                subGoal = graph.getNextNode(nodes, goalPose).position
                bugAlg(currentPose, subGoal, sensorReadings)
        else:
                bugAlg(currentPose, goalPose, sensorReadings)
```

Algorithm 1: One iteration of the Experience Graph Path Planning algorithm.

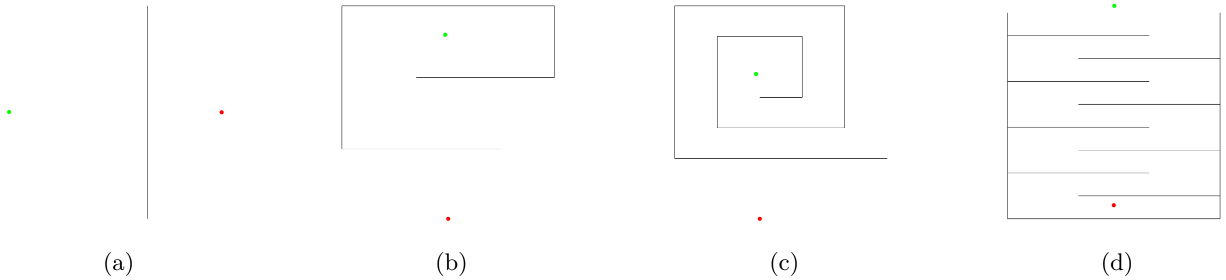


Figure 8.2: Well-known mazes used to test algorithms for efficiency and completeness. Black lines denote walls. The red dot points the start and the green one the goal.

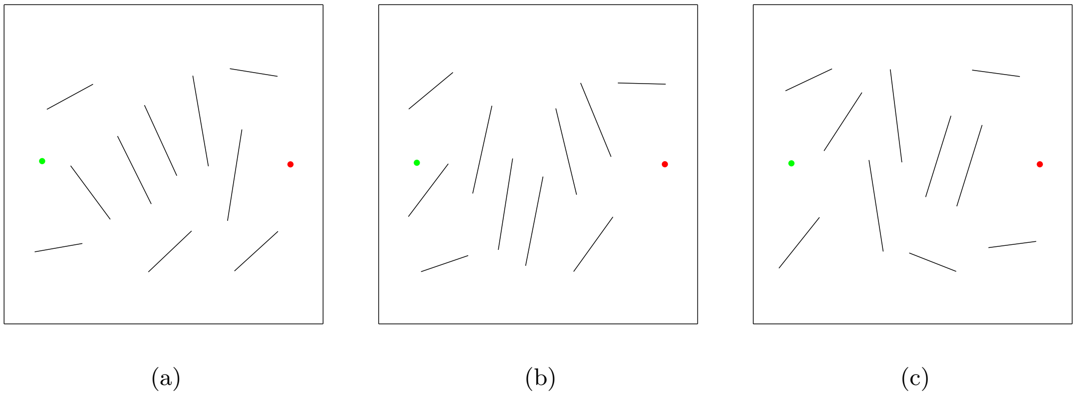


Figure 8.3: Randomly generated mazes to test unknown conditions. Black lines denote walls. The red dot represents the start and the green one the goal.

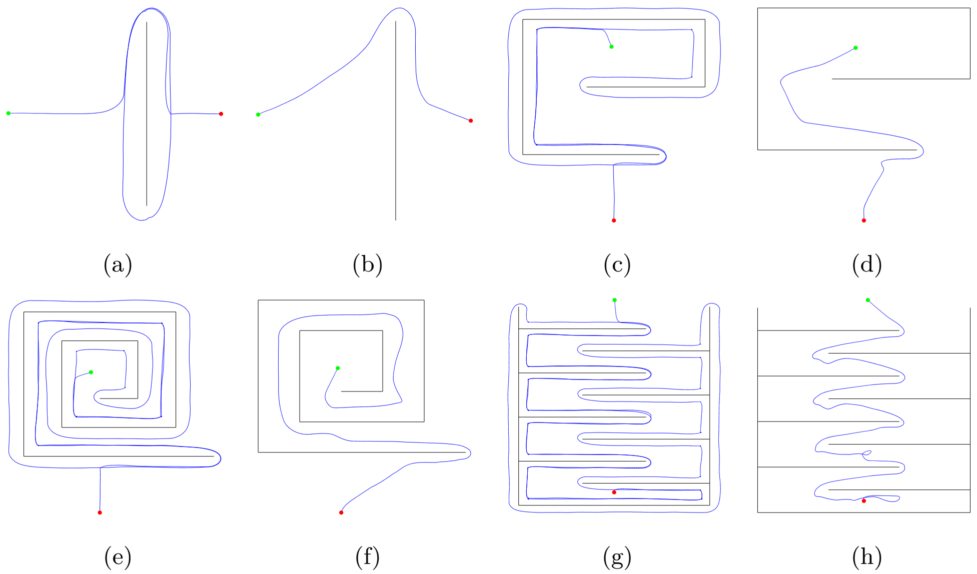


Figure 8.4: Results for the first (left) and second (right) run for each map. Black lines denote walls. The red dot represents the start and the green one the goal. The blue line represents the path.

8.4 Results

8.4.1 Qualitative Results on Well-Known Maps

Figure shows 8.4 the path for the first and second iteration over each map.

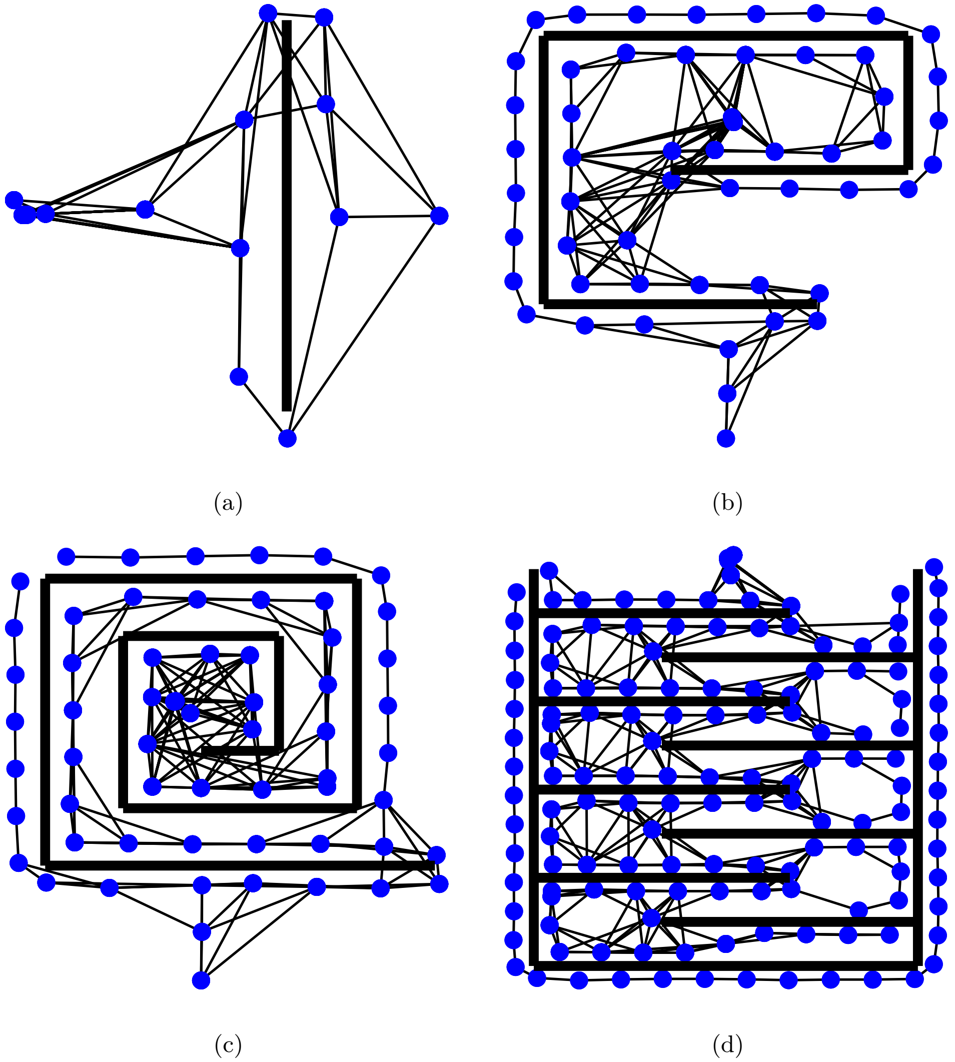
Figure 8.5 shows sample graphs created by the algorithm on these environments. It is worth noticing two aspects of the generated graphs:

- The graph captures the environment's connectivity, connecting only nodes that can be reached in the short term (i.e. no connections through large walls)

- Near some wall ends, some links do intersect the wall (see top of Figure 8.5a). They are generated when the agent is "seeing" both sides of the wall from some point A, so nodes from both sides of the wall are active and get connected to each other. This link is not an error though, because those two nodes are navigable in the short term, by going through point A.

8.4.2 Quantitative Results on Random Maps

Figure 8.6 shows the runtime in simulation steps for run zero (only bug algorithm), and the first and second runs (using the experience roadmap).



(a)

(b)

(c)

(d)

Figure 8.5: Sample graphs built by the algorithm in the example environments.

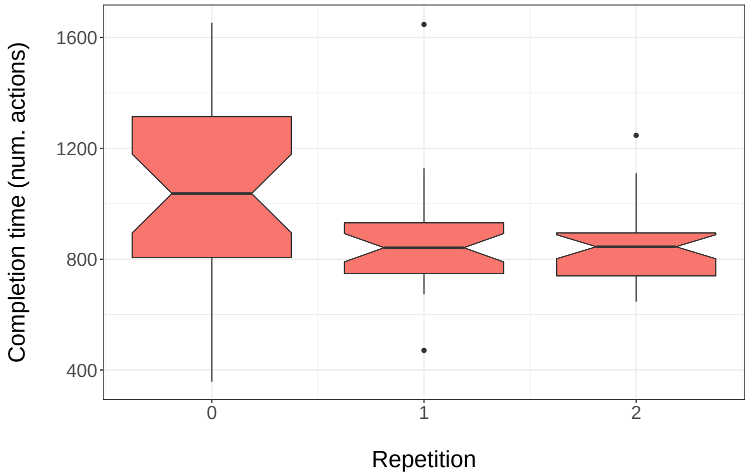


Figure 8.6: Time to reach the goal in simulation steps. Repetition 0 uses the underlying bug algorithm alone, while repetitions 1 and 2 use the created experience roadmap.

An Kruskal-Wallis test and Dunn post-hoc test were carried out to assess the significance of the difference between runtimes for the different repetitions. Significant difference was found between repetition 0 and 1 (p = 0.0121) and 0 and 2 (p = 0.0029).

Figure 8.7 shows example paths for repetitions 0, 1 and 2, for different randomly generated maps.

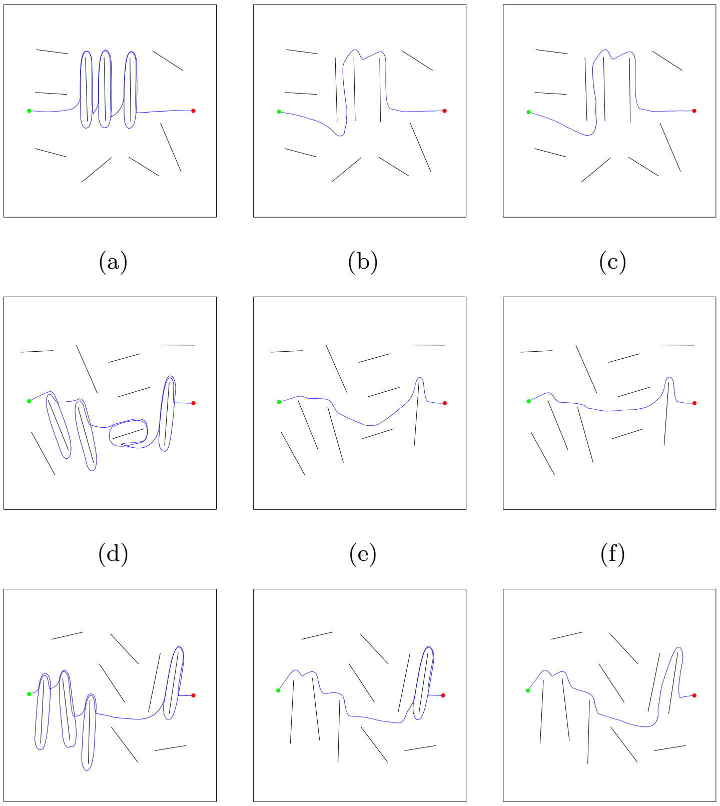
Figure 8.8 shows sample graphs generated over random maps, after repetition 0 and 1. Here one can observe how the graph improves after the repetition 1, which explains the slight improvement on repetition 2.

8.5 Discussion

A bio-inspired algorithm for path planning was developed and tested. This algorithm builds a roadmap directly from experience using a localization system, avoiding the need to generate a dense map of the environment and builds a roadmap from it. Thus, it lowers the memory and computation footprint.

The generated map goes beyond similar prior concepts like the world graph layer [12, 10, 59] by linking nodes that are locally visible, rather than just linking the traversed path.

The graph can be used after the first repetition to guide the underlying navigation algorithm through a shorter route.


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8.7: Example paths for repetition 0 (only bug algorithm - left), 1 (experience roadmap - mid) and 2 (experience roadmap - right). Black lines denote walls. The red dot represents the start and the green one the goal. The blue line represents the path.

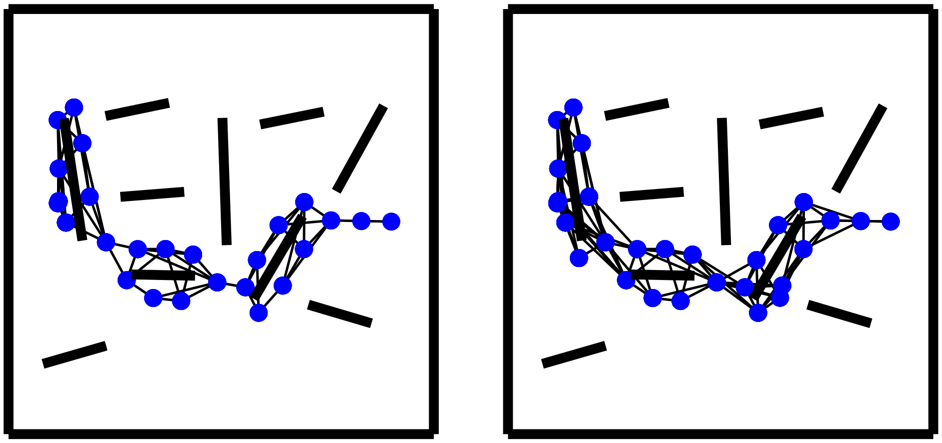

(a) (b)

Figure 8.8: Sample graphs built by the algorithm on random maps, after repetition 0 (a) and 1 (b).

8.5.1 Well-Known Environments

The qualitative results over well-known environments show that the algorithm is able to extract a suitable model in the first run, allowing it to perform a much better path on the second run. None of these maps show convergence issues when using the experience roadmap, thus there is no evidence that the use of this algorithm implies the loss of the completeness of the underlying bug algorithm.

8.5.2 Random Wall Environments

The quantitative results show that there is a significant decrease in the number of actions needed to reach the goal, between repetition 0 and the others. This means that the algorithm is effectively optimizing the route with respect to the bug algorithm. Even though there is no significant difference between repetition 1 and 2, it seems there is a trend of the algorithm to keep optimizing the path, by improving the model of the environment in the first run. This is exemplified by Figures 8.7 g-i.

8.5.3 Comparison to the State of the Art

When comparing to the state of the art of path planning (see chapter 2), one has to consider that this algorithm is working under the assumption of no dense map of the environment. Then, it must be compared against sensory based algorithms, i.e. bug algorithms, APFs and their variations. The results show that the Experience Roadmap is as complete as the underlying bug algorithms, while optimizing its route on the second and consecutive runs on the same environment. Thus, the Experience Roadmap algorithm does not suffer from the local minima problem linked to APFs, while it produces shorter paths than the complete bug algorithms.

Many variations of APF's and bug algorithms focus on improving the efficiency of completeness of the base algorithms (see chapter 2). We argue that any of those improvements can be used as underlying algorithms for the Experience Roadmap. It will then improve on the first run, when only the underlying algorithm is running, while still helping it on subsequent runs by means of guiding it through a more globally optimal route. One must also consider that there are some scenarios that can only be solved efficiently by using some sort of model. Consider the environment shown in Figure 8.9. When the agent reaches the first intersection, it has to decide whether to turn right or left. The sensory information will be symmetric for the left and right side, and the agent has no

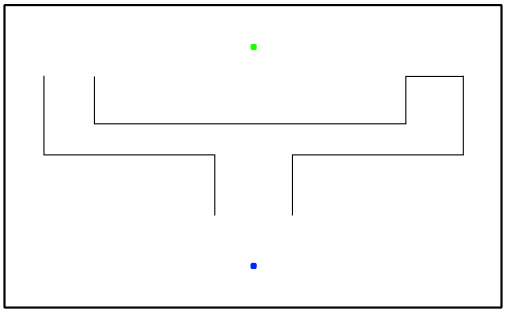


Figure 8.9: An environment that is impossible to solve efficiently with only reactive strategies. The agent has to go from the blue dot to the green one.

way of knowing that the right turn will lead to a dead end. Any strategy that is based on reactive behavior will always fail to make the right choice in either this environment or the opposite one (dead end on the left). Even in subsequent executions over the same environment, the algorithm will make the same choice. When using the Experience Roadmap, the optimal choice will be made after the maze is navigated once.

8.5.4 Computational Resources

The gains obtained by the algorithm are supported by the creation of the roadmap graph. Consequently, using the algorithm will require some more computational power and memory than the required by the underlying bug algorithm. However, we argue that the number of nodes of the graph is bounded by the explored area. This is due to the fact that nodes are only created when there are no close enough nodes around, or when all close enough nodes are not visible. Thus, every node has an “area of influence” greater than 0, where no other nodes are going to be created. Then, once the whole explored area is covered with the node’s areas of influence, no new nodes are going to be created until the agent goes outside this area. It can be concluded that the amount of memory to store the graph is bounded by the explored area. Shortest path search is bounded by the number of nodes (n^2 for Dijkstra’s algorithm), which makes computational time also bounded by the explored area. The fact that both memory and computational time are bounded by the explored area, and not by execution time, means the algorithm is suitable for long term operation.

8.5.5 Future Work

Some work is still pending. The algorithm used to navigate the nodes, or waypoints, in the path could be improved. This would remove little artifacts from the robot trajectories, as shown in Figure 8.4h.

Additionally, the model is not connected properly in the first pass when sharp turns are executed, see Figure 8.5c (top-left).

CHAPTER 9

CONCLUSIONS

9.1 State of the Art of Modeling

Chapter 3 described the state of the art of modeling rodent navigation. It studied the topic of rodent navigation as a black box, laying the grounds of what rodents can do, and the proposed models of how they do it. The main conclusion drawn is that there are many aspects that remain to be properly modeled in the rodent navigation system: towards the directions of better use of all modalities of information; towards a better understanding of working memory; towards a better interaction between control systems; and towards a better understanding of the model free and model based components in learning and decision making.

9.2 Multi-Scale Representation

This dissertation has analyzed the topic of multiple scales of representation of an agent location. It has shed light into the advantages of having the agent location expressed as a redundant, multi-scale system. In few words, larger scales of representation allow for faster generalization of the policy or the value map through the environment, speeding up learning and increasing the final policy's coherence and coverage.

Experiments performed for the multi-scale RL system in the taxi problem (chapter 6) showed that larger scales allow for faster value propagation. For the continuous case, place cell radius has a direct impact in the system's performance. Too small cells don't allow for value to propagate, and the algorithm is not able to converge. Too large cells lead to suboptimal policies that prevent the agent to reach the reward in later episodes.

When different scales of representation are combined, larger scales seem to be critical for value propagation but can improve action selection as well. Also, given larger scales are present to propagate value backwards, smaller scales can be used for action selection.

9.3 Model of the New Task

This dissertation has introduced a new experimental task developed in collaboration with the Fellous laboratory. This task contributes a new insight of the involvement of the temporal hippocampus (large scale) in navigation. A set of models with increasing complexity have been introduced, with the aim to reach the goal of modeling this complex task. Chapter 4 introduced the general framework for multi-scale reinforcement learning used throughout the dissertation. Chapter 5 introduced a model of the Morris maze using multiple scales of representation. This model was able to profit from the larger scales of representation to learn a more complete and coherent policy faster. However, this model did not account for obstacles or dynamic environments. Chapter 6 introduced an algorithm for navigation learning inspired in the previous chapter. It builds upon the previous model: including a comparison with a widely used algorithm in a well known task; and incorporating obstacles in the map. The algorithm takes advantage on the current knowledge about the interaction of place fields and obstacles, to include a feature essential for its correct functioning. Chapter 7 introduces the final model for this dissertation. It attempts to model the effects seen with real rats in the newly developed task. It introduces a variety of novel modeling features: semi-dynamic environments; multiple goals; avoidance learning; goal cueing (flashing); obstacle and place-field interaction with normal and obstacle cells; and hippocampus place cell deactivation.

The final model was able to explain the observed results mainly in terms of two novel contributions: multiple scales of representation and an asymmetric contribution of them to action selection and value estimation. As with any model, some predictions were drawn from it, which will hopefully be empirically verified in the future: the ventral striatum deactivation should have a similar effect to temporal hippocampus inactivation, under the hypothesis of their involvement in value estimation; the temporal deactivation should have an effect of dopamine phasic activity in the VTA, slowing down or preventing the activity propagation backwards in space; and replay events disruption could increase the need for the temporal hippocampus in a given task.

9.4 Experience Roadmap

Finally, the knowledge of the rat navigation system acquired during the elaboration of this work, was used to design a novel path planning algorithm. The algorithm, Experience Roadmap, allows to improve the efficiency of sensory based algorithms while keeping the memory and computation footprint low, due to the avoidance of building a dense map of the environment. Results show the algorithm is able to improve the routes, length-wise, without disrupting the underlying algorithm's completeness.

9.5 Future Work

There are many lines of future work. On the modeling side, almost every dimension of the rat navigation system listed in chapter 3 is a future line of work. The main one would be to implement model-based decision components that allow for a smarter agent, bridging the gap between real rat completion times and the animat's.

Implementing a model that would be able to account for all the seemingly contradictory results regarding temporal hippocampus involvement in navigation would be a great advance for neuroscience.

One big shortcoming of the implemented models is the use of high-level reinforcement learning algorithms, such as actor-critic. Even though these algorithms model the net effect of a variety of neural centers operating in harmony, they do it coarsely and they prevent the models from making easy to test predictions. A neural implementation of these centers would allow modelers to predict the activity of small ensembles of participating neurons, making the models more useful and more verifiable.

For the multi-scale RL for navigation algorithm, the interaction between the learning component and external components could be improved. This way, learning systems can become more rational during exploration, similar to what was done with the taxic component. Being able to include external components to the system can also make it safer, as safety checks can be hardwired instead of being learned.

The Experience Roadmap also presents opportunities for improvement. The algorithm used for navigating the graph could be improved to account for dynamical aspects, making the robot movements smoother and the trajectories free of artifacts. Better algorithms than Bug1 can be implemented to improve the first run as well, as long as they are complete algorithms.

REFERENCES

[1] Alice Alvernhe, Tiffany Van Cauter, Etienne Save, and Bruno Poucet. Different CA1 and CA3 Representations of Novel Routes in a Shortcut Situation. *The Journal of Neuroscience*, 28(29):7324–7333, July 2008.

[2] Michael A. Arbib. Perceptual structures and distributed motor control. *Comprehensive Physiology*, 1981.

[3] Angelo Arleo, Cyril Déjean, Pierre Allegraud, Mehdi Khamassi, Michael B. Zugaro, and Sidney I. Wiener. Optic flow stimuli update anterodorsal thalamus head direction neuronal activity in rats. *The Journal of Neuroscience*, 33(42):16790–16795, 2013.

[4] Angelo Arleo, Fabrizio Smeraldi, and Wulfram Gerstner. Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 15(3):639–652, May 2004.

[5] Antónia Arszovszki, Zsolt Borhegyi, and Thomas Klausberger. Three axonal projection routes of individual pyramidal cells in the ventral CA1 hippocampus. *Frontiers in Neuroanatomy*, 8:53, 2014.

[6] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorization for robot learning. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, volume 3, pages 1502–1509 vol.3, November 1996.

[7] Bernard W. Balleine and John P. O'Doherty. Human and Rodent Homologies in Action Control: Corticostriatal Determinants of Goal-Directed and Habitual Action. *Neuropsychopharmacology*, 35(1):48–69, September 2009.

[8] D. M. Bannerman, M. Grubb, R. M. J. Deacon, B. K. Yee, J. Feldon, and J. N. P. Rawlins. Ventral hippocampal lesions affect anxiety but not spatial learning. *Behavioural brain research*, 139(1):197–213, 2003.

[9] D. M. Bannerman, B. K. Yee, M. A. Good, M. J. Heupel, S. D. Iversen, and J. N. P. Rawlins. Double dissociation of function within the hippocampus: a comparison of dorsal, ventral, and complete hippocampal cytotoxic lesions. *Behavioral neuroscience*, 113(6):1170, 1999.

[10] Alejandra Barrera, Alejandra Cáceres, Alfredo Weitzenfeld, and Victor Ramirez-Amaya. Comparative Experimental Studies on Spatial Memory and Learning in Rats and Robots. *Journal of Intelligent & Robotic Systems*, September 2011.

[11] Alejandra Barrera, Gonzalo Tejera, Martin Llofriu, and Alfredo Weitzenfeld. Learning spatial localization: from rat studies to computational models of the hippocampus. *Journal of Spatial Cognition and Computation*, 15(1):27–59, 2015.

[12] Alejandra Barrera and Alfredo Weitzenfeld. Biologically-inspired robot spatial cognition based on rat neurophysiological studies. *Autonomous Robots*, 25(1-2):147–169, January 2008.

[13] H. Benbrahim, J.S. Doleac, J.A Franklin, and O.G. Selfridge. Real-time learning: a ball on a beam. In *, International Joint Conference on Neural Networks, 1992. IJCNN*, volume 1, pages 98–103 vol.1, June 1992.

[14] Hugh Carlton Blodgett. The effect of the introduction of reward upon the maze performance of rats. *University of California publications in psychology*, 1929.

[15] Aaron M Bornstein and Nathaniel D Daw. Multiplicity of control in the basal ganglia: computational roles of striatal subregions. *Current Opinion in Neurobiology*, 21(3):374–380, June 2011.

[16] Michael A. Brown and Patricia E. Sharp. Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens. *Hippocampus*, 5(3):171–188, 1995.

[17] Vegard Heimly Brun, Trygve Solstad, Kirsten Brun Kjelstrup, Marianne Fyhn, Menno P. Witter, Edvard I. Moser, and May-Britt Moser. Progressive increase in grid scale from dorsal to ventral medial entorhinal cortex. *Hippocampus*, 18(12):1200–1212, 2008.

[18] N. Brunel and O. Trullier. Plasticity of directional place fields in a model of rodent CA3. *Hippocampus*, 8(6):651–665, 1998.

[19] Neil Burgess, Andrew Jackson, Tom Hartley, and John O'Keefe. Predictions derived from modelling the hippocampal role in navigation. *Biological Cybernetics*, 83(3):301–312, August 2000.

[20] Neil Burgess, Michael Recce, and John O'Keefe. A model of hippocampal function. *Neural networks*, 7(6):1065–1081, 1994.

[21] K. Caluwaerts, M. Staffa, S. N'Guyen, C. Grand, L. Dollé, A. Favre-Félix, B. Girard, and M. Khamassi. A biologically inspired meta-control navigation system for the Psikharpax rat robot. *Bioinspiration & Biomimetics*, 7(2):025009, June 2012.

[22] Ricardo Chavarriaga, Thomas Strösslin, Denis Sheynikhovich, and Wulfram Gerstner. A computational model of parallel navigation systems in rodents. *Neuroinformatics*, 3(3):223–241, 2005.

[23] Fabian Chersi and Giovanni Pezzulo. Using hippocampal-striatal loops for spatial navigation and goal-directed decision-making. *Cognitive Processing*, 13(1):125–129, July 2012.

[24] Howie M. Choset. *Principles of robot motion: theory, algorithms, and implementation.* MIT press, 2005.

[25] David Coleman, Ioan A. Sucan, Mark Moll, Kei Okada, and Nikolaus Correll. Experience-Based Planning with Sparse Roadmap Spanners. *arXiv:1410.1950 [cs]*, October 2014. arXiv: 1410.1950.

[26] D. C. Conner, A. A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings*, volume 4, pages 3546–3551 vol.3, October 2003.

[27] M. Contreras, T. Pelc, M. Llofriu, A. Weitzenfeld, and J. M. Fellous. Effect of dorsal or ventral hippocampus inactivation on goal-directed spatial navigation in rats and computational models,. Washington, DC, 2014.

[28] M. Contreras, T. Pelc, M. Llofriu, A. Weitzenfeld, and J. M. Fellous. Ventral hippocampus inactivation impairs goal-directed spatial navigation in obstacle-laden environments. Washington, DC, 2015.

[29] R. Daily and D. M. Bevly. Harmonic potential field path planning for high speed vehicles. In *2008 American Control Conference*, pages 4609–4614, June 2008.

[30] Livia de Hoz, Jane Knox, and Richard GM Morris. Longitudinal axis of the hippocampus: both septal and temporal poles of the hippocampus support water maze spatial learning depending on the training protocol. *Hippocampus*, 13(5):587–603, 2003.

[31] Livia de Hoz and Stephen J. Martin. Double dissociation between the contributions of the septal and temporal hippocampus to spatial learning: The role of prior experience. *Hippocampus*, 24(8):990–1005, August 2014.

[32] Paul De Saint Blanquat, Vincent Hok, Etienne Save, Bruno Poucet, and Franck A. Chaillan. Differential role of the dorsal hippocampus, ventro-intermediate hippocampus, and medial prefrontal cortex in updating the value of a spatial goal. *Hippocampus*, 23(5):342–351, May 2013.

[33] Bryan D. Devan and Norman M. White. Parallel Information Processing in the Dorsal Striatum: Relation to Hippocampal Function. *The Journal of Neuroscience*, 19(7):2789–2798, April 1999.

[34] D. M. Diamond, M. Fleshner, N. Ingersoll, and G. M. Rose. Psychological stress impairs spatial working memory: relevance to electrophysiological studies of hippocampal function. *Behavioral Neuroscience*, 110(4):661–672, August 1996.

[35] David M. Diamond, Collin R. Park, Karen L. Heman, and Gregory M. Rose. Exposing rats to a predator impairs spatial working memory in the radial arm water maze. *Hippocampus*, 9(5):542–552, January 1999.

[36] Kamran Diba and György Buzsáki. Forward and reverse hippocampal place-cell sequences during ripples. *Nature Neuroscience*, 10(10):1241–1242, October 2007.

[37] Anthony Dickinson. Actions and habits: the development of behavioural autonomy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 308(1135):67–78, 1985.

[38] Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *arXiv:cs/9905014*, May 1999.

[39] Laurent Dollé, Denis Sheynikhovich, Benoît Girard, Ricardo Chavarriaga, and Agnès Guillot. Path planning versus cue responding: a bio-inspired model of switching between navigation strategies. *Biological Cybernetics*, 103(4):299–317, October 2010.

[40] George Dragoi, Kenneth D Harris, and György Buzsáki. Place Representation within Hippocampal Networks Is Modified by Long-Term Potentiation. *Neuron*, 39(5):843–853, August 2003.

[41] Yong Duan, Baoxia Cui, and Huaiqing Yang. Robot navigation based on fuzzy RL algorithm. In Fuchun Sun, Jianwei Zhang, Ying Tan, Jinde Cao, and Wen Yu, editors, *Advances in Neural Networks - ISNN 2008*, number 5263 in Lecture Notes in Computer Science, pages 391–399. Springer Berlin Heidelberg, January 2008.

[42] Olive Jean Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252, 1964.

[43] David Dupret and Jozsef Csicsvari. Reorganization of Hippocampal Place-Selective Patterns During Goal-Directed Learning and Their Reactivation During Sleep. In Masami Tatsuno, editor, *Analysis and Modeling of Coordinated Multi-neuronal Activity*, number 12 in Springer Series in Computational Neuroscience, pages 131–146. Springer New York, 2015.

[44] Valérie Ego-Stengel and Matthew A. Wilson. Disruption of ripple-associated hippocampal activity during rest impairs spatial learning in the rat. *Hippocampus*, 20(1):1–10, 2010.

[45] Uğur M. Erdem and Michael Hasselmo. A goal-directed spatial navigation model using forward trajectory planning based on grid cells. *European Journal of Neuroscience*, 35(6):916–931, March 2012.

[46] Uğur M. Erdem and Michael E. Hasselmo. A biologically inspired hierarchical goal directed navigation model. *Journal of Physiology-Paris*, 108(1):28–37, February 2014.

[47] Michael S. Fanselow and Hong-Wei Dong. Are The Dorsal and Ventral Hippocampus functionally distinct structures? *Neuron*, 65(1):7, January 2010.

[48] Stan B. Floresco, Christopher L. Todd, and Anthony A. Grace. Glutamatergic Afferents from the Hippocampus to the Nucleus Accumbens Regulate Activity of Ventral Tegmental Area Dopamine Neurons. *The Journal of Neuroscience*, 21(13):4915–4922, July 2001.

[49] David J. Foster and Matthew A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, March 2006.

[50] Antonella Gasbarri, Mark G. Packard, Elena Campana, and Claudio Pacitti. Anterograde and retrograde tracing of projections from the ventral tegmental area to the hippocampal formation in the rat. *Brain Research Bulletin*, 33(4):445–452, 1994.

[51] C. Gaskett, L. Fletcher, and A Zelinsky. Reinforcement learning for a vision based mobile robot. In *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings*, volume 1, pages 403–409 vol.1, 2000.

[52] P. Gaussier, A. Revel, J. P. Banquet, and V. Babeau. From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological Cybernetics*, 86(1):15–28, January 2002.

[53] Maya Geva-Sagiv, Liora Las, Yossi Yovel, and Nachum Ulanovsky. Spatial cognition in bats and rats: from sensory acquisition to multiscale maps and navigation. *Nature Reviews Neuroscience*, 16(2):94–108, February 2015.

[54] John Gigg. Constraints on hippocampal processing imposed by the connectivity between CA1, subiculum and subicular targets. *Behavioural Brain Research*, 174(2):265–271, November 2006.

[55] J. A. Gingerelli. Preliminary experiments on the causal factors in animal learning. II. *Journal of Comparative Psychology*, 9:245–274, 1929.

[56] Gabrielle Girardeau, Karim Benchenane, Sidney I. Wiener, György Buzsáki, and Michaël B. Zugaro. Selective suppression of hippocampal ripples impairs spatial memory. *Nature neuroscience*, 12(10):1222–1223, 2009.

[57] Jeremy P. Goodridge, Paul A. Dudchenko, Kimberly A. Worboys, Edward J. Golob, and Jeffrey S. Taube. Cue control and head direction cells. *Behavioral neuroscience*, 112(4):749, 1998.

[58] H. J. Groenewegen, E. Vermeulen-Van der Zee, A. te Kortschot, and M. P. Witter. Organization of the projections from the subiculum to the ventral striatum in the rat. A study using anterograde transport of Phaseolus vulgaris leucoagglutinin. *Neuroscience*, 23(1):103–120, October 1987.

[59] Alex Guazzelli, Mihail Bota, Fernando J. Corbacho, and Michael A. Arbib. Affordances. Motivations, and the World Graph Theory. *Adaptive Behavior*, 6(3-4):435–471, January 1998.

[60] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

[61] Niels Hansen and Denise Manahan-Vaughan. Dopamine d1/d5 receptors mediate informational saliency that promotes persistent hippocampal long-term plasticity. *Cerebral Cortex (New York, N.Y.: 1991)*, 24(4):845–858, April 2014.

[62] Leslie H. Hicks. Effects of overtraining on acquisition and reversal of place and response learning. *Psychological Reports*, 15(2):459–462, 1964.

[63] James C. Houk, James L. Adams, and Andrew G. Barto. A model of how the basal ganglia generate and use neural signals that predict reinforcement. In J. C. Houk, J. L. Davis, and D. G. Beiser, editors, *Models of information processing in the basal ganglia*, Computational neuroscience., pages 249–270. The MIT Press, Cambridge, MA, US, 1995.

[64] Wesley H. Huang, Brett R. Fajen, Jonathan R. Fink, and William H. Warren. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4):288–299, April 2006.

[65] Mohammad Abdel Kareem Jaradat, Mohammad H. Garibeh, and Eyad A. Feilat. Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field. *Soft Computing*, 16(1):153–164, June 2011.

[66] Kate J. Jeffery, Jonathan J. Wilson, Giulio Casali, and Robin M. Hayman. Neural encoding of large-scale three-dimensional space—properties and constraints. *Cognitive Science*, page 927, 2015.

[67] Adam Johnson and A. David Redish. Hippocampal replay contributes to within session learning in a temporal difference reinforcement learning model. *Neural Networks*, 18(9):1163–1171, 2005.

[68] Adam Johnson and A. David Redish. Neural Ensembles in CA3 Transiently Encode Paths Forward of the Animal at a Decision Point. *The Journal of Neuroscience*, 27(45):12176–12189, November 2007.

[69] Bethany Jones, Elizabeth Bukoski, Lynn Nadel, and Jean-Marc Fellous. Remaking memories: reconsolidation updates positively motivated spatial memory in rats. *Learning & Memory (Cold Spring Harbor, N.Y.)*, 19(3):91–98, February 2012.

[70] M. W. Jung, S. I. Wiener, and B. L. McNaughton. Comparison of spatial firing characteristics of units in dorsal and ventral hippocampus of the rat. *The Journal of Neuroscience*, 14(12):7347–7356, December 1994.

[71] Ishay Kamon, Elon Rimon, and Ehud Rivlin. Tangentbug: A range-sensor-based navigation algorithm. *The International Journal of Robotics Research*, 17(9):934–953, 1998.

[72] Dov Katz, Yuri Pyuro, and Oliver Brock. *Learning to Manipulate Articulated Objects in Unstructured Environments Using a Grounded Relational Representation.*

[73] Alexander T. Keinath, Melissa E. Wang, Ellen G. Wann, Robin K. Yuan, Joshua T. Dudman, and Isabel A. Muzzio. Precise spatial coding is preserved along the longitudinal hippocampal axis. *Hippocampus*, 24(12):1533–1548, December 2014.

[74] Oussama Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90–98, March 1986.

[75] H. Kimura, T. Yamashita, and S. Kobayashi. Reinforcement learning of walking behavior for a four-legged robot. In *Proceedings of the 40th IEEE Conference on Decision and Control, 2001*, volume 1, pages 411–416 vol.1, 2001.

[76] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, September 2013.

[77] Ansgar Koene and Tony J. Prescott. Hippocampus, Amygdala and Basal Ganglia based navigation control. In *Artificial Neural Networks–ICANN 2009*, pages 267–276. Springer, 2009.

[78] Isador Krechevsky. " Hypotheses" in rats. *Psychological Review*, 39(6):516, 1932.

[79] Jeffrey L. Krichmar and Florian Röhrbein. Value and reward based learning in neurorobots. *Frontiers in Neurorobotics*, 7, September 2013.

[80] William H. Kruskal and W. Allen Wallis. Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621, December 1952.

[81] Martin Krzywinski and Naomi Altman. Points of Significance: Visualizing samples with box plots. *Nature Methods*, 11(2):119–120, February 2014.

[82] Carien S. Lansink and Cyriel M. A. Pennartz. Associative Reactivation of Place–Reward Information in the Hippocampal–Ventral Striatal Circuitry. In Masami Tatsuno, editor, *Analysis and Modeling of Coordinated Multi-neuronal Activity*, number 12 in Springer Series in Computational Neuroscience, pages 81–104. Springer New York, 2015.

[83] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[84] Colin Lever, Stephen Burton, Ali Jeewajee, John O'Keefe, and Neil Burgess. Boundary Vector Cells in the subiculum of the hippocampal formation. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 29(31):9771–9777, August 2009.

[85] John E. Lisman and Anthony A. Grace. The Hippocampal-VTA Loop: Controlling the Entry of Information into Long-Term Memory. *Neuron*, 46(5):703–713, June 2005.

[86] M. Llofriu, G. Tejera, M. Contreras, T. Pelc, J. M. Fellous, and A. Weitzenfeld. Goal-oriented robot navigation learning using a multi-scale space representation. *Neural Networks*, 2015.

[87] Lauren L. Long, Jamie G. Bunce, and James J. Chrobak. Theta variation and spatiotemporal scaling along the septotemporal axis of the hippocampus. *Frontiers in Systems Neuroscience*, 9:37, 2015.

[88] Michael Loureiro, Lucas Lecourtier, Michel Engeln, Joëlle Lopez, Brigitte Cosquer, Karin Geiger, Christian Kelche, Jean-Christophe Cassel, and Anne Pereira De Vasconcelos. The ventral hippocampus is necessary for expressing a spatial memory. *Brain Structure and Function*, 217(1):93–106, 2012.

[89] Vladimir J. Lumelsky and Alexander A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2(1-4):403–430, 1987.

[90] Hans Maaswinkel and Ian Q. Whishaw. Homing with locale, taxon, and dead reckoning strategies by foraging rats: sensory hierarchy in spatial navigation. *Behavioural Brain Research*, 99(2):143–152, March 1999.

[91] Tamas Madl, Ke Chen, Daniela Montaldi, and Robert Trappl. Computational cognitive models of spatial memory in navigation space: A review. *Neural Networks*, 65:18–43, May 2015.

[92] Andrew P. Maurer, Shea R. Vanrhoads, Gary R. Sutherland, Peter Lipa, and Bruce L. McNaughton. Self-motion and the origin of differential spatial scaling along the septo-temporal axis of the hippocampus. *Hippocampus*, 15(7):841–852, 2005.

[93] Dr B. L. McNaughton, C. A. Barnes, and J. O'Keefe. The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats. *Experimental Brain Research*, 52(1):41–49, September 1983.

[94] Michael Milford and Gordon Wyeth. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *The International Journal of Robotics Research*, July 2009.

[95] Michael Milford, Gordon Wyeth, and David Prasser. RatSLAM on the edge: revealing a coherent representation from an overloaded rat brain. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4060–4065. IEEE, 2006.

[96] M.-L. Mittelstaedt and H. Mittelstaedt. Homing by path integration in a mammal. *Naturwissenschaften*, 67(11):566–567, November 1980.

[97] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, 2003.

[98] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, and others. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598, 2002.

[99] Eduardo F. Morales. Scaling up reinforcement learning with a relational representation. In *In Proc. of the Workshop on Adaptability in Multi-agent Systems*, pages 15–26, 2003.

[100] R. G. M. Morris, Paul Garrud, J. N. P. Rawlins, and John O'Keefe. Place navigation impaired in rats with hippocampal lesions. *Nature*, 297(5868):681–683, 1982.

[101] Richard G.M. Morris. Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12(2):239–260, May 1981.

[102] Edvard I. Moser, Emilio Kropff, and May-Britt Moser. Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, 31(1):69–89, 2008.

[103] M. B. Moser, E. I. Moser, E. Forrest, P. Andersen, and R. G. Morris. Spatial learning with a minislab in the dorsal hippocampus. *Proceedings of the National Academy of Sciences*, 92(21):9697–9701, October 1995.

[104] May-Britt Moser and Edvard I. Moser. Functional differentiation in the hippocampus. *Hippocampus*, 8(6):608–619, January 1998.

[105] O. Motlagh, S. H. Tang, N. Ismail, and A. R. Ramli. An expert fuzzy cognitive map for reactive navigation of mobile robots. *Fuzzy Sets and Systems*, 201:105–121, August 2012.

[106] Omid Reza Esmaeili Motlagh, Tang Sai Hong, and Napsiah Ismail. Development of a new minimum avoidance system for a behavior-based mobile robot. *Fuzzy Sets and Systems*, 160(13):1929–1946, July 2009.

[107] R. U. Muller, E. Bostock, J. S. Taube, and J. L. Kubie. On the directional firing properties of hippocampal place cells. *The Journal of Neuroscience*, 14(12):7235–7251, December 1994.

[108] R. U. Muller and J. L. Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, 7(7):1951–1968, July 1987.

[109] Norman L. Munn. *Handbook of psychological research on the rat; an introduction to animal psychology*, volume xxvi. Houghton Mifflin, Oxford, England, 1950.

[110] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, October 2015.

[111] B. Nemec, M. Zorko, and L. Zlajpah. Learning of a ball-in-a-cup playing robot. In *2010 IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, pages 297–301, June 2010.

[112] James Ng and Thomas Bräunl. Performance comparison of bug navigation algorithms. *Journal of Intelligent and Robotic Systems*, 50(1):73–84, 2007.

[113] J. O'keefe. Spatial memory within and without the hippocampal system. *ResearchGate*, January 1984.

[114] J. O'Keefe and N. Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428, May 1996.

[115] J O'Keefe and J Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1), November 1971.

[116] John O'Keefe. Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1):78–109, 1976.

[117] John O'Keefe and Lynn Nadel. *The Hippocampus as a Cognitive Map.* Oxford University Press, Oxford : New York, December 1978.

[118] David S. Olton. Mazes, maps, and memory. *American Psychologist*, 34(7):583–596, July 1979.

[119] David S. Olton and Robert J. Samuelson. Remembrance of places passed: spatial memory in rats. *Journal of Experimental Psychology: Animal Behavior Processes*, 2(2):97, 1976.

[120] Mark G. Packard. Exhumed from thought: Basal ganglia and response learning in the plus-maze. *Behavioural Brain Research*, 199(1):24–31, April 2009.

[121] John M. Pearce, Amanda DL Roberts, and Mark Good. Hippocampal lesions disrupt navigation based on cognitive maps but not heading vectors. *Nature*, 396(6706):75–77, 1998.

[122] Giovanni Pezzulo, Matthijs A. A. van der Meer, Carien S. Lansink, and Cyriel M. A. Pennartz. Internally generated sequences in learning and executing goal-directed behavior. *Trends in Cognitive Sciences*, 18(12):647–657, December 2014.

[123] Brad E. Pfeiffer and David J. Foster. Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*, 497(7447):74–79, May 2013.

[124] Justus Piater, Sébastien Jodogne, Renaud Detry, Dirk Kraft, Norbert Krüger, Oliver Kroemer, and Jan Peters. Learning visual representations for perception-action systems. *The International Journal of Robotics Research*, 30(3):294–307, March 2011.

[125] Helen H. J. Pothuizen, Wei-Ning Zhang, Ana L. Jongen-Rêlo, Joram Feldon, and Benjamin K. Yee. Dissociation of function between the dorsal and the ventral hippocampus in spatial learning abilities of the rat: a within-subject, within-task comparison of reference and working spatial memory. *European Journal of Neuroscience*, 19(3):705–712, February 2004.

[126] B. Poucet and M. C. Buhot. Effects of medial septal or unilateral hippocampal inactivations on reference and working spatial memory in rats. *Hippocampus*, 4(3):315–321, 1994.

[127] Bruno Poucet, Thom Herrmann, and Marie-Christine Buhot. Effects of short-lasting inactivations of the ventral hippocampus and medial septum on long-term and short-term acquisition of spatial information in rats. *Behavioural brain research*, 44(1):53–65, 1991.

[128] Dale Purves. *Neuroscience*. Sinauer Associates, 2012.

[129] A. David Redish. *Beyond the Cognitive Map: From Place Cells to Episodic Memory.* MIT Press, 1999.

[130] A. David Redish and David S. Touretzky. Cognitive maps beyond the hippocampus. *Hippocampus*, pages 15–35, 1997.

[131] Frank Restle. Discrimination of cues in mazes: A resolution of the 'place-vs.-response' question. *Psychological Review*, 64(4):217–228, July 1957.

[132] Aude Retailleau, Stephanie Etienne, Martin Guthrie, and Thomas Boraud. Where is my reward and how do I get it? Interaction between the hippocampus and the basal ganglia during spatial learning. *Journal of Physiology-Paris*, 106(3–4):72–80, May 2012.

[133] Martin Riedmiller, Thomas Gabel, Roland Hafner, and Sascha Lange. Reinforcement learning for robot soccer. *Autonomous Robots*, 27(1):55–73, July 2009.

[134] B. F. Ritchie, B. Aeschliman, and P. Pierce. Studies in spatial learning. VIII. Place performance and the acquisition of place dispositions. *Journal of Comparative and Physiological Psychology*, 43(2):73–85, 1950.

[135] Bruno Rivard, Yu Li, Pierre-Pascal Lenck-Santini, Bruno Poucet, and Robert U. Muller. Representation of Objects in Space by Two Classes of Hippocampal Pyramidal Cells. *The Journal of General Physiology*, 124(1):9–25, July 2004.

[136] Demetris K Roumis and Loren M Frank. Hippocampal sharp-wave ripples in waking and sleeping states. *Current Opinion in Neurobiology*, 35:6–12, December 2015.

[137] Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L. McNaughton, Menno P. Witter, May-Britt Moser, and Edvard I. Moser. Conjunctive Representation of Position, Direction, and Velocity in Entorhinal Cortex. *Science*, 312(5774):758–762, May 2006.

[138] Tommi Jaakkola Satinder P. Singh. Reinforcement learning with soft state aggregation. 1999.

[139] Nathan W. Schultheiss, James R. Hinman, and Michael E. Hasselmo. Models and Theoretical Frameworks for Hippocampal and Entorhinal Cortex Function in Memory and Navigation. In Masami Tatsuno, editor, *Analysis and Modeling of Coordinated Multi-neuronal Activity*, number 12 in Springer Series in Computational Neuroscience, pages 247–268. Springer New York, 2015.

[140] Patricia E. Sharp, editor. *The Neural Basis of Navigation.* Springer US, Boston, MA, 2002.

[141] Chaoxia Shi, Yanqing Wang, and Jingyu Yang. A local obstacle avoidance method for mobile robots in partially known environment. *Robotics and Autonomous Systems*, 58(5):425–434, May 2010.

[142] Satinder P. Singh and Jordan MI. Jaakkola. Reinforcement Learning with Soft State Aggregation. 1999.

[143] Satinder P. Singh, Tommi Jaakkola, and Michael I. Jordan. Reinforcement Learning with Soft State Aggregation. In *Advances in Neural Information Processing Systems 7*, pages 361–368. MIT Press, 1995.

[144] William E. Skaggs and Bruce L. McNaughton. Theta Phase Precession in Hippocampal. *Hippocampus*, 6:149–172, 1996.

[145] B. F. Skinner. *The behavior of organisms: an experimental analysis.* Appleton-Century, Oxford, England, 1938.

[146] Willard S. Small. An Experimental Study of the Mental Processes of the Rat. *The American Journal of Psychology*, 11(2):133–165, 1900.

[147] Willard S. Small. Experimental Study of the Mental Processes of the Rat. II. *The American Journal of Psychology*, 12(2):206–239, 1901.

[148] KENNETH W. Spence and RONALD Lippitt. An experimental test of the sign-gestalt theory of trial and error learning. *Journal of Experimental Psychology*, 36(6):491, 1946.

[149] John Staddon and Yael Niv. Operant conditioning. *Scholarpedia*, 3(9):2318, 2008.

[150] R.j. Steele and R.g.m. Morris. Delay-dependent impairment of a matching-to-place task with chronic and intrahippocampal infusion of the NMDA-antagonist D-AP5. *Hippocampus*, 9(2):118–136, January 1999.

[151] Stephanie B. Stolz and Dale F. Lott. Establishment in rats of a persistent response producing a net loss of reinforcement. *Journal of Comparative and Physiological Psychology*, 57(1):147–149, 1964.

[152] Jeffrey J. Stott and A. David Redish. A functional difference in information processing between orbitofrontal cortex and ventral striatum during decision-making behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 369(1655):20130472, 2014.

[153] Jeffrey J. Stott and A. David Redish. Representations of value in the brain: an embarrassment of riches? *PLoS Biol*, 13(6):e1002174, 2015.

[154] Bryan A. Strange, Menno P. Witter, Ed S. Lein, and Edvard I. Moser. Functional organization of the hippocampal longitudinal axis. *Nature Reviews. Neuroscience*, 15(10):655–669, October 2014.

[155] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 1998.

[156] Y. Takahashi, M. Asada, and K. Hosoda. Reasonable performance in less learning time by real robot based on incremental state space segmentation. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, volume 3, pages 1518–1524 vol.3, November 1996.

[157] J. S. Taube, R. U. Muller, and J. B. Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *The Journal of Neuroscience*, 10(2):420–435, February 1990.

[158] Jeffrey S. Taube. The Head Direction Signal: Origins and Sensory-Motor Integration. *Annual Review of Neuroscience*, 30(1):181–207, 2007.

[159] Jeffrey S. Taube and Heather L. Burton. Head direction cell activity monitored in a novel environment and during a cue conflict situation. *Journal of Neurophysiology*, 74(5):1953–1971, 1995.

[160] Edward J. Tehovnik and Marc A. Sommer. Effective spread and timecourse of neural inactivation caused by lidocaine injection in monkey cerebral cortex. *Journal of neuroscience methods*, 74(1):17–26, 1997.

[161] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005. Published: Hardcover.

[162] Michel Tokic, Wolfgang Ertel, and Joachim Fessler. The crawler, a class room demonstrator for reinforcement learning. In *Twenty-Second International FLAIRS Conference*, March 2009.

[163] E. C. Tolman, B. F. Ritchie, and D. Kalish. Studies in spatial learning. I. Orientation and the short-cut. 1946. *Journal of Experimental Psychology. General*, 121(4):429–434, December 1992.

[164] Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.

[165] EDWARD CHACE Tolman. Sign-Gestalt or conditioned reflex. *Psychological Review*, 40(3):246, 1933.

[166] David S. Touretzky. The Rodent Navigation Circuit. In Patricia E. Sharp, editor, *The Neural Basis of Navigation*, pages 217–233. Springer US, 2002.

[167] David S. Touretzky and A. David Redish. Theory of rodent navigation based on interacting representations of space. *Hippocampus*, 6(3):247–270, 1996.

[168] Olivier Trullier, Sidney I. Wiener, Alain Berthoz, and Jean-Arcady Meyer. Biologically based artificial navigation systems: Review and prospects. *Progress in neurobiology*, 51(5):483–544, 1997.

[169] Jose L. Valdes, Bruce L. McNaughton, and Jean-Marc Fellous. Off-Line Reactivation of Experience-Dependent Neuronal Firing Patterns in the Rat Ventral Tegmental Area. *Journal of Neurophysiology*, page jn.00758.2014, June 2015.

[170] Rong-Jong Wai, Chia-Ming Liu, and You-Wei Lin. Design of switching path-planning control for obstacle avoidance of mobile robot. *Journal of the Franklin Institute*, 348(4):718–737, May 2011.

[171] Alfredo Weitzenfeld, Michael A. Arbib, and Amanda Alexander. *The Neural Simulation Language: A System for Brain Modeling*. MIT Press, Cambridge, MA, USA, 2002.

[172] Alfredo Weitzenfeld, S. Gutierrez-Nolasco, and N. Venkatasubramanian. MIRO: An embedded distributed architecture for biologically inspired mobile robots. In *ICAR-03*, Coimbra, Portugal, 2003.

[173] I. Q. Whishaw, J. C. Cassel, and L. E. Jarrad. Rats with fimbria-fornix lesions display a place response in a swimming pool: a dissociation between getting there and knowing where. *The Journal of Neuroscience*, 15(8):5779–5788, August 1995.

[174] M. A. Wilson and B. L. McNaughton. Dynamics of the hippocampal ensemble code for space. *Science (New York, N.Y.)*, 261(5124):1055–1058, August 1993.

[175] James C. Woodson, Deric Macintosh, Monika Fleshner, and David M. Diamond. Emotion-induced amnesia in rats: working memory-specific impairment, corticosterone-memory correlation, and fear versus arousal effects on memory. *Learning & Memory (Cold Spring Harbor, N.Y.)*, 10(5):326–336, October 2003.

[176] Toshiyuki Yasuda and Kazuhiro Ohkura. A reinforcement learning technique with an adaptive action generator for a multi-robot system. In Minoru Asada, John C. T. Hallam, Jean-Arcady Meyer, and Jun Tani, editors, *From Animals to Animats 10*, number 5040 in Lecture Notes in Computer Science, pages 250–259. Springer Berlin Heidelberg, January 2008.

[177] Pei-Yan Zhang, Tian-Sheng Lü, and Li-Bo Song. Soccer robot path planning based on the artificial potential field approach with simulated annealing. *Robotica*, 22(05):563–566, October 2004.

[178] Y. Zhu, T. Zhang, and J. Song. An improved wall following method for escaping from local minimum in artificial potential field based path planning. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 6017–6022, December 2009.

[179] Yi Zhu, Tao Zhang, Jingyan Song, and Xiaqin Li. A new bug-type navigation algorithm for mobile robots in unknown environments containing moving obstacles. *Industrial Robot: An International Journal*, 39(1):27–39, January 2012.

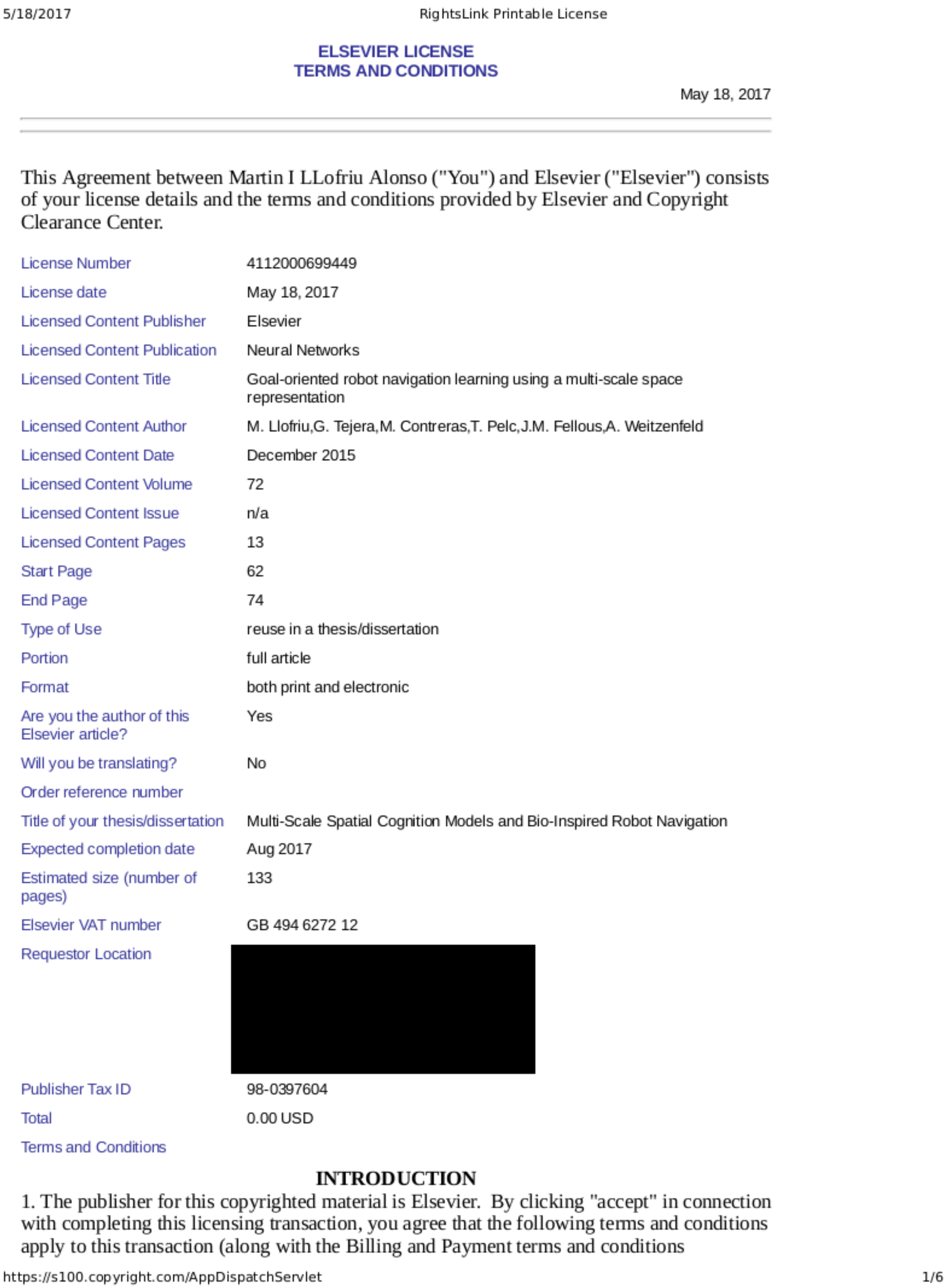
[180] Yi Zhu, Tao Zhang, Jingyan Song, and Xiaqin Li. A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge. *Knowledge-Based Systems*, 27:302–313, March 2012.

[181] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso. SSL-vision: The shared vision system for the RoboCup small size league. *RoboCup 2009: Robot Soccer World Cup XIII*, pages 425–436, 2009.

APPENDICES

Appendix A: Permission for Reuse

The permissions for reuse for the content on chapters 4 and 5 is included below.



ELSEVIER LICENSE
TERMS AND CONDITIONS

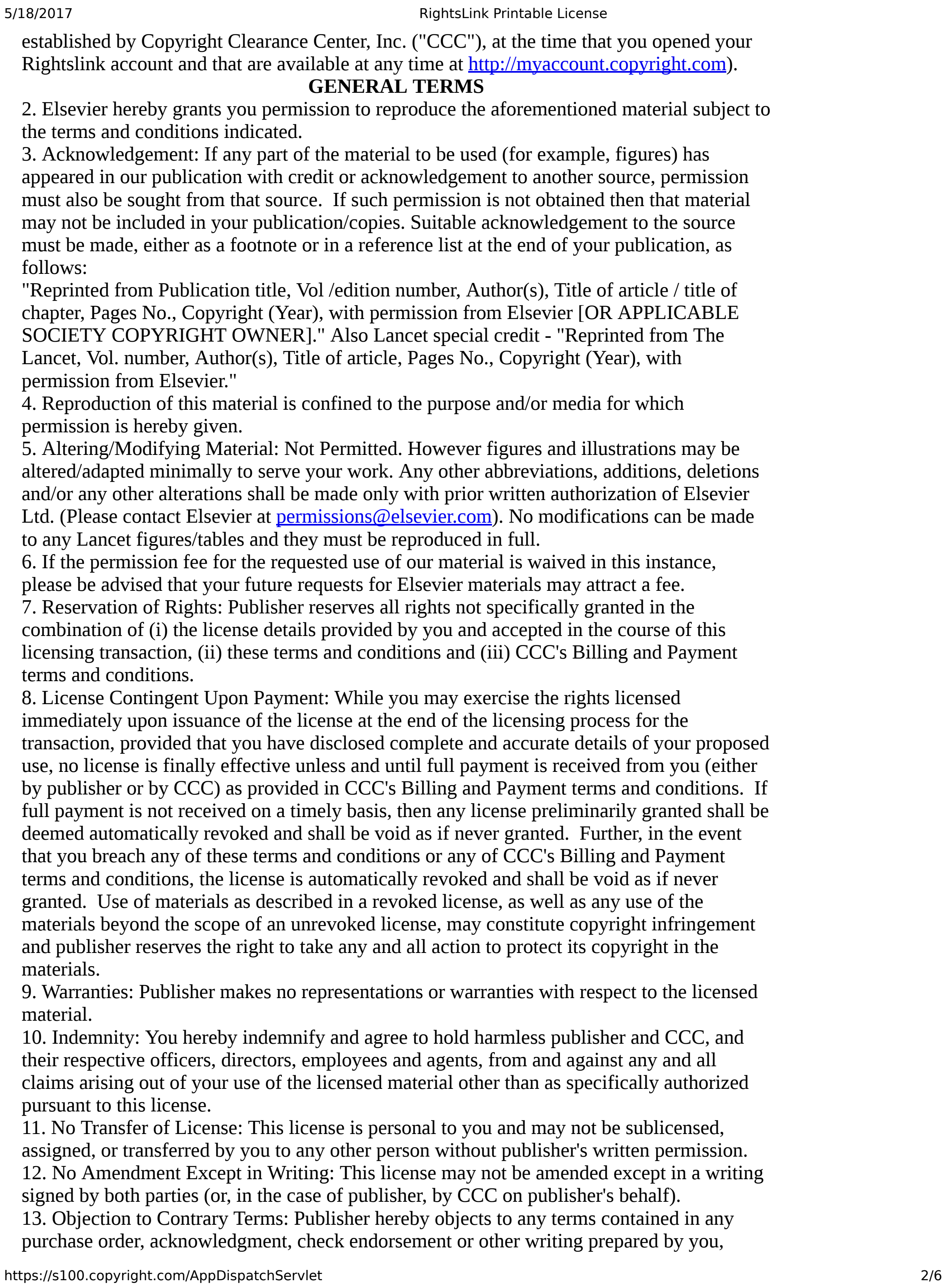
May 18, 2017

This Agreement between Martin I LLofriu Alonso ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	4112000699449
License date	May 18, 2017
Licensed Content Publisher	Elsevier
Licensed Content Publication	Neural Networks
Licensed Content Title	Goal-oriented robot navigation learning using a multi-scale space representation
Licensed Content Author	M. Llofriu,G. Tejera,M. Contreras,T. Pelc,J.M. Fellous,A. Weitzenfeld
Licensed Content Date	December 2015
Licensed Content Volume	72
Licensed Content Issue	n/a
Licensed Content Pages	13
Start Page	62
End Page	74
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	both print and electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	Multi-Scale Spatial Cognition Models and Bio-Inspired Robot Navigation
Expected completion date	Aug 2017
Estimated size (number of pages)	133
Elsevier VAT number	GB 494 6272 12
Requestor Location	[redacted]
Publisher Tax ID	98-0397604
Total	0.00 USD
Terms and Conditions	

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions



established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

GENERAL TERMS

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier at permissions@elsevier.com). No modifications can be made to any Lancet figures/tables and they must be reproduced in full.

6. If the permission fee for the requested use of our material is waived in this instance, please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

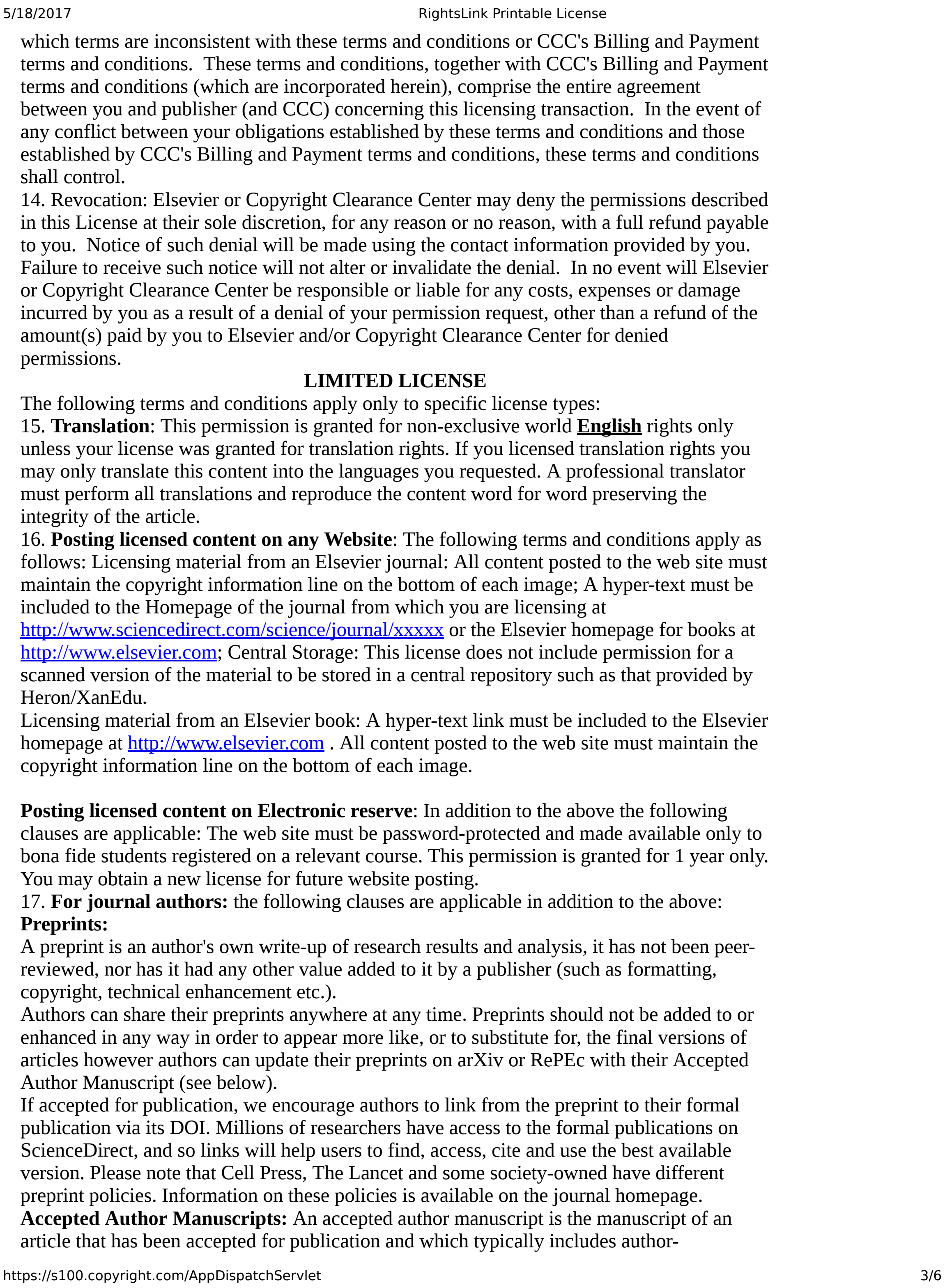
9. Warranties: Publisher makes no representations or warranties with respect to the licensed material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing: This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you,



which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation**: This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article.

16. **Posting licensed content on any Website**: The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at http://www.sciencedirect.com/science/journal/xxxxx or the Elsevier homepage for books at http://www.elsevier.com; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at http://www.elsevier.com . All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors:** the following clauses are applicable in addition to the above:

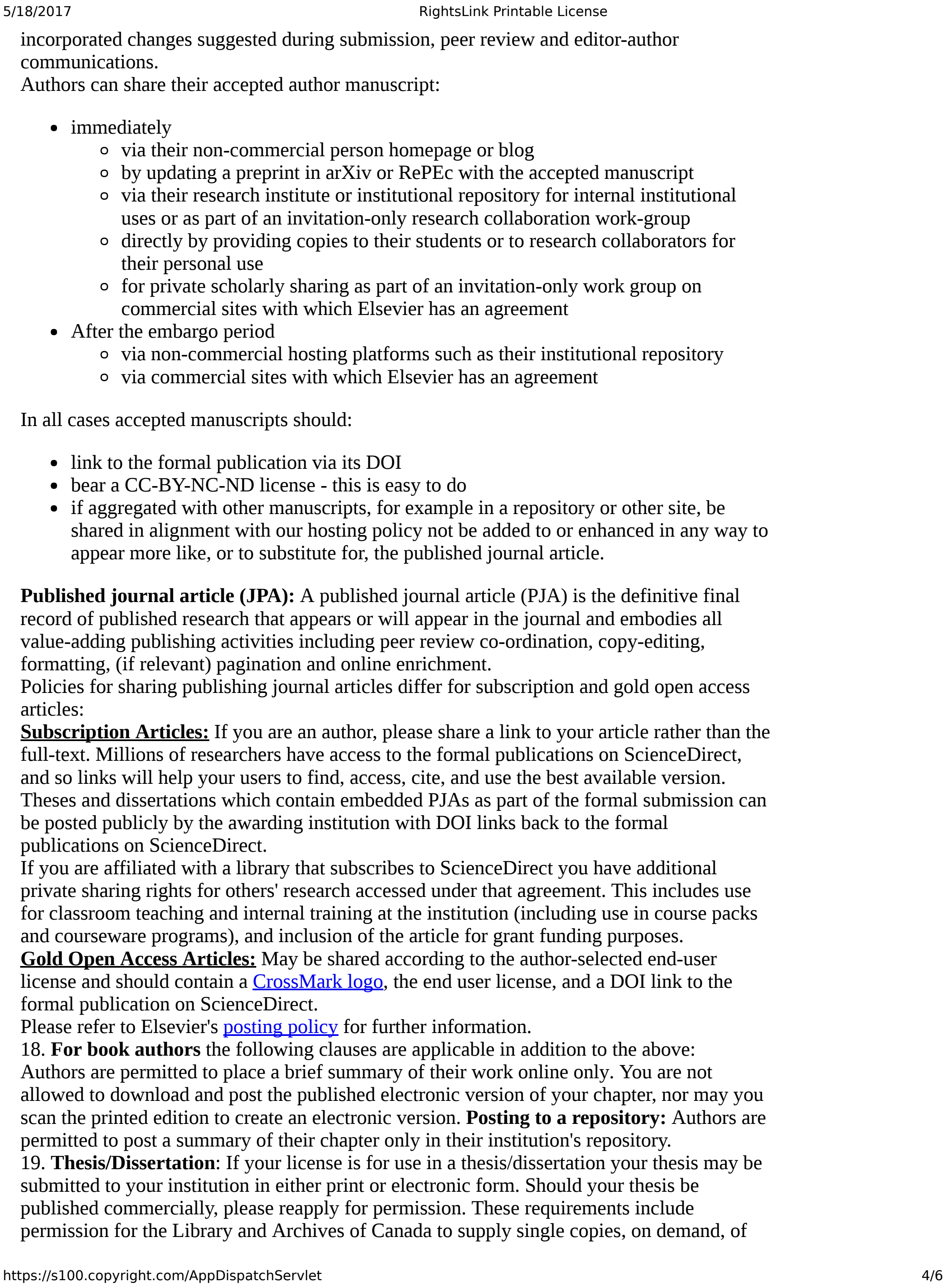
Preprints:

A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).

Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-



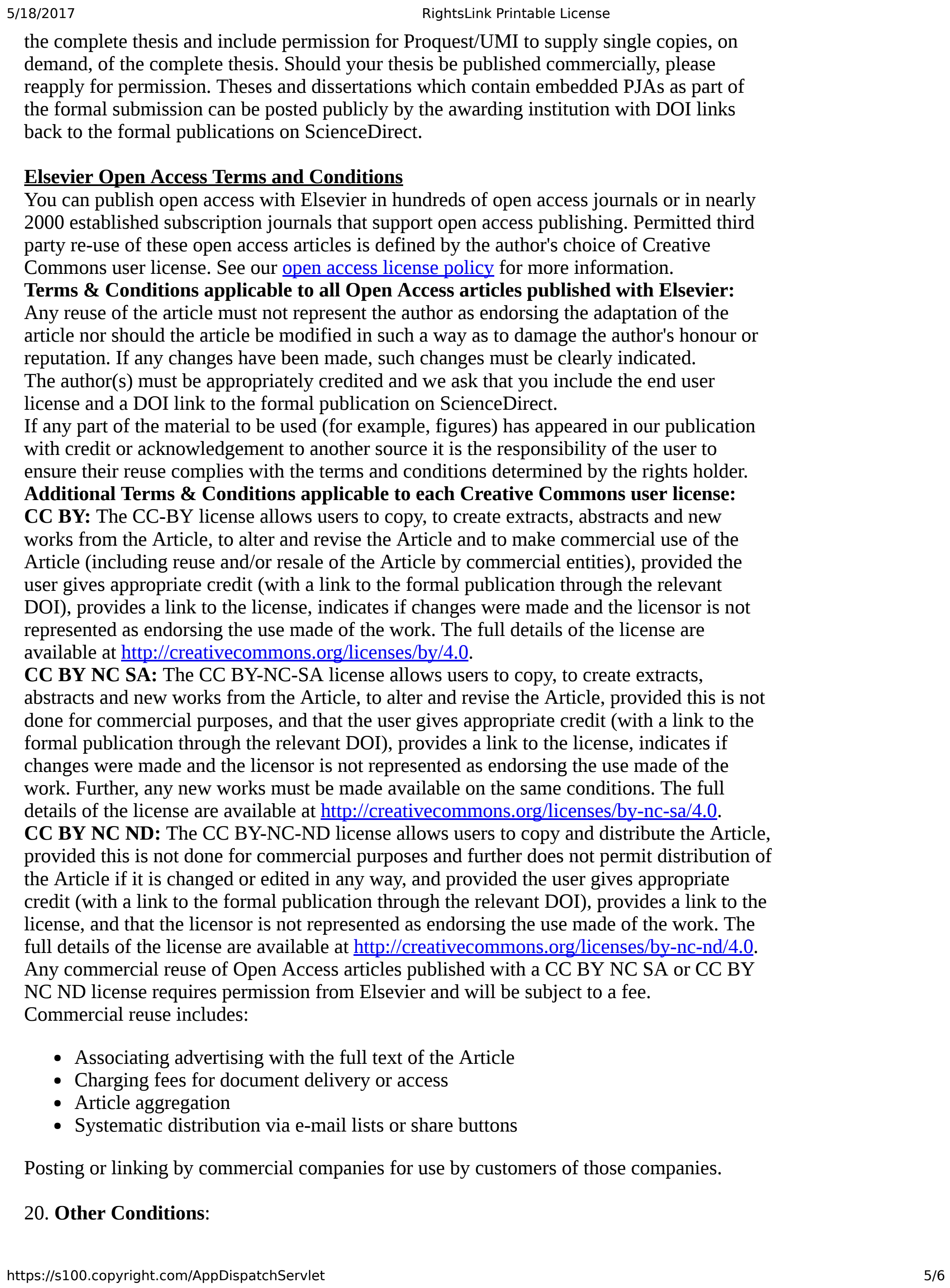
incorporated changes suggested during submission, peer review and editor-author communications.
Authors can share their accepted author manuscript:

- immediately
  - via their non-commercial person homepage or blog
  - by updating a preprint in arXiv or RePEc with the accepted manuscript
  - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group
  - directly by providing copies to their students or to research collaborators for their personal use
  - for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- After the embargo period
  - via non-commercial hosting platforms such as their institutional repository
  - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.
Policies for sharing publishing journal articles differ for subscription and gold open access articles:
Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.
Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.
If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.
Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a CrossMark logo, the end user license, and a DOI link to the formal publication on ScienceDirect.
Please refer to Elsevier's posting policy for further information.
18. **For book authors** the following clauses are applicable in addition to the above:
Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.
19. **Thesis/Dissertation**: If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of



5/18/2017 RightsLink Printable License

the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author's choice of Creative Commons user license. See our open access license policy for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated.

The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect.

If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

CC BY: The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at http://creativecommons.org/licenses/by/4.0.

CC BY NC SA: The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. Further, any new works must be made available on the same conditions. The full details of the license are available at http://creativecommons.org/licenses/by-nc-sa/4.0.

CC BY NC ND: The CC BY-NC-ND license allows users to copy and distribute the Article, provided this is not done for commercial purposes and further does not permit distribution of the Article if it is changed or edited in any way, and provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, and that the licensor is not represented as endorsing the use made of the work. The full details of the license are available at http://creativecommons.org/licenses/by-nc-nd/4.0.

Any commercial reuse of Open Access articles published with a CC BY NC SA or CC BY NC ND license requires permission from Elsevier and will be subject to a fee.

Commercial reuse includes:

- Associating advertising with the full text of the Article
- Charging fees for document delivery or access
- Article aggregation
- Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.

20. **Other Conditions**:

https://s100.copyright.com/AppDispatchServlet 5/6

v1.9

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.