UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

A Computational Model of Simulated Motion in Rats

Permalink

https://escholarship.org/uc/item/4dx4v8jn

Author

Doubleday, Steven Jay

Publication Date

2020

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, availalbe at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

A Computational Model of Simulated Motion in Rats

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Social Science with a concentration in Mathematical Behavioral Science

by

Steven Jay Doubleday

Dissertation Committee:
Professor Michael Lee, Chair
Professor Jeffrey Krichmar
Professor Ramesh Srinivasan

2020

© 2020 Steven Jay Doubleday

DEDICATION

To Liz, who has made it all possible

and in memory of Chuck Ives – “there’s so much to be done”

TABLE OF CONTENTS

Page

LIST OF FIGURES **vi**

LIST OF TABLES **viii**

ACKNOWLEDGMENTS **ix**

VITA **xi**

ABSTRACT OF THE DISSERTATION **xii**

1 Introduction **1**
- 1.1 Overview . . . 1
- 1.2 Spatial navigation . . . 4
- 1.3 Adaptive value of simulated motion . . . 12
- 1.4 Contribution . . . 13
- 1.5 Roadmap . . . 14

2 Spatial navigation **16**
- 2.1 Overview . . . 16
- 2.2 Place cells and navigation . . . 17
- 2.3 Electrical activity and memory mechanisms in the hippocampus . . . 18
- 2.4 Models of memory and learning . . . 20
  - 2.4.1 Complementary Learning Systems . . . 20
  - 2.4.2 Attractor networks . . . 21
  - 2.4.3 Autoassociative networks . . . 22
- 2.5 Anatomy of the hippocampal formation and parahippocampal region . . . 27
- 2.6 Hippocampus . . . 27
  - 2.6.1 CA3 as an autoassociative network . . . 29
  - 2.6.2 Hippocampus as a hash index to memories . . . 32
- 2.7 Path integration and spatial navigation . . . 33
  - 2.7.1 Head direction cells and modeling the animal's orientation . . . 34
  - 2.7.2 Spatial attractor model . . . 36
  - 2.7.3 Grid cells and Medial Entorhinal Cortex . . . 38
  - 2.7.4 Multi-dimensional information in Lateral Entorhinal Cortex . . . 42
  - 2.7.5 Remapping . . . 44

2.7.6 Location of the spatial attractor dynamics . . . 46
2.7.7 CA1 information processing . . . 47
2.8 Trajectories . . . 49
2.9 Two hypotheses regarding preplay trajectories . . . 52
2.10 Communication between regions . . . 61

3 Petri nets and X-nets **71**
3.1 Overview . . . 71
3.2 Petri nets . . . 72
3.3 X-nets for SMS . . . 76
3.4 Petri net extensions to support composition and external systems . . . 78
3.5 Hierarchical net extensions . . . 79
3.5.1 Include hierarchy . . . 79
3.5.2 Executable Petri net . . . 80
3.5.3 Merge places . . . 81
3.5.4 External system integration through the Runner API . . . 81
3.5.5 External interfaces . . . 83
3.5.6 External places . . . 84
3.5.7 External transitions . . . 84
3.5.8 Petri net persistence . . . 85
3.6 Synchronization of Petri net processing with the external system . . . 86
3.7 Petri net summary . . . 88

4 Computational Model **89**
4.1 Overview . . . 89
4.2 Simulated Motion System . . . 92
4.2.1 Components . . . 94
4.3 Model Assumptions . . . 106
4.4 Model Processing . . . 110
4.5 Managing the proliferation of Place IDs . . . 114

5 Methods **121**
5.1 Overview . . . 121
5.2 Model parameters . . . 123
5.2.1 Ripples . . . 124
5.2.2 Grids . . . 125
5.2.3 Head direction cells . . . 126
5.2.4 Sparse Place IDs . . . 128
5.3 Model execution . . . 129
5.3.1 Time steps . . . 129
5.3.2 Initial conditions . . . 130
5.4 Log files . . . 130

6 Results **133**
6.1 Overview . . . 133
6.2 Model navigation behavior . . . 138
6.3 Analysis of retraced trajectories . . . 140
6.3.1 Results for retraced trajectories . . . 143
6.4 Analysis of exploration efficiency . . . 150
6.4.1 Results for exploration efficiency . . . 152
6.5 Analysis of PlaceSystem saturation . . . 157
6.5.1 Results for PlaceSystem saturation . . . 158
6.6 Results summary . . . 161

7 Conclusion **162**
7.1 Summary . . . 162
7.2 Future work . . . 164
7.2.1 Rewards . . . 164
7.2.2 CA3 models . . . 165

Bibliography **167**

Appendix A Engineering SMS **174**

Appendix B Additional findings **185**

Appendix C Testable predictions **206**

Appendix D Model Behaviors **209**

Glossary **217**

LIST OF FIGURES

Page

1.1 Simulated motion as implemented in SMS 10

2.1 Associative memory 24
2.2 Hebb-Marr networks 26
2.3 Anatomy of hippocampal formation and parahippocampal region 28
2.4 Ring attractor model for head orientation 37
2.5 Grid cell properties 40
2.6 Trisynaptic and monosynaptic paths 43
2.7 Environment design in Dragoi and Tonegawa (2011) 58
2.8 Superficial MEC and CA1 reconstruction 66
2.9 Alternating MEC Layer 3 and CA1 activity 68

3.1 Inbound and outbound arcs 72
3.2 Disabled transition 73
3.3 Inhibitory arc 73
3.4 Enabled transition fires 74
3.5 Arc weight expression 74
3.6 Random firing of enabled transitions 75
3.7 Structure of actions 77
3.8 Include hierarchy 80
3.9 Merge places 82
3.10 External places 84
3.11 External system uses Runner to set up processing 85
3.12 External system is notified upon place marking 86
3.13 External transition 87

4.1 System data flow 92
4.2 Data flow within the hippocampal formation 93
4.3 Run Petri net 96
4.4 Turn Petri net 97
4.5 Move Petri net 98
4.6 Navigate Petri net 99

6.1 SMS model behavior 136
6.2 Trajectory events 137

6.3 Grid squares explored 152
6.4 Saturation percentage 157

LIST OF TABLES

Page

5.1 SMS model parameter space, defining 36 scenario sets. 124

6.1 Regression of Successful Retrace Percentage on Model Parameters 147
6.2 Regression of Energy on Model Parameters and Time Steps 155
6.3 Regression of Saturation Percentage on Model Parameters and Time Steps . 160

ACKNOWLEDGMENTS

I've been kicking around longer than the average grad student, so I owe a debt of gratitude to a lot of people. My committee has been uniformly supportive for years, and through a couple of twists and turns, including one where I had come to a complete standstill. Michael Lee has been cheerful, pragmatic, and helpful all along in his "pastoral" role, for which I am very grateful. Jeff Krichmar has been in the neuroscience and computational weeds with me as I tried to piece this together, and I've enjoyed and learned a lot from him and his lab students. Ramesh Srinivasan introduced me to neuroscience, and put up with many naive questions as I tried to come up to speed.

I got a crash introduction to the daunting complexities of the hippocampus and entorhinal cortex from Bruce McNaughton, who wrote the book, or at least the articles, on these regions. He also explained various computational models of their functioning, without which this thesis would never have happened. Bruce also patiently endured my questions; I couldn't have been more fortunate.

Sarah Tattersall took a legacy PIPE codebase that had done what most codebases do when untended – sprawl in various directions, with myriad small and large problems – and refactored it relentlessly down to something that *works*. Without that extremely solid base, I would have had to do the same thing, likely not as well, in order to start on the extensions I contribute here.

Mike Hill has been a good friend and teacher for years. Whatever I know about the practicalities of writing decent software I learned from him. GeePawHill is resolutely human in his approach to software, and his decency shows through in every interaction. To the extent that I have something to suggest to computational neuroscientists about how to approach software, it's what I learned from Mike.

IMBS is a small and interesting collection of folks. Students join because their interests don't fit easily in any of the usual slots, and each seems to proceed orthogonally to all the others. Social Dynamics is where people come together to learn from each other, and in the years I attended, was presided over by Don Saari, Louis Narens, and Brian Skyrms. Early on, Don encouraged me by saying it was possible to succeed in IMBS by being good either in math or in computers. That was a relief, as I certainly wasn't good in math. Louis has guided a generation or more of students through the program with the least perceptible of nudges. In an era when most people think a big, heavy-handed system is the only way forward, Louis has kept the program small and made it a haven for folks to chart the most idiosyncratic of paths. Brian offers wine and cheese and evolutionary game theory in equal parts, with great intelligence, patience, and good humor. I originally joined IMBS part-time with a warm welcome from Doug White, who is very good at social networking, both professionally and personally. I learned a great deal on a range of topics from Michael McBride, Jean-Paul Carvalho, Igor Kopylov, Robin Keller, Stergios Skaperdas, and Natalia Chernyshoff. Kim Romney was generous with his time, and very insightful and thoughtful in discussing the core questions that brought me to IMBS.

Jerry Feldman has been the guiding intelligence of my graduate years. In his "Molecule to Metaphor" book I found an introduction to a framework to help me make sense of questions I had been wrestling with for decades. He was very welcoming in person, and the summer I spent in Berkeley reading and coming up to speed was a highlight. Jerry patiently addressed a long list of topics, prefacing most responses with, "that's a different thesis, but here's the party line..." This isn't the thesis he had envisioned for me, but it would never have happened at all without his steering.

My first organized thinking started under the guidance of three professors at Haverford College, Bill Hohenstein, Wyatt MacGaffey, and Mark Gould. The abiding memory of those years is sitting in Mark's living room, being served hot tea and difficult questions. For me, the path from there to IMBS might have taken a long time, but it was very direct.

The mentor of my working years was Chuck Ives. An extraordinary person, Chuck's accomplishments would seem hard to believe, were it not for the detail of his stories and his unfailing honesty in all matters. Over many many years of conversations, he shaped much of the way I think about the world. Without Chuck's influence I would not have had the courage to enter grad school at the unlikely age of 55, nor the hubris to believe I could still make a useful contribution upon finishing.

For all of us, the story begins with family. My father was a Quaker, who looked for the worth in each person, and whose thoughtfulness I've tried to emulate. My mother has been a practical person for a long life, realizing that not much gets done if we're not healthy. She has passed on good habits, and an impatience to get going. In his writing, my brother Jock introduced me to the importance of metaphor, long before I let go of my rationalistic prejudices and accepted the current science. My children, Scott and Kate, have been the source of love, laughter, and purpose from the beginning, continuing on with Adria and my granddaughters, Madeline and Lilyanna. My wife, Liz, has been my loving other half for more than half a lifetime. She paid the mortgage while I dived into school, and encouraged me for years. As the years dragged on, she held on to some shreds of patience – now we finally get to celebrate!

Permission to use copyrighted figures has been granted by Springer Nature Customer Service Centre GmbH, Elsevier, The American Association for the Advancement of Science, and Srini Narayanan.

Funding support was provided by University of California, Irvine, Institute of Mathematical Behavioral Sciences for summer work in 2013 and 2015, and by the International Computer Science Institute in 2014, as part of the iARPA Metaphor project.

VITA

Steven Jay Doubleday

EDUCATION

Doctor of Philosophy in Mathematical Behavioral Science **2020**
University of California, Irvine *Irvine, California*

Bachelor of Arts in Sociology / Anthropology **1979**
Haverford College *Haverford, Pennsylvania*

PUBLICATIONS

Simulating the Market for Protection **2013**
CSSSA 2013

Processing Natural Language About Ongoing Actions in ECG **2017**
AAAI Spring Symposium Series

PROFESSIONAL EXPERIENCE

Building and running computer systems **1982–present**
Investment banking and Healthcare *New York and California*

ABSTRACT OF THE DISSERTATION

A Computational Model of Simulated Motion in Rats

By

Steven Jay Doubleday

Doctor of Philosophy in Social Science with a concentration in Mathematical Behavioral Science

University of California, Irvine, 2020

Professor Michael Lee, Chair

During quiet waking periods a rat generates sequences of activation in the hippocampus that predict its future motion in two-dimensional space. Termed trajectories, the mechanism by which these sequences are generated is not yet known, despite very active research. The theory suggested here is that they are part of a process of the animal simulating its future motion. Simulations are hypothetical events or scenarios, and there is substantial evidence of simulation in humans. In language understanding, for example, evidence suggests we simulate the action implied by a statement, as part of understanding it. Recent evidence suggests that for motion in two-dimensional space, rats may also be simulating their future actions.

In rats, hippocampal trajectories may have different functions in different behavioral contexts, but the prevailing thinking has long been that all the trajectories originate in the hippocampus. I will suggest that a subset of trajectories may originate in areas outside the hippocampus, as part of a process of simulated motion. Simulated motion is the execution of a motor routine without activation of the motor neurons that would cause actual physical motion. I suggest that the simulated motion updates a path integration system, which has

been hypothesized as existing mainly in the medial entorhinal cortex. In turn, this activates the place cells in the hippocampus whose place fields constitute the observed trajectory.

The principal contribution of this work is a computational model of simulated motion that generates trajectories similar to those observed experimentally. The model integrates prior computational models of the functioning of the hippocampus and the adjacent entorhinal cortex. To model the motor routines that drive both physical and simulated motion, I developed extensions to Petri net software. Petri nets are a formalism that is well-suited to abstracting asynchronous, distributed processes such as those occurring in the brain. Methodological contributions include the extensions to open source Petri net software, which help make Petri nets a better candidate for use in computational neuroscience. In addition, the computational model is well-supported with automated tests, which makes it a practical example of how complex computational neuroscience models may evolve over time to incorporate more and more findings, increasing their usefulness.

Chapter 1

Introduction

1.1 Overview

During quiet waking periods a rat generates sequences of activation,"trajectories", in the hippocampus that depict, or "preplay", its future motion in two-dimensional space (Pfeiffer & Foster, 2013). The mechanism by which these preplay trajectories are generated is not known; the theory to be explored here is that they are part of a process of the animal simulating its future motion, a process that will be proposed to begin in cortical areas outside the hippocampus. An alternate hypothesis is that the trajectories originate in the hippocampus. An origin within the hippocampus has been proposed for similar trajectories that appear to "replay" the animal's past actions. These replay trajectories arise in somewhat different behavioral circumstances and are presumed to have different functions (Wilson & McNaughton, 1994; Lee & Wilson, 2002; Diba & Buzsáki, 2007), but the default hypothesis that all these trajectories are generated by processes local to the hippocampus would be parsimonious. The neuroscientific goal of this dissertation is to explore the theory that in

some circumstances trajectories are a byproduct of simulated motion, and are driven from outside the hippocampus.

Simulations can be defined as "imaginative constructions of hypothetical events or scenarios" (Schacter et al., 2012). There is substantial evidence for simulation in humans, for example during language understanding (Bergen, 2012). I suggest that a subset of simulation mechanisms may also be employed by rats. In rats, the focus will be narrowed to simulated motion in two-dimensional space. Simulated motion is the execution of a motor routine without activation of the motor neurons that would cause actual physical motion. I suggest that the simulated motion updates a path integration system, which has been hypothesized as existing mainly in the medial entorhinal cortex (McNaughton, Battaglia, Jensen, Moser, & Moser, 2006). In turn, this activates the place cells in the hippocampus whose place fields constitute the observed trajectory.

The core of this work is development of a computational model of spatial navigation during simulated motion, called the Simulated Motion System (SMS). Execution of SMS under various parameter settings is explored, and the results suggest that the system's behavior is consistent with some of the current understanding of spatial behavior in rats. The methodological goal of this dissertation is to show the value of computational tools and methods that are used in SMS, some of which are new to computational modeling in neuroscience.

The computational model will make use of Petri nets, a formalism developed for the modeling of distributed systems. A Petri net defines states, and transitions between states. Directed arcs define the conditions under which transitions execute, to move the Petri net from one state to another. Petri nets execute without a centralized point of control, which makes them suitable for modeling asynchronous and concurrent processes, including neural processes. A specific application of Petri nets has been to model behaviors as parameterized motor routines, termed X-nets. Previously used in modeling aspects of language, X-nets are directly applicable to modeling spatial motion. Extensions to Petri nets, developed in prior work by

the author, enable larger Petri nets to be built from smaller Petri nets, and to interact with other software components. This makes it feasible to use Petri nets as part of a larger system to model complex processes, such as those to be explored here. The Petri nets make visible important behaviors and events and the connections between processes.

The computational model is a system of interacting components, several of which are adapted directly from long-standing models that have proven useful in accounting for observations about the hippocampus and related regions. The contribution made here is not the addition of a new model for any particular brain region. Rather, it is in combining several component models and demonstrating that their interaction helps us understand how simulated motion might work.

An under-appreciated aspect of model usefulness is the ability to adapt to new facts. As software systems, models may be more or less easy to change. SMS was built using a constant reliance on small automated tests. Such tests precisely define each new behavior that the software needs to demonstrate, even prior to the new code being written. During development, the tests also assure the developer, many times per day, that the system continues to demonstrate all its expected behaviors. This makes it straightforward to update "test-driven" systems such as SMS, to address new facts.

Adaptability is an engineering consideration, not a scientific finding. But adaptability has a practical scientific value, as it keeps us from having to start over with a new computational model when the facts change. In the case of simulated motion, there is much active research that might require that SMS be modified to address new findings. Recent findings are inconclusive regarding a central neuroscientific prediction of the simulated motion hypothesis, that there be communication from the entorhinal cortex to the hippocampus during simulated motion. O'Neill, Boccara, Stella, Schönenberger, and Csicsvari (2017) found little evidence for such communication; Yamamoto and Tonegawa (2017) found significant evidence. Resolving these discrepancies may require a more realistic model. Particularly in the area of the

hippocampus, SMS uses a very simplified model, which could not account for many of the complexities of hippocampal functioning that might become relevant as the research evolves. The concluding chapter will address how such components could be replaced in future work, while preserving the overall structure and value of the system.

1.2 Spatial navigation

Pfeiffer and Foster (2013) found that rapid hippocampal sequences of activation of place fields in rats in an open arena during quiet waking periods predict the subsequent movement of the rats. In their experiment, rats learned to leave a home location to forage randomly for a reward, and upon finding the reward, to then return to the home location to receive a second reward. The rats' performance in finding the second reward at the home location was significantly faster than their performance in finding the first reward at the random location. During quiet waking periods, while the rat was consuming a reward, the recordings showed the generation of trajectory events, "temporally compressed, two-dimensional trajectories across the environment". Some of these trajectories ended at the home location, and others ended at other locations. Some previous experiments had interpreted such trajectories as replay of past paths of motion, but the analysis by Pfeiffer and Foster makes clear the trajectories predict future behavior rather than recall previous behavior.

These findings suggest a simple hypothesis, that the animal is simulating its motion, in effect asking, "what would happen if I moved in that direction?" To understand what that hypothesis would mean requires a review of some of the very large literature on spatial navigation in rats.

Understanding spatial navigation in rats begins with the discovery that some neurons in the hippocampus of freely moving rats fire reliably when the animal is in a given location of the

current environment, and are largely silent elsewhere (O'Keefe & Dostrovsky, 1971; O'Keefe, 1976). The activity of these place cells varies with two major modes of rat behavior, characterized by patterns in the local field potentials (LFP) measured in the hippocampus (O'Keefe & Nadel, 1978). LFP oscillations in the hippocampal and retrohippocampal formations in the frequency range of 6-10 Hz are known as theta, and are associated with exploration of novel environments or novel objects in known environments, and movement that involves spatial displacement. Large-amplitude irregular activity (LIA) is a second pattern found in the hippocampus, with a major frequency slower than theta; it is observed during quiet periods without spatial displacement, e.g., while sitting quietly, grooming, eating, drinking, or during slow-wave sleep. At times during LIA there are sharp wave ripples (SWR), consisting of a large amplitude change in LFPs accompanied by a high-frequency burst of activity, originating in CA3 and propagating to CA1, the subiculum and deep layers of the entorhinal cortex (Buzsáki, S., & Vanderwolf, 1983; Buzsáki, 1986; Buzsáki, Horvath, Urioste, Hetke, & Wise, 1992).

As a rat moves from place to place during active exploration, sequences of place cells are activated, trajectories, whose place fields depict the animal's physical path. Trajectories are observed under circumstances other than physical movement. Some appear to replay the animal's past actions, during REM sleep (Louie & Wilson, 2001), or during SWRs in slow-wave sleep (Wilson & McNaughton, 1994; Lee & Wilson, 2002), or during SWRs in quiet waking periods (D. J. Foster & Wilson, 2006; Diba & Buzsáki, 2007; Karlsson & Frank, 2009). In the context of a general theory of the relationship between the hippocampus and the neocortex (McClelland, McNaughton, & O'Reilly, 1995), replay has been suggested as a candidate for a mechanism to repeatedly update the neocortex with recent experiences, so that the neocortex can gradually learn the structure of those experiences. Under this hypothesis of their function, it is plausible to think of replay trajectories as originating in the hippocampus through some local process.

Other trajectories appear to preplay the animal's actions, i.e., to reflect the future activity of the animal. This has been observed during active exploration, particularly during vicarious trial and error (VTE) behavior at a choice point (Johnson & Redish, 2007). During VTE, animals are observed to orient briefly toward each possible path in turn. Simultaneously, hippocampal trajectories are observed, sweeping forward down the available paths in turn. Trajectories that reflect future activity are also observed during quiet waking SWRs (Pfeiffer & Foster, 2013). Although they seem to have different functions, as a practical matter, replay and preplay trajectories are often found together and may be difficult to distinguish (Diba & Buzsáki, 2007; Davidson, Kloosterman, & Wilson, 2009; Singer & Frank, 2009).

Given the similarities and co-occurrence of replay and preplay trajectories, it is parsimonious to suggest that they all originate in the hippocampus. An alternative to this hippocampal hypothesis is the simulated motion hypothesis, which suggests that preplay trajectories are part of a process of simulated motion, where the animal simulates exploration of the environment in a given direction to ascertain whether that direction seems likely to be rewarding. In this hypothesis, the origin of a preplay trajectory is external to the hippocampus.

The simulation hypothesis is situated here within the context of several theories regarding the function of the hippocampal and retrohippocampal areas in support of spatial navigation (reviewed in Navratilova and McNaughton (2014)). A basic concept is that of an attractor network, a dynamic system consisting of an N-dimensional space whose state at a given moment is defined by a vector, and where attractor states can be identified, toward which states in a close neighborhood of the attractor tend to evolve over time. When configured as an autoassociative network, where linkages exist among all the elements that constitute a given system state, these models have been used to represent memory (Hopfield, 1982). The structure of the CA3 region of the hippocampus, with support from the dentate gyrus, is compatible with the hypothesis that CA3 functions as an autoassociative network (McNaughton & Morris, 1987; McNaughton & Nadel, 1990; Treves & Rolls, 1992).

Attractor models have also been used to represent aspects of the spatial position of an animal (McNaughton & Nadel, 1990). An example of this followed the discovery that the allocentric angular orientation of a rat with respect to a fixed landmark is represented by head direction cells (Ranck Jr, 1985), which appear to behave as a system (Taube, Muller, & Ranck, 1990a, 1990b). This means that if one head direction cell changes its preferred orientation of firing, other head direction cells will be observed to change their orientations in the same direction and with the same angular displacement. This head direction system can be modeled through a one-dimensional attractor model conceptualized as a ring (McNaughton, Leonard, & Chen, 1989; McNaughton, Chen, & Markus, 1991; Skaggs, Knierim, Kudrimoti, & McNaughton, 1995; Zhang, 1996; Redish, Elga, & Touretzky, 1996), where the animal's current orientation is represented by a bump, a population of cells whose activation is self-sustaining. Self-motion cues and visual cues will cause the bump to move, representing the updated orientation.

Many species are capable of path integration, tracking movements through an environment using self-motion cues, possibly corrected by sensory cues from landmarks, such that direct routes to arbitrary points elsewhere in the environment can be calculated as needed (Mittelstaedt & Mittelstaedt, 1980). Coupling the head direction model with another attractor model of the animal's position in two-dimensional space, McNaughton et al. (1996) and Samsonovich and McNaughton (1997) generated a model that accounts for path integration behavior and many findings regarding experimentally-induced discrepancies between self-motion and sensory cues (Gothard, Skaggs, & McNaughton, 1996; O'Keefe & Burgess, 1996; Knierim, Kudrimoti, & McNaughton, 1995; Sharp, Kubie, & Muller, 1990; Wilson & McNaughton, 1993). The model introduced the concept of a chart, an abstract plane where place cells are arranged such that the pattern of neural activation is a localized activity packet centered on the animal's head, when mapped to a given environment. Each different environment has its own chart.

This model pre-dated the discovery of grid cells in the medial entorhinal cortex (MEC), whose firing pattern resembles a pattern of tessellated equilateral triangles overlaid on the rat's environment, providing a stable metric for space (Hafting, Fyhn, Molden, Moser, & Moser, 2005). The model required relatively minor changes to accommodate grid cells, and made some predictions that were consistent with later grid cell discoveries. The accumulation of evidence provides significant support for the hypothesis that the path integration function and spatial attractor dynamics reside in the MEC, with collaboration from the hippocampus and lateral entorhinal cortex (LEC) (Colgin et al., 2010; J. K. Leutgeb, Leutgeb, Moser, & Moser, 2007).

In contrasting the roles of the medial and lateral entorhinal cortices, Deshmukh (2014) suggests that the MEC processes self-motion cues in support of path integration, while LEC generates both spatial and non-spatial representations from external sensory input. The introduction of the animal to a novel environment causes global remapping – allocation of a new chart in MEC. In turn, the population of grid cells in that new chart activates a new population of place cells in the hippocampus to represent place fields at the various locations on that chart. Changes in the inputs from LEC, on the other hand, cause rate remapping – modification of the firing rates of the place cells allocated to the current chart, to represent information about landmarks, the current task, and rewards (S. Leutgeb et al., 2005; Navratilova, Hoang, Schwindel, Tatsuno, & McNaughton, 2012).

A primary point for this work is that these models suggest that the spatial attractor dynamics of where the animal is on the current chart is driven out of the medial entorhinal cortex. The dynamics will cause one population of grid cells to be activated on one chart, representing the animal's current location, and the bump will move across the chart in response to self-motion cues. The grid cells project directly to place cells in the hippocampus, and the activation of place cells is an immediate consequence of the activation of grid cells one synapse upstream. Global remapping is interpreted as occurring when the attractor dynamics in MEC cause a

different, possibly novel, chart to be activated, causing a different population of grid cells to be activated, and consequently, a different population of place cells. This theory has different consequences for the hippocampal and simulated motion hypotheses for the origin of trajectories, consequences to be explored in the computational model.

Although physical proximity of place cells in the hippocampus does not imply physical proximity of their place fields (O'Keefe, 1976; O'Keefe & Speakman, 1987; Gothard et al., 1996), the establishment of sequences of place cells through the traversal of physical trajectories does establish a mapping of relative distance between physical locations in the hippocampus for a given environment. This could enable novel trajectories to be generated in the hippocampus, through some mechanism such as gradient descent toward locations that have been previously identified as rewarding. This is consistent with the hippocampal hypothesis of the origin of preplay trajectories.

There is some evidence, however, that SWRs containing preplay trajectories may occur prior to an animal's first physical exploration of an environment (Dragoi & Tonegawa, 2011, 2013; Ólafsdóttir, Barry, Saleem, Hassabis, & Spiers, 2015), (but see Silva, Feng, and Foster (2015) for a suggestion that these results have been incorrectly interpreted, and therefore that physical experience is required for expression of a place field). Under the spatial attractor model, these reports are not consistent with the hippocampal hypothesis of a hippocampal origin for preplay trajectories. Entry into a new environment should cause global remapping, driven by allocation of a new chart in MEC. The new place cells corresponding to this chart cannot be predicted endogenously within the hippocampus – they are allocated by a process external to the hippocampus, solely by virtue of their connectivity with the grid cells of the new chart. These results are consistent, on the other hand, with the simulated motion hypothesis. Simulated motion in the motor cortex might cause the active location on the new chart to move to chart coordinates that correspond to areas of the environment that

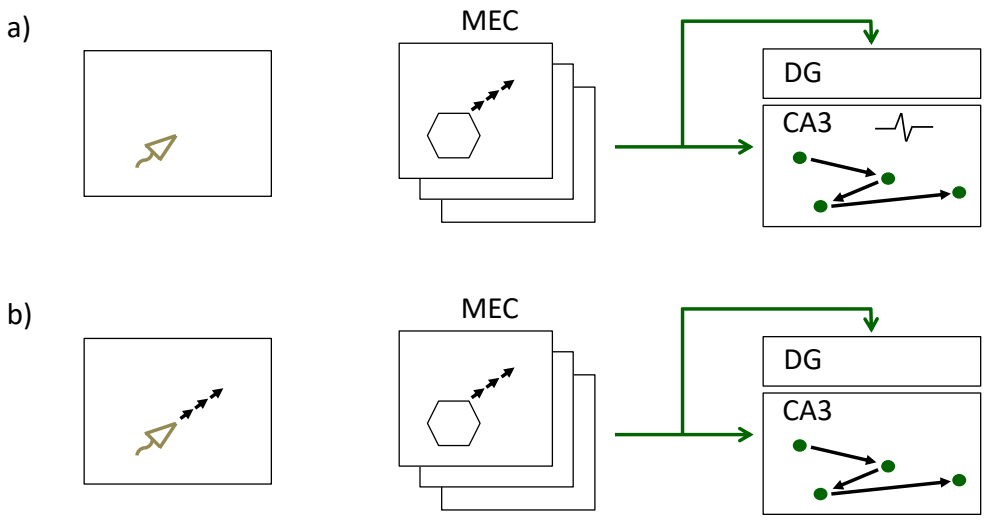


Figure 1.1: **a)** *left*: animal in quiet waking period, simulating motion toward novel locations. *middle*: simulated motion generates trajectory across MEC. *right*: MEC update to DG/CA3 activates new populations as part of SWR, observable as trajectory. **b)** *left*: animal in physical motion, retracing simulated path. *middle*: physical motion generates trajectory across MEC. *right*: MEC update to DG/CA3 re-activates populations, re-generating trajectory.

have not yet been physically explored. This would cause activation of the connected place cells for the first time, generating a trajectory prior to physical exploration (Figure 1.1).

The hypothesis of simulated motion implies that at least some active SWRs do not originate in the hippocampus, but are triggered by processes in motor areas, in turn updating medial entorhinal cortex and then hippocampus. This presupposes coherent communication between these areas, i.e., activity at the same frequency and phase across areas (Fries, 2005). Specifically, we should expect to find assemblies oscillating at the same frequency and phase at the first step upstream from the hippocampus, in medial entorhinal cortex. The research on SWRs largely contradicts this supposition, instead suggesting that SWRs originate in the hippocampus (Buzsáki et al., 1983; Buzsáki, 1986; Buzsáki et al., 1992; Chrobak & Buzsáki, 1994, 1996) (but with some small counterevidence in Chrobak and Buzsáki (1996), and see Norimoto, Matsumoto, Miyawaki, Matsuki, and Ikegaya (2013)). O'Neill et al. (2017) specifically looked for communication during quiet waking periods between the superficial layers

of MEC and the CA1 region to which those layers project. The study did find rapid trajectories in MEC, which is consistent with the simulated motion hypothesis. But there was little evidence of coherent communication with CA1 during these events. SWRs occurred in hippocampus, but at different times, and without much overlap in the place fields referenced between CA1 and MEC.

In a study of mice running on a long track, however, Yamamoto and Tonegawa (2017), found evidence of bursts of activity in superficial layers of the MEC, preceding SWRs in CA1[1] during quiet waking periods. Long tracks have been observed to be associated with multiple SWRs, that together cover longer distances than are typically found in a single SWR (Davidson et al., 2009). The authors observed alternating bursts of activity in such cases across both MEC and CA1, starting with MEC, as would be expected under the simulated motion hypothesis.

Weighing in on the question of communication between entorhinal cortex and hippocampus, Trimper, Trettel, Hwaun, and Colgin (2017) raise methodological concerns about some recent studies. They observe that cells in MEC fire in a regular spatial pattern, not just in one place as do their counterparts in hippocampus. They demonstrate that this can lead to spurious correlations in the firing rates across the two regions, potentially contributing to misleading conclusions.

We see that the research is equivocal on the communication between superficial layers of MEC and CA3 during quiet waking periods, but such communication is necessary for the viability of the simulated motion hypothesis. Absence of such communication would throw the hypothesis into serious doubt. It is possible that the discrepancies in findings stem from the underlying complexity of the dynamics of the hippocampus. The SMS model as used

[1]Both studies sampled from CA1, not CA3, but as SWRs propagate from CA3 to CA1, findings for CA1 have similar implications for CA3.

here does not have a model of CA3 that can address these dynamics. The implications of this for the evolution of SMS will be discussed later.

1.3 Adaptive value of simulated motion

The (Pfeiffer & Foster, 2013) results appear to establish the fact that rats are doing something that looks like simulated motion. The great majority of this work focuses on how simulated motion might work. Here I briefly speculate about the possible value to the animal of simulating its motion. Although at some level the animal in the Pfeiffer and Foster experiment already "knows" that home is rewarding, perhaps there is additional information to be gained by simulating the movement to home. As discussed earlier and in Appendix B, several studies involving spatial alternation tasks have found information about the task coded in the physical trajectory itself. The animal in this experiment is solving a type of spatial alternation task; perhaps the quiet waking SWRs trajectories that are generated carry information that enable the animal to more frequently predict which path will be rewarding. This generates a testable prediction: for quiet waking SWRs during spatial alternation tasks, it should be possible to predict the direction of the trajectory after the choice point, from the firing rates of the trajectory on the common path prior to the choice point.

Perhaps the simplest explanation of the value of simulated motion is as a lower-cost and faster method of exploring the environment. It takes less energy to think about moving in a novel direction than to do the physical movement, and trajectories in SWRs are at least an order of magnitude faster than physical motion. On the other hand, there is clearly sensory information available at each physical location that is not available during simulated motion. A simple analysis of this tradeoff is part of the study results.

Finally, perhaps simulated motion is a form of computation, to which the hippocampus is well-suited. Having found a reward at a particular location, the animal will associate a reward with that place. The hippocampus is able to repeatedly retrieve a place, and trigger other cortical areas to retrieve information associated with that place. Given this, simulated motion to a place and retrieving a rewarding association to that place would constitute a computationally low-cost way to find potentially rewarding future paths. It is certainly the case that there are mechanisms in neocortex that can also calculate such paths, but from an evolutionary perspective, these might be pre-dated by a hippocampal mechanism. In effect, the animal's planning could be as simple as simulating multiple trajectories across an environment until encountering one whose place cells cause the retrieval of a rewarding association.[2] If this proposal has value, it is only as a thought experiment about how animals might have benefitted from simulated motion at an evolutionary stage where the archicortex was present and neocortex was rudimentary. Possible modifications to SMS to implement such a planning mechanism will be discussed as future work in Chapter 7.

1.4 Contribution

This work makes two contributions. SMS provides a straightforward computational account in support of the hypothesis that some trajectories observed in the hippocampus may be the result of the animal simulating its motion. SMS makes extensive use of prior models of several regions, and integrates them into an extensible whole. Current evidence is equivocal on the simple version of the simulated motion hypothesis. Depending on how the evidence develops, components of the model could be modified to address the findings, while maintaining the usefulness of the overall structure.

[2]This would certainly be insufficient to master a task as complex as in Pfeiffer and Foster (2013), where every place in the arena is rewarding, but only under certain conditions. For such tasks, complementary learning systems theory (discussed below) suggests a neocortex is needed to infer the structure of the experiment.

The second contribution of this work is methodological. The extensions I have written to existing software make Petri nets and X-nets useful tools for high-level modeling of neural processes. Also, SMS incorporates a number of existing models into a larger system, which currently demonstrates a number of behaviors consistent with spatial navigation findings, and should be able to demonstrate a number of additional findings without major alteration. The approach taken to writing SMS enables the model to evolve in significant ways to incorporate new findings. In the near term, this could include more realistic models of CA3, which might enable the system to better account for recent findings.

1.5 Roadmap

To motivate the computational model, in **Chapter 2**, I review a portion of the literature on spatial navigation in rats, focusing on what is known about the hippocampus and entorhinal cortex and how this accounts for navigation behavior in rats. Part of the literature deals with some of the existing computational models that have attempted to account for significant parts of the known findings, and which are re-used within SMS. Explaining the computational model begins in **Chapter 3**, which covers Petri nets, X-nets, and the extensions written to make use of them within SMS. **Chapter 4** explains the details of the computational model, identifying those areas of the findings that are or are not addressed within SMS. **Chapter 5** explains important parameters of the model, and how the results were generated. **Chapter 6** reviews results from running the model across a limited parameter space. This demonstrates the support of the model for the theory of simulated motion. Finally, in **Chapter 7**, I conclude by reviewing the contributions of this work, and explaining how future modifications to the model could be made to address additional findings. In **Appendix A**, I discuss how SMS was built, with an emphasis on the importance of automated tests to its adaptability. **Appendix B** gives additional findings from the literature that SMS should

attempt to account for, as it evolves. **Appendix C** lists a number of testable neuroscientific predictions suggested by the simulated motion hypothesis and the SMS model. **Appendix D** summarizes the behaviors that SMS currently demonstrates, as well as those it could potentially demonstrate, with comments on the modifications that would be entailed.

Chapter 2

Spatial navigation

2.1 Overview

The hippocampus in rodents has been the focus of research interest for many decades, and major discoveries have arisen there that have implications for functioning throughout the brain. In this chapter, I review a portion of this very large literature that focuses on the contribution of the hippocampus and entorhinal cortex to navigation behavior in rats. Part of the literature deals with some of the existing computational models that have attempted to account for significant parts of the known findings, and which are re-used within SMS. As presently used within SMS, these models only exhibit a small portion of their possible range of behaviors, but they should be able to exhibit much more with future work. Additional spatial navigation findings are reviewed in Appendix B. The combination of these findings is then summarized in Appendix D as a set of behaviors that SMS currently generates, or could generate in the future. In the latter part of the chapter, I explain recent findings that potentially could not be accounted for under the present version of SMS. In the discussion of future work in the last chapter I will discuss how SMS could be modified to account for such

findings, as well as many other more established findings (possible modifications detailed in Appendix D). This study assumes a basic knowledge of neuronal structure and dynamics; the models used here do not attempt to account for that level of detail.[1] It is possible, however, that the future changes envisioned for SMS will need to incorporate models at that level of detail, e.g., spiking network models.

2.2 Place cells and navigation

The understanding of the role of the hippocampus and associated regions in support of spatial navigation begins with the discovery that some neurons in the hippocampus of a freely moving rat fire reliably when the animal is in a given location of the current environment, and are largely silent elsewhere (O'Keefe & Dostrovsky, 1971; O'Keefe, 1976). Such a neuron is a *place cell*, and the area of the environment to which it is responsive is its *place field*. Based on this insight, O'Keefe and Nadel (1978) gave a broad-ranging treatment of spatial navigation, organizing existing observations regarding the hippocampus and rodent behavior into a framework that continues to be useful.

O'Keefe and Nadel (1978) discussed different modes of navigation, distinguishing *routes* from *maps*. Routes are connected locations from a starting point to a goal, navigated perhaps from memory or by reference to a landmark or a compass orientation. Routes imply limited flexibility in navigation; an error in following a route may mean failure to reach the goal. In contrast, a map is a set of locations with spatial relationships, and possibly, additional information about what has been experienced at a given location. Navigating by use of a map is more complex than route-based navigation, but more flexible, because a map enables routes to be calculated between any desired points. Empirically, it was observed that route-

[1] Concepts in italics that are not described elsewhere are briefly defined in the glossary.

based navigation can be accomplished without a functioning hippocampus, but an intact hippocampus is required for map-based navigation.

Navigation makes use of multiple systems within the brain, depending on context. Inputs to navigation may include both sensory (*allothetic*) cues from the visual, auditory or olfactory systems, as well as internal (*idiothetic*) self-motion cues from the vestibular, proprioceptive and motor systems. Navigation uses a combination of frames of reference. *Allocentric* references orient the animal to features of the environment, e.g., "head towards the tower". *Egocentric* references orient the animal with respect to itself, e.g., "turn to your left".

2.3 Electrical activity and memory mechanisms in the hippocampus

O'Keefe and Nadel (1978) characterized rodent behaviors in terms of typical patterns of electrical activity observed in the hippocampus. Now referred to as *local field potentials* (LFP), the measurements of voltages in small areas of the brain vary with the firing of excitatory and inhibitory neurons, and the variations in different cortical areas fall into characteristic frequency ranges under different behavioral conditions.

O'Keefe and Nadel (1978) characterized the patterns that had been reported in various animals. In rats, the hippocampus shows periodic variations in LFP in a frequency range of 6-10 Hz. Known as *theta oscillations* or simply *theta*, this pattern is seen during exploration of novel environments or novel objects in known environments, and movement that involves spatial displacement. Theta is observed in the hippocampal formation and the parahippocampal region. A second pattern found in the hippocampus is *large-amplitude irregular activity* (LIA), whose major frequency is slower than theta. LIA is observed during quiet

periods without spatial displacement, e.g., while sitting quietly, grooming, eating, drinking, or during slow-wave sleep.

At times during LIA there are sharp waves (SPW) that are accompanied by short bursts of activity (Buzsáki et al., 1983). The combination has been termed a *sharp wave ripple* (SWR). Some trajectories, e.g., those observed in Pfeiffer and Foster (2013), seem to occur during SWRs. The "sharp wave" consists of a large amplitude change in local field potentials, triggered by a burst of firing in a sizeable population within the CA3 region of the hippocampus, and propagating from CA3 to the CA1 region and beyond, lasting from 30ms to 120ms. Within the time course of the sharp wave, there are high frequency oscillations, "ripples", at approximately 200 Hz, observed near the pyramidal cell bodies in CA1, and apparently controlled by the firing of inhibitory interneurons (Buzsáki et al., 1992). This pattern of inhibition controls the firing of the CA1 pyramidal cells; limiting the number of spikes in a SWR to as few as one, or occasionally two to three, locked to the negative phase of the ripple. Despite the low absolute numbers of spikes observed during SWRs, the probability of pyramidal cell firing during the presence of a SWR is several times greater than in its absence.

The hippocampus is also where another major discovery was made, concerning the mechanism by which experience changes the structure of the connections between neurons, thereby implementing one mechanism for memory of past events. Hebb (1949) proposed that the simultaneous firing of the neurons on either side of a *synapse* (a location where two neurons interact), would cause the efficiency of transmission to increase at that synapse; this became known as *Hebb's rule*. A group of neurons connected together by such strengthened synapses, such that activation of one neuron would lead to the activation of the others, was termed an *assembly*. Bliss and Lø mo (1973) reported that short high-frequency bursts of stimulation to neurons at the origin of the *perforant path* from the entorhinal cortex to granule cells in the dentate gyrus area of the hippocampus in anesthetized rabbits resulted in a sustained

increase in the efficiency of the synapses involved. This increase in efficiency lasted hours to days, and was termed *long-term potentiation* (LTP). The LTP mechanism implements Hebb's rule, and enables the establishment of cell assemblies. LTP was subsequently found in other major pathways in the hippocampal formation, and in many other areas of the brain.

A rodent forms stable place fields upon its first exposure to a novel location. This does not require LTP; the process depends only on the existing synaptic structure. However, LTP is required if the place fields are to be remembered across different exposures to the same environment. If LTP is blocked during exploration of a novel environment, the animal will again treat the environment as novel if it is removed from the environment and then reintroduced to it at a later time. When LTP is allowed to occur normally, the animal will recognize the environment on a later reintroduction, and the same place fields will be expressed as were created during its first exploration.

2.4 Models of memory and learning

2.4.1 Complementary Learning Systems

McClelland et al. (1995) proposed a general theory, the *Complementary Learning Systems* theory (CLS), about the relationship between the fast-learning mechanisms in the hippocampus and the slower-learning mechanisms in the neocortex. The hippocampus learns on one exposure to a novel location, while the neocortex learns over multiple trials. The authors suggest that the neocortex learns *structure*, "any systematic relationship that exists within or between the events that, if discovered, could then serve as a basis for efficient representation of novel events or for appropriate responses to novel inputs" (McClelland et al., 1995, 436). The neocortex is modeled as a neural network where knowledge about the structure

of events is implemented as the set of weights between nodes of the network. Under this model, the learning process proceeds optimally when novel information is presented to the neural network interleaved with existing information, such that the weights are updated in small increments. This enables the network to incorporate new knowledge with minimal interference with existing knowledge. The authors proposed that a function of the information being generated by the hippocampus during SWR events (e.g., during slow-wave sleep) was to provide the multiple trials necessary for the neocortex to incorporate new information into the weights of its network in a gradual fashion. In this model, at least some of the information in SWR events is *replay* of previous experience stored in the hippocampus. This contrasts with the proposed model of simulation, which involves SWR *preplay* of possible future actions.[2]

2.4.2 Attractor networks

One set of models for memory and spatial navigation is based on neural *attractor networks* (reviewed in Navratilova and McNaughton (2014)). An attractor network is a dynamic system that can be described as an N-dimensional space whose state at a given moment is defined by a vector. An *attractor* is a state or set of states of the system, toward which states in a close neighborhood of the attractor tend to evolve over time. A *point attractor* is a single locally stable limit point. If a system can support multiple such point attractors, then that system can act as a form of memory (Hopfield, 1982). A *continuous attractor network* is one where attractors are spread evenly across a "manifold". The dimensionality of the manifold is typically low. Absent outside input, the system will persist indefinitely in one of the attractor states. Given noise or external inputs greater than some uniform and relatively low barrier level, the system will migrate to another of the attractor states on the

[2]The terms preplay and replay are both used to refer SWR events that appear to reference future trajectories, perhaps because it can be difficult to infer the function of a particular trajectory. For my purposes, I will use the term preplay.

manifold (McNaughton & Nadel, 1990). This type of network can serve as a representation of an animal's position during motion in space.

2.4.3 Autoassociative networks

Mapping from one set of inputs to some existing pattern is a basic cognitive capability. Retrieving an image of an apple in response to hearing the word "apple" is *heteroassociative* mapping; the input pattern may be of a different type from the output pattern. The other case is *autoassociative* mapping, where linkages exist among all the elements that constitute a given system state (McNaughton & Nadel, 1990); the input and output patterns are of the same type. When such linkages exist, the system functions as an attractor network. The tendency of the system to move toward an attractor state when in the corresponding basin of attraction is termed *pattern completion*. For example, an incomplete or partially erroneous input such as "appl" might lead to the retrieval of "apple".

A linear algebraic implementation of an associative memory starts with an implementation of Hebb's rule, as described in McNaughton and Nadel (1990) (Figure 2.1). Two binary input vectors become associated with each other through the use of a square matrix where the elements of one vector map to the rows, and the elements of the other vector map to the columns. An input element value of one is considered "active". The element at the intersection of an active row and active column is itself activated. Retrieval of one vector given the other is accomplished first by multiplying the matrix, for example, in the row dimension, by the given vector, resulting in setting to zero all elements in the rows corresponding to inactive elements in the input vector. The columns are then summed, and integer division (discarding any remainder) is performed on each column by the number of elements that are active in the input vector. This will result in an output row where the values of one will correspond to the active positions in the vector to be retrieved, i.e., the

output row is equal to the row that was to be retrieved. Such a matrix can store a number of separate pairs of input and output vectors with a high probability of successful retrieval, up to a saturation level where approximately half the possible intersection points in the matrix are active. Reducing input vectors to some sparse representation will substantially increase the storage capacity of the network. The storage capacity will be maximized when the input vectors are orthogonal to each other. Finally, if the two input vectors are identical, the system has the property of autoassociation. This adds a capability of pattern completion; if an incomplete vector is presented to it, the system may return the complete vector.

Marr suggested plausible neural implementations of heteroassociative and autoassociative networks; hence McNaughton and Nadel (1990) termed them *Hebb-Marr* networks (Figure 2.2). In a Hebb-Marr network, there are *principal cells*, whose output state represents the desired vector. There are three inputs to the principal cells. One set of inputs has sufficiently powerful synaptic connections, *detonator synapses*, that their firing is highly likely to cause firing by the principal cells. The set of detonator synapses that fire represents the desired pattern to be stored by the network. A second set of inputs comes via exhaustive modifiable network connections. These *modifiable pathways* may not have initially active synapses, but the synapses are activated via an implementation of Hebb's rule, when inputs fire concurrently with the detonator synapses. The final set of inputs to the principal cells is inhibitory, and has the characteristic of increasing its inhibition linearly in the number of simultaneous inputs. If the inhibitory cells, *inhibitory interneurons*, are evenly and broadly connected to the principal cells, this will have the effect of the integer division described above, resulting in binary output. Principal cells with few active inputs will be suppressed from firing; principal cells with as many or more active inputs as the number of simultaneous inputs, will fire. In a heteroassociative network, this processing causes two input patterns to be associated with each other. In a basic autoassociative network, the pattern on the detonator and modifiable pathways is the same, and results in the network associating the input pattern with itself, i.e., "remembering" the input pattern.

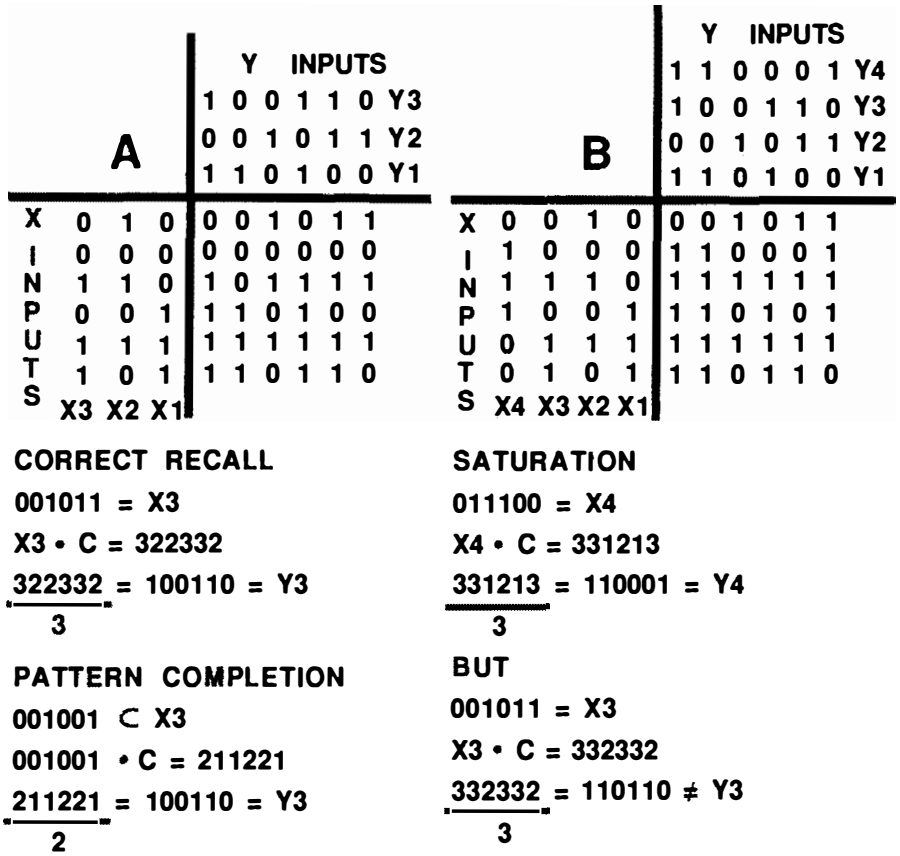


Fig. 1. *The fundamental principles of distributed associative memory can be easily illustrated by considering storage and recall of event pairs (X1:Y1, etc.) using a mechanism known as a correlation matrix. In this example, two sets of six binary information channels converge at a matrix C of 36 nodes. All nodes have an initial value of zero, but are permanently changed to a value of one whenever the corresponding X and Y input lines have the value one simultaneously. The matrix in* **(A)** *has experienced three sets of paired events. Any member of a pair can be recalled by multiplying the matrix by the other member, and dividing by the number of ones in the multiplier (decimals are ignored). Moreover, complete recall can also be accomplished by presenting only part of the corresponding pattern, provided that that part is unique (i.e. is not also a part of a different event). All three patterns stored in A can be correctly recalled, in spite of the fact that they overlap each other to some degree. Although the number of patterns such a matrix can store increases dramatically with the size of the matrix, there is always a limit above which recall of both new and older events become inaccurate. This is illustrated in* **(B)**, *where the storage of a fourth pair, which can itself be recalled correctly, has interfered with the recall of event 3. In general, this limit depends on the magnitudes and relative overlap of the input vectors. This saturation effect forms the basis for an experimental test of the relation between LTE and memory described in the text.*

Figure 2.1: Associative memory. Reprinted from Trends in Neurosciences, Vol 10, McNaughton, Bruce L and Morris, Richard G M, "Hippocampal synaptic enhancement and information storage within a distributed memory system", Pages 408-415, Copyright (1987), with permission from Elsevier

In a *recurrent autoassociative network*, the outputs from the principal cells are also inputs back to the same cells; these are termed recurrent connections, and occur when branches of the principal cell *axons* synapse on the *dendrites* of the same cells. In addition to supporting retrieval of the input pattern, this structure also supports a form of error recovery. In the event that a presented input pattern is corrupted and does not result in retrieving a pattern, the correct pattern may still be retrieved by progressively lowering the inhibition threshold in the network, such that matching on a smaller number of the input connections will cause the entire pattern to be retrieved. If this still does not result in retrieval, the pattern can be treated as "novel", and stored as a new pattern.

Recurrent autoassociative networks also support storage of what Hebb termed *phase sequences* – representations of successive sensory-motor events. Depending on timing, the inputs in time t will be fed back through the recurrent connections to be associated with the inputs in time $t+1$. This can continue through multiple periods, resulting in the storage of temporally-related events. Later, retrieval of the first event in the sequence will enable the retrieval of each later event in succession.

As applied to the systems to be discussed below, the networks of interest are recurrent autoassociative networks; they will be referred to simply as "autoassociative networks".

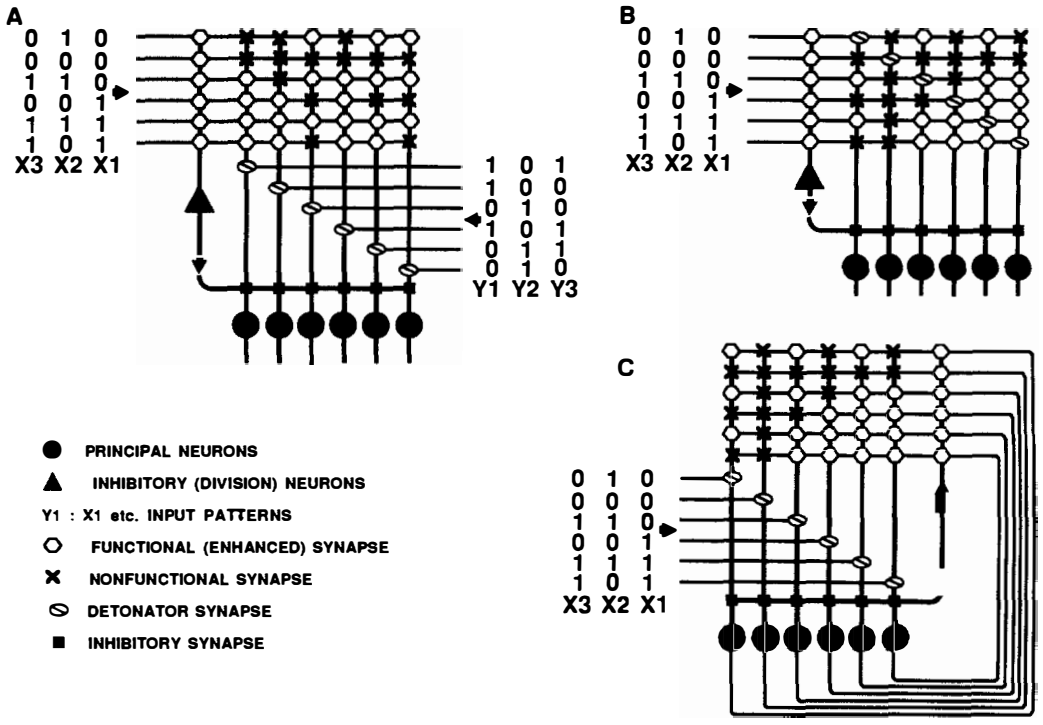


Fig. 3. (A) *A simple 'neuronal' implementation of the heteroassociative correlation matrix of Fig. 1A. The Y input fibres have exclusive control of the output cells by virtue of their terminating in 'detonator' synapses. The detonators also initiate 'Hebbian' synaptic enhancement of those terminals from the non-specific X input that were co-active with the particular Y input. Accurate recall is enabled by the inhibitory interneuron which is presumed to feed forward an inhibitory signal that divides the excitation of the principal neurons by the number of X fibres active in the current event. In real nervous systems, this division may be approximated by inhibitory synapses whose equilibrium potentials lie near the postsynaptic resting potential, and hence shunt the synaptic current arriving at the soma via the dendrites. While a single detonator input has been used here for simplicity, the idea should not be taken too literally. The same general principles apply when convergence of several or many inputs are required to initiate postsynaptic activation, with the exception that the output pattern will be some transform of the input, rather than identical to it.* **(B)** *A simple linear autoassociative network which has the ability to complete a stored representation when presented with only part of the original. This network is equivalent to making the X and Y inputs of Fig. 2A the same, and might be implemented simply by having each input fibre make a small number of detonator contacts, and a large number of weak, but modifiable, ones. Evidence for such a configuration has been detected within the fascia dentata.* **(C)** *A* recurrent *autoassociative network which associates the input to the principal cells with their own output. Depending on the details of its implementation this net can act as a pattern completion device with the additional capacity to keep patterns active by reverberation. If this reverberatory activity is not explicitly silenced before presentation of new information, this network will store event sequences and recall them from a fragment of the initial event in a sequence.*

Figure 2.2: Hebb-Marr networks. Reprinted from Trends in Neurosciences, Vol 10, McNaughton, Bruce L and Morris, Richard G M, "Hippocampal synaptic enhancement and information storage within a distributed memory system", Pages 408-415, Copyright (1987), with permission from Elsevier

2.5 Anatomy of the hippocampal formation and parahippocampal region

The *hippocampal formation* (Figure 2.3) is part of *allocortex* (or *archicortex*), a set of old structures in evolutionary terms, whose structure is largely conserved across mammals. The hippocampal formation, located bi-laterally in the medial temporal lobe, consists of the *dentate gyrus* (DG), sub-areas of the *cornu ammonis*, *CA1*, CA2 and *CA3*, and the *subiculum* (SUB). The *neocortex* or simply *cortex*, is a set of systems that support a variety of functions, but which share a common *six-layer cellular structure* and basic information processing characteristics. There is a continuous connection from neocortex to the three layers of the hippocampal formation, passing through transitional structures known as the *parahippocampal region* (or *retrohippocampal region*), and consisting of the *presubiculum* (PrS) and *parasubiculum* (PaS) and the *entorhinal cortex* (EC). EC consists of the *medial entorhinal cortex* (MEC) and the *lateral entorhinal cortex* (LEC)(Moser et al., 2014).[3]

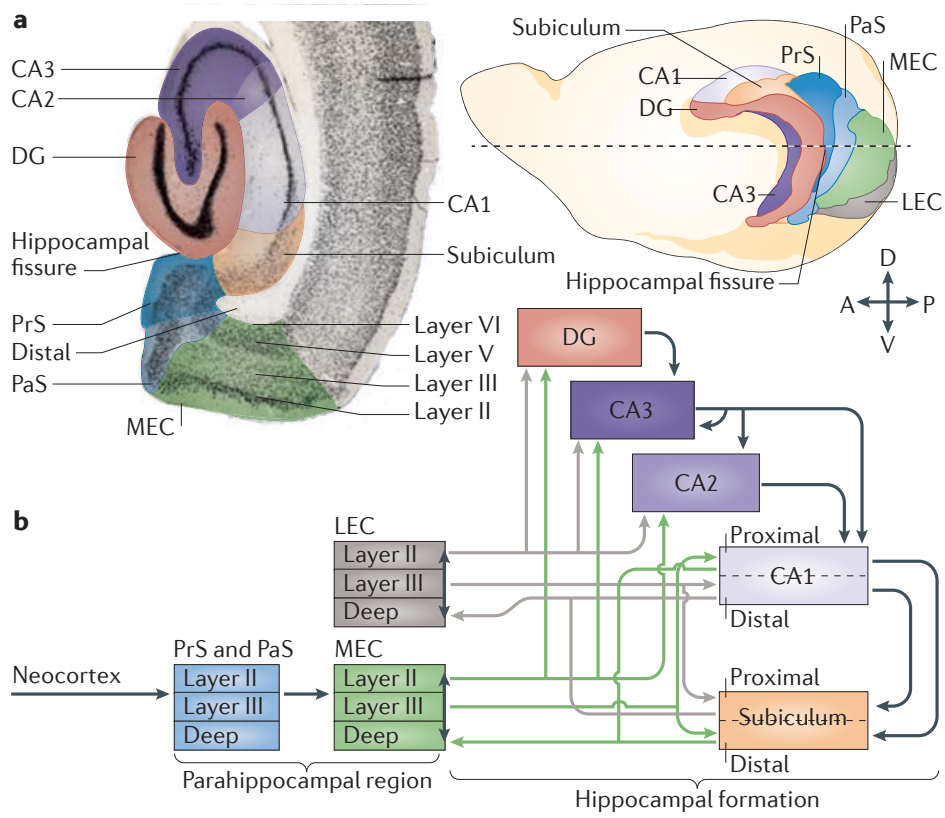
2.6 Hippocampus

The striking feature of the hippocampus is the place-specific firing of its principal cells, the *pyramidal cells* of CA3 and CA1. It is important to note, in advance of the consideration of the interaction between movement and navigation, that place-specific firing only occurs in the context of at least potential movement. Place cell firing is abolished in animals that are restrained and therefore unable to move (T. C. Foster, Castro, & McNaughton, 1988). Further details of the three main areas of the hippocampus, dentate gyrus, CA3, and CA1, will take place as part of discussion of models of their function.

[3]In the SMS model, I use the term HippocampalFormation to encompass models that correspond to both the hippocampus and the entorhinal cortex, not distinguishing the hippocampal formation from the parahippocampal region.

Anatomy of hippocampal formation and parahippocampal region

In the current standard connectivity model of the hippocampal formation and parahippocampal region (see the figure, part **b**), the MEC provides input to the hippocampal formation, with layer II projecting to the DG, CA3 and CA2, and layer III projecting to CA1 and the subiculum. CA1 and the subiculum provide output to entorhinal cortex layer V. All entorhinal layers seem to be reciprocally connected (indicated by the double-headed arrows). This connectional route, in green, is paralleled by a similarly organized route starting and ending in the LEC, indicated in grey. The two pathways converge onto single neurons in the DG, CA3 and CA2 but target different neurons in CA1 and the subiculum. Projections from and to the MEC link to neurons in CA1 close to CA2 (proximal) and neurons in the subiculum close to the PrS (distal), and the opposite pattern holds for projections from and to the LEC. Inputs selective for the MEC originate from the PrS and the PaS.



A, anterior; D, dorsal; P, posterior; V, ventral.

Figure 2.3: Anatomy of hippocampal formation and parahippocampal region, Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Reviews Neuroscience ("Grid cells and cortical representation", Moser, E. I., Roudi, Y., Witter, M. P., Kentros, C., Bonhoeffer, T., and Moser, M.-B.) (2014)

2.6.1 CA3 as an autoassociative network

McNaughton and Morris (1987) suggested that the CA3 area, with support from the dentate gurus, might function as a Hebb-Marr autoassociative network, efficiently storing and retrieving sequences of place representations (Rolls (1987) briefly outlines a similar model, although without an explicit inhibitory mechanism). In this theory, the CA3 area functions as an autoassociative memory, by virtue of its large number of recurrent excitatory connections. The principal cells of CA3 are its *pyramidal cells*, whose axons, the *Schaffer collaterals*, project both to CA1, and back into the *apical dendrites* of CA3 (Figure 2.6 depicts the connectivity of these areas, including the connections relevant to the autoassociative model). This recurrent connectivity meets one requirement of a recurrent autoassociative network, that the outputs from one time period are part of the inputs to the next time period (McNaughton & Nadel, 1990).

From the perspective of CA3, the same information comes from entorhinal cortex via two paths, both of which are part of the *perforant path* that connects the EC to the hippocampus. The information comes directly from EC to CA3, via what is termed the *monosynaptic path*. The perforant path includes projections from both medial and lateral entorhinal cortex layer two. Information also comes indirectly to CA3, via what is termed the *trisynaptic circuit*, so called because it involves three connections. First, the perforant path connects the EC to the *granule cells* of the *dentate gyrus*. The second connection carries the information from the granule cells to the *pyramidal cells* in CA3, via the granule cell axons, the *mossy fibers*.[4] The two paths serve two distinct functions in the autoassociative model (McNaughton & Morris, 1987; Treves & Rolls, 1992). The mossy fiber projections (axons) from DG to CA3 are unusual in two respects, each of which is consistent with one of the requirements for an autoassociative network. Typically, axonal connections run across the branches of a dendritic

[4]The third connection is from CA3 to CA1, via the Schaffer collaterals. Among other destinations, CA1 connects back to the deep layers of entorhinal cortex, completing the circuit.

tree, connecting at relatively few synapses for a given cell, but potentially connecting with many cells. Mossy fiber connections, however, run roughly parallel to the CA3 pyramidal cell dendritic tree, making connections at many sites (on the order of 10^3) of a given cell's dendritic tree, and these connections are generally *proximal* to the cell body. This means the probability is high that the pyramidal cell will fire as a result of the granule cell's firing. This seems to meet the requirement for *detonator synapses* in an autoassociative network, synapses that fire with high probability to ensure that the target vector is conveyed accurately to the network.

The second characteristic of the mossy fiber projections is that a given axon synapses on only a small number of dendrites in CA3 (on the order of 10^1), and a given pyramidal cell in CA3 is likely to be contacted by a small number of different granule cells (McNaughton & Nadel, 1990). This means the probability that a given granule cell is connected to a given pyramidal cell is very low, and that a given granule cell's firing is likely to contribute to the firing of a small number of cells in CA3. In addition, the number of granule cells in the DG is roughly an order of magnitude greater than the number of cells in the EC that are the source of the perforant path, and the number of pyramidal cells in CA3. Taken together, these characteristics suggest that two functions of the dentate gyrus are to generate a sparse input vector to be conveyed to the autoassociative network in CA3, and to minimize the correlation between two input vectors (*orthogonalization*) (McNaughton & Nadel, 1990). Sparse representations are important to maximizing the storage capacity of an autoassociative network. Orthogonal representations minimize the likelihood of overlap between two inputs, thereby minimizing the probability of retrieving an incorrect pattern. This model suggests an explanation for the fact that physical proximity of place cells in the hippocampus does not imply physical proximity of their place fields (O'Keefe, 1976; O'Keefe & Speakman, 1987; Gothard et al., 1996). Although it is likely that the sensory inputs for places that are close in time and space will have much overlapping information, the process of expanding that information into a larger state space in the dentate gyrus, and then compressing it again for representation in

CA3, also while limiting the number of pyramidal cells receiving information from a given granule cell, can be expected to lead to very different representations of two input vectors in CA3.

In contrast to the small number of connections made from the DG granule cells to a given CA3 pyramidal cell, the recurrent connections from the CA3 pyramidal cells' Schaffer collaterals back to the CA3 dendrites are very numerous, on the order of 10^4. These recurrent connections with many modifiable synapses seem to meet the autoassociative network requirement for "modifiable pathways", or *modifiable synapses* (McNaughton & Nadel, 1990).

The final set of inputs to an autoassociative network is inhibitory, acting to suppress firing of pyramidal cells whose current active inputs do not meet the threshold activation level of the input pattern. There appear to be inhibitory interneurons in the hippocampus to meet this requirement. Specifically, the inhibitory response has the effect of shunting current away from the pyramidal cell dendrites, thereby reducing their probability of firing. As well, the response of at least some inhibitory neurons is significantly faster than that of the pyramidal cells, allowing the inhibition to be set up prior to the point where a pyramidal cell might reach threshold for firing. Also, there is evidence that the inhibitory response is linearly related to the activity of the inputs, a requirement for the integer division operation (McNaughton & Morris, 1987; McNaughton & Nadel, 1990).

In summary, for purposes of the computational model of simulation, I will treat the DG and CA3 components as implementing a Hebb-Marr autoassociative network, storing and retrieving information about places in the environment.

2.6.2 Hippocampus as a hash index to memories

Under the assumption that attractor dynamics is a useful model for understanding much of the functioning in the hippocampal and parahippocampal region, the next sections will include discussion of where the attractor dynamics resides. As just discussed, CA3 has the autoassociative network structure necessary to support those dyamics, but there are alternatives. These in turn raise the question of what additional or alternative functions might be supported by CA3. Colgin et al. (2010) suggest that the recurrent connections of CA3 could support the encoding of phase sequences of place fields as trajectories. Alternatively, or in addition, the phase sequences could function as index-code sequences to enable the retrieval of episodic memories whose details are stored in various areas of neocortex. Trajectories will be discussed later; in this section, I will briefly review theories of how information stored in the hippocampus could organize memories elsewhere in the cortex.

Under complementary learning systems theory, the hippocampus and neocortex (McClelland et al., 1995) require some coordination mechanism. Teyler and DiScenna (1986) suggested that the output of the hippocampus might be the equivalent of an "index" to memories stored in the various cortical areas. In this *memory indexing theory*, each cortical area stores memories of the inputs it processes (visual, auditory, etc.). Cortical areas typically have the recurrent connectivity necessary to implement autoassociative networks. In this view, and in Paller (1997), an event in memory consists of local memories stored in various cortical areas that are tied together via a common index created by the hippocampus. In the terms suggested by Paller (1997), the population that represents a memory in a single cortical area (e.g., visual or auditory cortex) is a *neocortical ensemble*, while the collection of memories distributed across multiple cortical areas related to a single episode is a *neocortical consortium*. The orthogonal representation created by the hippocampus might serve as a unique index to the current episode, and could be stored as part of the local memory in each cortical area. Recollection of the entire episode would then consist of sending the index

from the hippocampus to the cortical areas, which would perform pattern completion based on the index to retrieve the complete memory, including the local component. As originally proposed by Teyler and DiScenna (1986), it was suggested that the hippocampus maintains a map of the functional structures of the cortex, so as to address them directly through the index. This was later found not to be neuro-biologically plausible, but it seems not to be critical for the theory. Rather than thinking of the hippocampal output as an index into some cortical data structure, it may be more plausible to think of it as a "hash", a vector generated from the episode that is unique, for practical purposes. The communication from the hippocampus may be analogous to a "broadcast" of the hash to all cortical areas, each of which attempts to perform pattern completion. Those areas for which pattern completion succeeds then retrieve the corresponding local memory, and the ensemble of retrieved memories constitutes recollection of the entire episode.

As originally suggested by McClelland et al. (1995), repetition of memories promotes the gradual update of semantic knowledge in the cortex. Paller (1997) suggests that this takes the form of cortical-to-cortical connections that strengthen over time with co-activation, such that hippocampal input is no longer necessary for recollection of some of the information related to the episode. Unlike other processes related to the retention of information, this *consolidation* process is not automatic, but requires some sort of active rehearsal during awake and/or sleep periods, which is a function of the relevance of the information to the animal's ongoing cognition. This active rehearsal process is proposed as a major function of *replay*, discussed in Section 2.8.

2.7 Path integration and spatial navigation

There is growing evidence that the parahippocampal region implements a *path integration* function. Path integration means keeping track of one's movements through an environment,

using self-motion cues, such that direct routes to arbitrary points elsewhere in the environment can be calculated as needed. Path integration is supported by particular types of cells, and interactions among cell assemblies; we will review them in the context of attractor models of their function.

2.7.1 Head direction cells and modeling the animal's orientation

First reported by Ranck Jr (1985), *head direction cells* respond to the horizontal orientation of the animal's head, independent of the animal's activity, location in the environment, or the position of the animal's trunk (Taube et al., 1990a). Originally reported in the *postsubiculum* (the dorsal portion of the presubiculum), head direction cells are found in several areas, including the medial entorhinal cortex (Sargolini et al., 2006). Within a population of head direction cells, orientations are represented equally through 360°. The firing rate for a given cell is strongest in one particular direction, and the firing rate falls off linearly on either side of the preferred direction. The preferred direction and firing rates of head direction cells are stable for long periods of time, at least on the order of days (Taube et al., 1990a).

Head direction cells are allocentric; they are oriented to features of the environment, e.g., a prominent visual cue (Taube et al., 1990b). If the visual cue is rotated to a different position in the environment, the preferred directions of the population of head direction cells will rotate uniformly to follow the cue. This suggests that the population of head direction cells functions as a system, such that changes propagate to all the cells in the population. The updates to the head direction cells are primarily from *idiothetic* (self-motion) cues, including cues from the vestibular, proprioceptive and motor systems. These self-motion cues enable the animal to maintain a representation of the animal's current orientation, a key component of path integration, although in the absence of periodic correction from visual cues, errors will accumulate over time (Mittelstaedt & Mittelstaedt, 1980).

An attractor network can be used to model the head direction system. McNaughton et al. (1989) suggested a basic extension of the Hebb-Marr network model to enable the conditional association of movements with the set of features that characterize a particular local environment. This results in a population of cells that represents the conjunction of movements and environment features. Projecting this information back into the Hebb-Marr network at each time period enables the network to associate the current view with the previous view and the movements that led to the current view. This is an example of the ability of Hebb-Marr networks to store sequences of information.

Using this model, and adding attractor dynamics, enabled McNaughton et al. (1991) to model the head direction system. Combining head direction cells with angular velocity self-motion input enables the head direction system to represent a new direction following a detectable motion to the right or left. Combining this with input from a local view of the environment enables the system to correct for error due to drift that would otherwise accumulate over time. The authors suggest that place cells from the hippocampus may provide the local view information, and that cells in the posterior parietal visual areas and posterior sensorimotor cortex may provide the angular velocity input, consisting of some combination of ideothetic cues.

A current version of an attractor model for head-direction is summarized in McNaughton et al. (2006) (see Figure 2.4); detailed implementations are described in Skaggs et al. (1995), Zhang (1996) and Redish et al. (1996). The network can be represented as a ring, where each cell on the ring has a preferred direction. A given cell is connected to nearby cells on the ring with synaptic strength declining with distance away on the ring. Note that the ring is a logical construct defined by synaptic connections, and does not imply any particular physical topography of the head direction cells. Global feedback inhibition suppresses firing in all but the most active cells. These attractor properties generate a single activity "bump" in the network, representing the animal's current head direction.

Modeling the animal's change of head direction can be accomplished by two groups of conjunctive neurons. Combining the current head orientation from cells in the ring, with current motion from the vestibular system, one group responds to clockwise movement, and the other responds to counter-clockwise movement. This results in the activity bump moving around the ring in the direction of the animal's head movement.

2.7.2 Spatial attractor model

The head direction attractor model is one-dimensional, representing the angle of the animal's head in relation to some landmark. McNaughton et al. (1996) outlined an attractor model to account for an animal's movement in two-dimensional space; the model was later implemented and tested via simulation in Samsonovich and McNaughton (1997). At a high level, the extension to two dimensions might seem straightforward, but there is an important complication. The head direction model seems to need only a single network to represent the possible head directions. This network shifts as a unit upon entry into a known environment, orienting to a known salient cue. Place fields, however, have a fixed relationship to one another only within a single given environment. Upon entry to a new environment, a given set of place cells will form new place fields whose spatial relationships are uncorrelated with those of other environments. To model this, McNaughton et al. (1996) introduce the concept of a *path integration reference frame* (later simply termed a *chart* (Samsonovich & McNaughton, 1997)), an abstract plane, where place cells are arranged symbolically, such that the pattern of neural activation is a localized activity packet centered on the animal's head, when mapped to a given environment. Each different environment has its own chart. Upon entry into a known environment, the uncorrelated nature of the charts results in only a single activity bump emerging, on a single chart.

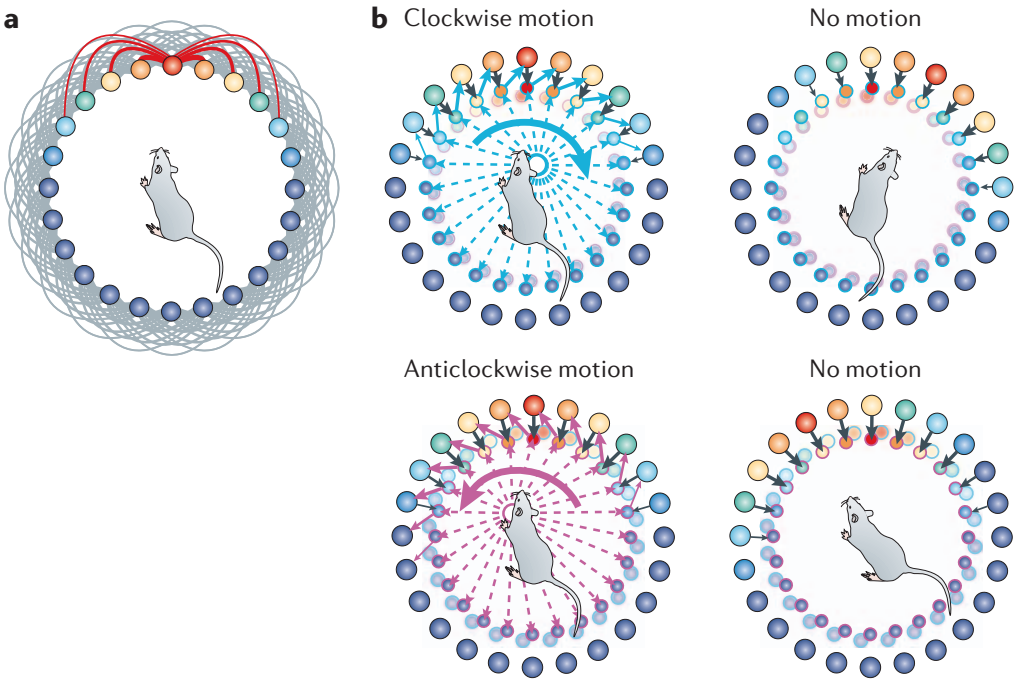


Figure 1 | **One-dimensional attractor map model for head direction encoding based on neural integration of head angular velocity signals. a** | Head direction cells are arranged symbolically in a circle in order of their relative head directional preferences. Each cell (coloured dots) connects with nearby cells with a synaptic strength (or connection probability) that declines as a function of distance (red and grey lines). The network is subject to global feedback inhibition (not illustrated) that limits the total neural activity. Activity in such a network has a most probable configuration in which the activity is focused at one point and declines with distance from that point (warm colours represent high activity, progressively cool colours represent progressively lower activity). Such a network would keep track of head direction if the hill or 'bump' of activity could be made to rotate around the ring in correspondence with changes in head direction. **b** | Rotation of the bump in the clockwise or anticlockwise directions can be achieved by an intermediate group of two types of conjunctive neuron that receive information about head angular velocity from the vestibular system (dashed arrows) and information about current head orientation from the cells immediately above them in the outer ring. The intermediate group of cells must be of two classes: cells receiving information about clockwise motion project to the right of the cells in the outer ring from which they receive input, whereas cells receiving anticlockwise vestibular signals project to the left. These hidden layer cells drive the activity bump in the corresponding direction around the ring. In the absence of motion, activation of all hidden layer cells is assumed to be below threshold. In this figure, only active connections are indicated, with the line thickness representing firing rate.

Figure 2.4: Ring attractor model for head orientation. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Nature Reviews Neuroscience ("Path integration and the neural basis of the 'cognitive map'", McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., and Moser, M.-B.) (2006)

To enable the model to perform path integration calculations, McNaughton et al. (1996) introduce angular and linear motion information. The angular information is provided by incorporating the head direction model previously described. Adding information on linear velocity in an analogous manner enables the activity bump on the active chart to track the animal's position from self-motion information. As in the head direction model, the final component is the incorporation of sensory information through conjunctive associations with particular locations on the active chart. This enables the correction of cumulative errors in path integration by reference to external landmarks. For a number of findings that provide evidence in support of this attractor model, see Appendix B.

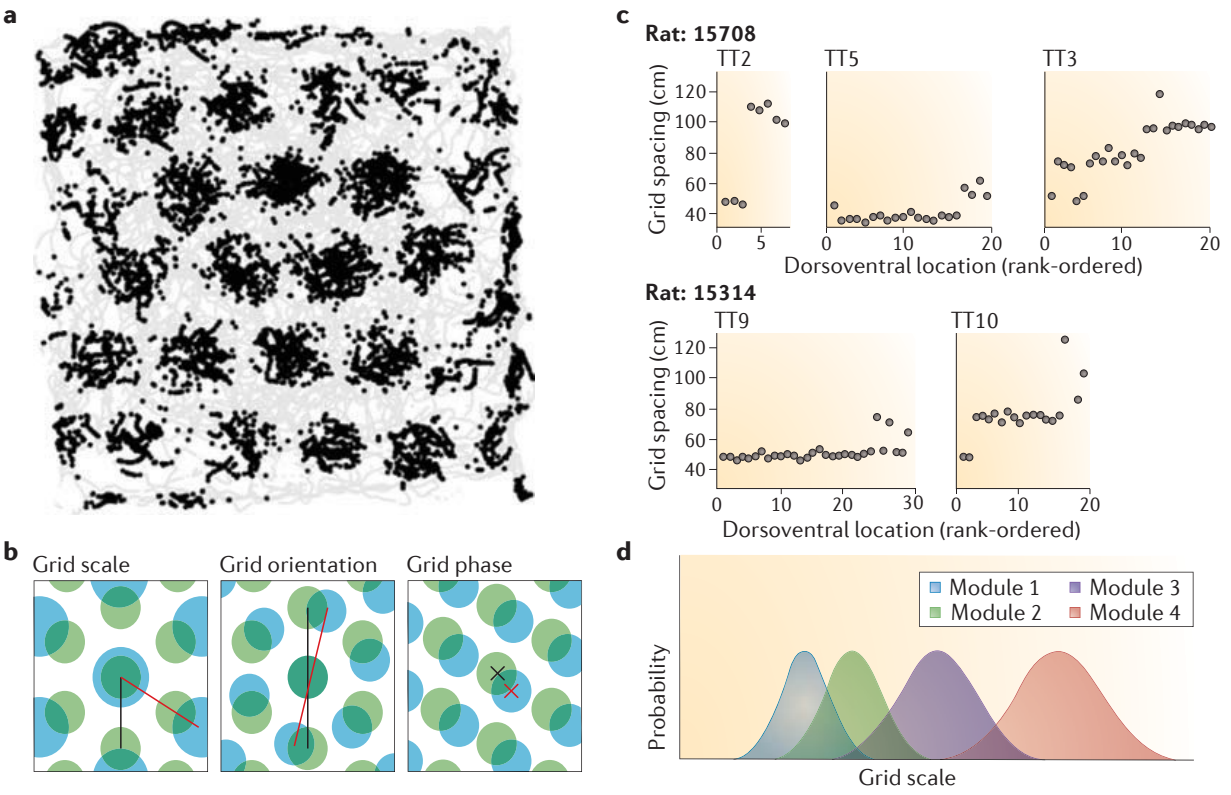
2.7.3 Grid cells and Medial Entorhinal Cortex

The abstract plane that defines a chart in Samsonovich and McNaughton (1997) was given a possible biological implementation by the discovery of cells in layer two of the medial entorhinal cortex whose firing pattern resembles a pattern of tessellated equilateral triangles overlaid on the rat's environment (Hafting et al., 2005). The original model had addressed the problem of how to avoid "falling off the edge" of a chart by defining the plane as a torus of rectangular fields, repeating at regular intervals. The newly discovered *grid cells* had multiple firing fields, each surrounded by an inhibitory field. The distances between firing fields for a given cell was constant across the environment, resulting in a distinctive hexagonal pattern (Figure 2.5), similar to the toroidal model, but using a rhombus rather than a rectangle as the base unit. The vertex locations of the triangles were correlated with the phase of firing of the cell. Neighboring cells in the MEC showed firing fields of the same size, spacing, and orientation (angle of a line passing through the vertices of the hexagon), but the phases varied, suggesting that the animal's entire environment could be represented by the population of grid cells. Consistent with the spatial attractor model, grid

fields were found to rotate with external cues, to be expressed immediately on entry to a new environment, and to persist in darkness.

Moving from dorsal to ventral locations in medial entorhinal cortex, the physical spacing between locations and the size of the firing field of grid cells both increased monotonically with distance from the postrhinal border. This was striking because it was analogous to a long-standing observation about hippocampal place fields, that they increase in size and decrease in spatial resolution moving from dorsal to ventral locations in the hippocampus (Jung, Wiener, & McNaughton, 1994). Self-motion cues appear to drive the underlying mechanism that determines the scale of the spatial representation. On a circular track, Terrazas et al. (2005) trained rats to walk (WALK) or drive themselves (CAR) [5] to locations around the track for a reward. Optic flow and vestibular cues provided constant feedback, but the self-driven trials removed the proprioceptive feedback and/or motor commands (*efference copy*) associated with ambulatory motion. Self-motion cues were further reduced in a third set of trials, where the entire "world" of the track, floor and surrounding curtains rotated around a fixed vehicle (WORLD). This removed vestibular cues, leaving only optic flow cues. The authors observed a drop in theta amplitude for the CAR and WORLD scenarios from the WALK scenarios. As theta amplitude is known to increase with the animal's speed, this suggests that the animal perceived itself as moving more slowly. In addition, the information content of place fields decreased and their size increased, in moving from the WALK to the CAR scenarios, and again in moving from the CAR to the WORLD scenarios. The head direction signal was presumably intact, making it appear to the animal that it was moving more slowly around a smaller track. The conclusion was that self-motion velocity information acts to change the gain on the spatial scale of place fields. Maurer, VanRhoads, Sutherland, Lipa, and McNaughton (2005) concluded that the increase in place field size moving from

[5]The "Rat Train Apparatus" (RTA) was a platform with a controlling lever, mounted on a G-scale toy train.



Basic properties of grid cells. **a** | Spatial firing pattern of a grid cell from layer II of the rat medial entorhinal cortex (MEC). The grey trace shows the trajectory of a foraging rat in a 2.2 m wide square enclosure. The locations at which the grid cell spikes are superimposed on the trajectory are shown in black. Each black dot corresponds to one spike. Note the periodic hexagonal pattern of the firing fields of the grid cell. **b** | Cartoons of firing patterns of pairs of grid cells (shown in blue and green), illustrating the differences between grid scale, grid orientation and grid phase. Lines in left and middle panels indicate two axes of the grid pattern (which define grid orientation); crosses in the panel on the right indicate grid phase (*x*–*y* location of grid fields). **c** | Modular organization of the grid scale. Grid spacing is shown as a function of position along the recording track in the MEC, with cells (represented by grey circles) rank-ordered from dorsal to ventral and one panel per tetrode (TT). On each tetrode, grid spacing increases in discrete steps. **d** | A schematic showing that the increase in grid scale across modules follows a geometric progression rule. From one module to the next, average grid scale increases by a constant factor (1.4 in this case). Part **a** is reprinted from Moser, E. I. & Moser, M. B. Grid cells and neural coding in high-end cortices. *Neuron* **80**, 765–774 (2013)[229]. Copyright (2013), with permission from Elsevier. Part **c** from REF. 24, Nature Publishing Group.

Figure 2.5: Grid cell properties. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Reviews Neuroscience ("Grid cells and cortical representation", Moser, E. I., Roudi, Y., Witter, M. P., Kentros, C., Bonhoeffer, T., and Moser, M.-B.) (2014). (a) Reprinted from Neuron, Vol 80, Moser, E. I. and Moser, M. B., "Grid cells and neural coding in high-end cortices", Pages 765–774, Copyright (2013), with permission from Elsevier (c) adapted from Nature 292, Stensola, H. et al, "The entorhinal grid map is discretized", Pages 72–78 (2012).

dorsal to ventral locations in the hippocampus could originate from a change in the gain in the self-motion signal being projected into the hippocampus.

The change in grid cell spatial scale moving from dorsal to ventral locations poses a challenge for the spatial attractor model. The attractor dynamics assume that the spatial scale is invariant on a given chart. The theory would have to postulate multiple charts, each with different spatial scales, to account for the observations. Stensola et al. (2012) tested the variation in spatial scale across a number of locations in the MEC, covering at least a third of the dorsal to ventral axis. The objective was to determine if the MEC spatial scale varies continuously or discretely. They found four to five discrete "modules", with a scale increment of approximately 1.42 between modules, moving from dorsal to ventral locations. Modules appeared to be independent of one another in several aspects. Within a module, the orientation was constant, but orientations varied across modules. Most important, the functional behavior of modules varied independently. When the square test enclosure was shrunk and later expanded, the fields associated with grid cells in different modules rescaled to different degrees. These findings are consistent with the attractor model, suggesting multiple charts, each with its own activity bump. The authors suggest that the convergence of signals from multiple charts might enable finer discrimination of position and greater ability to achieve independent representations of different positions in the hippocampus.

With the discovery of grid cells, McNaughton et al. (2006) suggested how the spatial attractor model of Samsonovich and McNaughton (1997) could be updated. Apart from having a biological substrate for the chart concept, the updated model proposed that the interaction of multiple grid cells at different spatial scales might generate a single spatial code for a corresponding place field in the hippocampus.

2.7.4 Multi-dimensional information in Lateral Entorhinal Cortex

In contrast to medial entorhinal cortex, the lateral entorhinal cortex is thought to convey both spatial and non-spatial information (Deshmukh, 2014) – objects and landmarks and their attributes. LEC connectivity largely parallels that of MEC (Figure 2.3). In CA3, MEC axons synapse proximally to the CA3 pyramidal cell bodies, while LEC axons synapse distally. This should lead to a stronger influence of the spatial components of MEC on the firing of CA3, modulated by the influence of LEC. In CA1, the distinction between MEC and LEC is manifested differently; MEC axons terminate on cells in that portion of CA1 that is proximal to CA3, while LEC axons terminate in the area of CA1 that is distal from CA3.[6]

In a review of current understanding, Deshmukh (2014) suggests that the functional distinction between MEC and LEC is best understood in terms of their information inputs. MEC generates a spatial representation from internal path integration information (self-motion cues), while LEC generates both spatial and non-spatial representations from external sensory input. Cells in LEC respond in the presence of objects in the environment, as well as odors. Cells can fire in the presence of familiar objects, novel objects, and moved or "misplaced" objects. LEC response appears to be modulated by input from the amygdala, suggesting that the salience of an object influences whether it will be represented in LEC.

Some cells in LEC have a strong spatial component to their firing, which might represent a local landmark – an object that is sufficiently close that the animal's movement in relation to it allows an estimation of distance. There are cells in LEC whose firing seems to be conjunctive for object and place – "what" and "where".

[6]The functional significance of this difference is unclear to me.

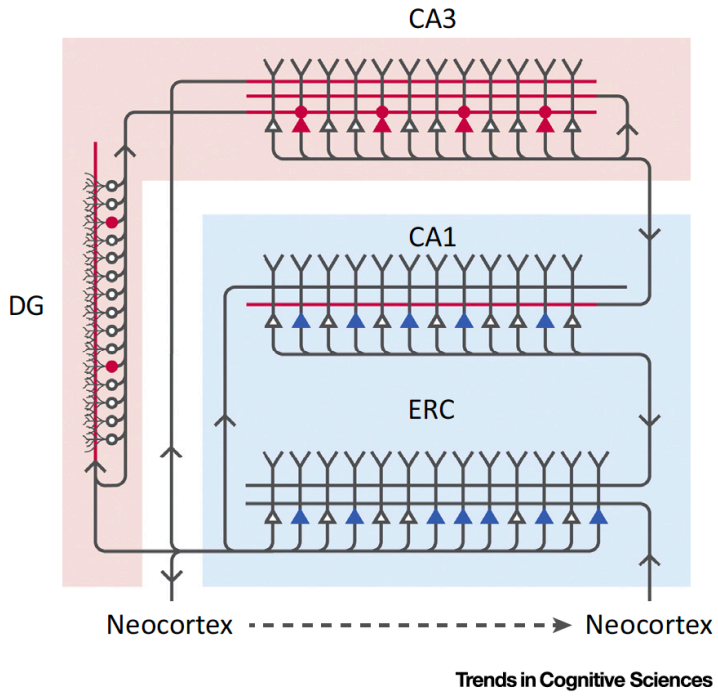


Figure 2.6: Trisynaptic and monosynaptic paths from entorhinal cortex (ERC) to hippocampus, and return. The trisynaptic circuit begins at the far left, from ERC through DG, and on to CA3. The monosynaptic path is depicted just to the right of that, directly from ERC to CA3. Reprinted from Trends in Cognitive Sciences, Vol 7, Kumaran, D., Hassabis, D., and McClelland, J. L., "What learning systems do intelligent agents need? complementary learning systems theory updated", Pages 512-534, Copyright (2016), with permission from Elsevier

2.7.5 Remapping

Despite the underlying repeatability of place cell activity, there are many conditions under which place cell firing varies. S. Leutgeb et al. (2005) introduced a distinction based on whether the setting involves changes to cues at a constant location, or changes to locations with constant cues. They found that rats tested in the same location with different cues (black vs. white walls, square vs. circular enclosure) showed the same place fields, but those fields fired at very different rates. They termed this *rate remapping*. When rats were tested in identical enclosures in two different rooms, two different sets of place fields were expressed, uncorrelated with each other. They termed this *global remapping*. This distinction had previously been obscured, in part because the differences in firing rates that can occur in rate remapping may make it appear that a field has disappeared, when it may be just firing at a very low rate. In the terms of Deshmukh (2014), rate remapping reflects changes in external sensory input processed by the lateral entorhinal cortex; global remapping reflects changes in the internal path integration information processed by medial entorhinal cortex.

In these terms, the findings from Knierim et al. (1995) show global remapping. Animals that were placed in a cylinder with a cue card that was rotated between sessions showed the same place fields across sessions, but the place fields rotated in roughly the same direction and angle as the cue card. However, animals that were consistently disoriented before being placed in the cylinder sometimes showed global remapping – they expressed new place fields between sessions. In the terms of the spatial attractor model, because their self-motion cues were randomly reset prior to each entry into the environment, they were unable to consistently retrieve the same chart to represent the environment, and therefore treated it as novel.

Many experiments involving running back and forth on a linear track show such different firing patterns that the common assumption had been that the two directions were being

treated as two different environments by the animal. Although typically some locations on the track appeared to have a place field in both directions, others appeared to have different place fields in one direction than another. Navratilova, Hoang, et al. (2012) studied the development of place fields over time in a linear track environment. They found that initial exploration of the track leads to a single set of fields, but repeated traversals strengthen the response of a given field in one direction and weaken it in another, so that the set of fields come to appear to be disjoint. This is characteristic of rate remapping; the animal views that the track is part of a single environment, but distinguishes cues and tasks demands in one direction from the other.

Fyhn, Hafting, Treves, Moser, and Moser (2007) studied whether the distinction between global and rate remapping in the hippocampus, has an analog in the grid cell populations in medial entorhinal cortex. The authors recorded from CA3 and MEC in rats under three conditions. In the same location, enclosure wall colors were reversed. As expected, place fields in CA3 showed rate remapping. Also in the same location, a square enclosure was replaced with a circular enclosure. Finally, similar enclosures were placed in two different rooms. Under both of these latter conditions, place fields in CA3 showed global remapping. During rate remapping, grid cell populations in MEC showed little change in grid spacing, orientation and spatial phase distribution, and the location of vertices was largely unchanged. Unlike the distinct changes in firing rates observed in CA3, firing rates in MEC remained largely unchanged. During the global remapping for different enclosures in the same room, the grid spacing, orientation and phase distribution stayed the same, but there was a spatial shift of 10.0 to 23.8 cm. The higher displacement was close to half the grid spacing. During the global remapping in different rooms, there was both a spatial shift and a significant rotation, ranging from 66° clockwise to 78° counterclockwise. In the terms of the spatial attractor model, we might interpret the authors' findings as the result of either maintaining the same head orientation (global remapping in the same location) or changing to a new head orientation (global remapping in different rooms). The authors also found that the

changes in CA3 and MEC happened at the same time. Rats trained under dark and light conditions in the same enclosure on alternate trials developed two different representations in both CA3 (different place fields) and MEC (displaced and rotated grids). During testing, shifting between light to dark to light conditions triggered switches between the pairs of representations. The switches happened simultaneously, in both CA3 and MEC, as would be predicted by the spatial attractor model.

2.7.6 Location of the spatial attractor dynamics

The above results suggest that attractor dynamics are affecting both place cells in hippocampus and grid cells in medial entorhinal cortex. As noted previously and in Appendix B, both CA3 and MEC have structures that could implement such dynamics.

Multiple findings are consistent with the hypothesis that global remapping is triggered by the recruitment of new populations of grid cells in MEC, which immediately recruit a new population of place cells in CA3 through the direct MEC layer two projections to CA3. When the current active chart in MEC stays stable through changes, the CA3 place cell population stays stable. Minor changes conveyed through DG result in orthogonal representations being projected to CA3, driving significant rate remapping and updating of the pattern stored in the CA3 autoassociative network. The replacement of one active set of cells in CA3 with an orthogonal set of cells shifts the autoassociative network to a different stable state. Driven by a shift to a different chart in MEC, this update is a different mechanism from autoassociative attractor dynamics.

In summary, spatial attractor models account for a broad array of findings, but are not without objections (see Appendix B). In a review of the state of models of path integration, Navratilova and McNaughton (2014) outline the strengths and outstanding problems of attractor models of path integration, concluding that they provide explanations for a variety

of phenomena, but by themselves cannot address all the complexities of the experimental findings. The authors contrast the attractor models to models that account for the pattern of grid cell firing as an effect of the interference of the oscillations of LFPs at different frequencies. Each set of theories accounts for some evidence not accounted for by the other, but the authors suggest that the balance of the evidence supports the attractor models. Given their usefulness in accounting for a large number of findings, for purposes of the SMS computational model, I will adopt a simplified form of spatial attractor model.

2.7.7 CA1 information processing

CA1 and CA3 are the two major hippocampal areas that express place fields. Both areas receive inputs from the entorhinal cortex, although the details of this connectivity differ. Perhaps the most important difference between them is that CA1 does not have the recurrent connections that are present in CA3. Instead, CA1 gets much of its input from CA3, and in turn passes information on to the subiculum and to the deep layers of EC, and thence to much of the neocortex.

S. Leutgeb, Leutgeb, Moser, and Moser (2006) studied differences between CA3 and CA1, both in the speed with which rate remapping develops, and in the magnitude of rate remapping. Rats were placed in the same box, but the wall colors were varied between black and white across trials. As expected, stable place fields were expressed under the two conditions, but there was rate remapping when the wall colors changed. They found that rate remapping happens rapidly in CA3, and the magnitude of the change in rates can be very great, perhaps an order of magnitude. This is consistent with the model of CA3 as an autoassociative network, where differences are expressed as orthogonalized population vectors. Rate remapping developed more slowly in CA1, sometimes requiring trials over multiple days, and the magnitude of the rate changes was smaller. The authors offered a functional interpretation

of these differences, in support of the function of the hippocampus in storing episodic memories. Both CA3 and CA1 express place fields immediately upon entry to novel environments, reliably coding "where" events take place. In responding immediately to nonspatial cues, CA3 encodes "what" happens at a given place. But the rate changes of CA3 place fields may result in some conditions being represented by very low firing rates. (As noted earlier, this may be the source of findings in other studies of place fields disappearing; the fields might still be expressed, but at rates so low they fall below the operational threshold that defined an active place field.) Very low firing rates may make it more difficult to accurately store sequences of place fields in the hippocampus. Because rate changes in CA1 are smaller, firing rates are more consistent, which might enable CA1 to more reliably store "when" events happen in relation to one another.

Complementary learning systems theory has been updated to account for some of the many findings since it was originally proposed (Kumaran, Hassabis, & McClelland, 2016). As depicted in Figure 2.6, a basic difference between CA3 and CA1 is that with its recurrent connections and inputs from dentate gyrus, CA3 responds rapidly to changes in its inputs, making it well-suited for "one-shot" learning. CA1, on the other hand, receives outputs from CA3 along a pathway that functions like most inter-area connections in the brain, subject to Hebbian learning at a relatively slow pace (although, see Ketz, Morkonda, and O'Reilly (2013) for evidence of faster and more accurate learning in CA1 over classical Hebbian learning). As CA1 is the source of hippocampal output to the entorhinal cortex and the rest of the brain, this might have the function of keeping the representation of a place relatively stable over time. Stability of that representation would be important under memory indexing theory, to enable other cortical areas to retrieve the correct associations to a given place.

At the current level of modeling in SMS, these suggested differences of function between CA3 and CA1 are not needed, so SMS does not model CA1. (In Appendix D, incorporation of some model of CA1 functioning is briefly described as a possible future enhancement.)

2.8 Trajectories

Following the usage of Pfeiffer and Foster (2013), I will term activation of a sequence of adjacent place fields a *trajectory*. The firing of cells from multiple place fields during each cycle of the theta oscillation strengthens a portion of the trajectory. It also has the effect of temporally compressing events that originally occurred at a slower time scale. The process of initially establishing a sequence of place fields, e.g., ABCD, generally occurs over the time scale of the animal's movement, on the order of 10^0 seconds. Strengthening the relationship between place fields ABCD during the theta cycle happens over a time scale on the order of 10^{-1} seconds. This suggests that at some level, what is important is that B happened after A, and that the elapsed time between the two events is less important. In this section, and in additional detail in Appendix B, I will review a number of findings regarding trajectories, happening under different behavioral conditions and at various time scales, beginning with findings at the time scale of the animal's movement through the environment.

Frank, Brown, and Wilson (2000) found that trajectories encode more than just the spatial locations through which an animal has passed. Recording from both CA1 and entorhinal cortex while rats performed a spatial alternation task in a W-shaped maze, they found rate remapping, related to aspects of the task. The firing rates of some place fields on the central stem showed "retrospective" or "prospective" coding. Some firing rates correlated with whether the animal had come from the left or right arm of the maze (retrospective coding). Other firing rates correlated with the animal's next choice, of either the left or right arm (prospective coding). In a similar experiment, Wood, Dudchenko, Robitsek, and Eichenbaum (2000) also found that the firing rates of a majority of place fields of rats on the stem of a T-maze during an alternation task differed significantly during left and right trials. Both studies found this effect to be independent of running speed and head direction. Ferbinteanu and Shapiro (2003) confirmed task-related firing of place cells, using an alternation task on a plus-maze to better separate retrospective from prospective coding.

Animals with lesions of the fornix (eliminating communication between the hippocampus and subcortical structures) performed at chance levels, confirming that the task does require a functioning hippocampus. These studies establish that trajectories do not simply represent sequences of spatial locations, but may also represent the context of the task, and future actions.

In an early finding of temporally compressed trajectories that appear to reflect future activity rather than replay of past activity, Johnson and Redish (2007) recorded units from CA3 in rats performing tasks in T-mazes. At choice points in the maze, they found that the place field populations frequently represented locations other than the one the animal currently occupied. At a high level of significance, the locations represented were in the forward direction, rather than in the direction behind the animal. The trajectories were specific; they frequently moved progressively down one arm of the maze at a time, and then switched to the other arm. The authors noted that at choice points the animals engaged in behavior known as *vicarious trial and error* (VTE) – small head movements alternating between the choice directions. This VTE behavior tended to be associated with higher levels of non-local representations. Local field potentials at the choice points showed theta oscillations, and gamma oscillations (30-80 hz). Oscillations in the SWR range (100-250hz) were absent at the choice points, but frequently occurred while the animal was consuming a reward.

We would expect that one of the most important things to remember about a place in an environment is whether it has been rewarding or not. Singer and Frank (2009) studied the impact on SWRs of the presence or absence of a reward. Recording from CA3 in rats on a multiple arm maze, they trained rats on two spatial alternation tasks, comprised of sequences of arm traversals. If a sequence was executed correctly, there was a reward at the end of each arm in the sequence. There was considerable overlap in the arms of the two sequences, and during the test sessions, the sequence that was currently correct shifted back and forth at unpredictable intervals. The rats were able to detect when a sequence was no longer

rewarding, and shift to the other sequence. This design created a situation where the only difference in traversing a given arm at two different times was the presence or absence of a reward, enabling an analysis of the SWRs that occurred at the well, where reward was or was not present. At the rewarding wells, the authors found increases in the time spent at the well, the number of SWRs, the average number of spikes fired per neuron, the mean firing rate, and the proportion of cells that were active. Place cells with fields on the track were also represented more frequently on rewarded than unrewarded trials. Although some of these measures are clearly correlated, analysis showed the independent increases in each to be significant as well. When an arm was rewarding during learning of a new sequence, the rate of SWRs and the probability of cell activation were higher than on rewarding trials for familiar sequences. Finally, they noted that place cells from the previous path in the sequence and the next path in the sequence were both represented more in rewarding trials than unrewarding trials; this suggests that the SWRs might have included both replay and preplay sequences. Taken together, and in the context of the theory of hippocampal replay to update the cortex, these findings suggest that rewarding spatial sequences are replayed at a high rate, increasing the probability of effective learning of an adaptive response by the animal.

In summary, trajectories differ along multiple dimensions. They happen at different time scales, under different behavioral conditions, represent paths in forward or reverse directions, and appear to replay past paths or preplay future actions. Trajectories also contain information about the context and outcomes of the current task. For this study, I am interested in preplay trajectories during quiet waking SWRs, as well as preplay trajectories during active movement, especially during periods of vicarious trial and error. In the next section I contrast two hypotheses that attempt to account for these preplay trajectories.

2.9 Two hypotheses regarding preplay trajectories

In this section I discuss two possible hypotheses regarding preplay trajectories during awake SWRs and during periods of theta accompanying vicarious trial and error in active navigation.

1. Hippocampal hypothesis: preplay trajectories are generated as a result of internal processes within the hippocampus.

2. Simulation hypothesis: preplay trajectories are generated as part of a process of simulated motion.

These are not the only possible hypotheses that might be proposed; the precise role that trajectories in the hippocampus play in supporting spatial navigation is a subject of active discussion (D. J. Foster & Knierim, 2012). Having observed many different types of trajectory in the hippocampus, a beginning question is whether some or all trajectories originate in the hippocampus, or elsewhere. My hypothesis of simulated action suggests that the preplay trajectories are part of a process that starts elsewhere. Clearly a simpler alternative is to suggest that the preplay trajectories are generated by internal processes within the hippocampus. That alternative remains plausible for replay trajectories in REM and SWS sleep, given their hypothesized role in driving learning in the neocortex. It is parsimonious to suggest that similar mechanisms might generate preplay trajectories, although the function of preplay trajectories is presumably different.

The discussion begins by summarizing Pfeiffer and Foster (2013) (see Appendix B for a detailed review). Pfeiffer and Foster (2013) recorded from CA1 of well-trained rats in a familiar square arena, containing wells that could be refilled from below with a reward. Each of approximately 30 daily trials consisted of two phases. In the first phase, the rat started at a well, designated the Home location, and foraged for a reward randomly placed in some other

well, designated the Random location. Finding the Random location and consuming the reward completed the first phase. In the second phase, which began automatically without a signal, the Home location was filled with a reward. Finding and consuming the reward at the Home location completed the second phase and the trial. The Random location varied from trial to trial. The Home location was where the rat was placed on the first trial of the day, and varied from day to day.

The routes taken to Home wells were shorter and faster than routes to the Random wells, suggesting the rats remembered where the Home well was, but did not know where the Random well was for each trial. Trajectories were detected during SWRs that occurred while the rats were consuming rewards. The trajectories occurred at similar levels when the animals were at the Home well ("home trajectories") and at the Random well ("away trajectories"). Away trajectories ended at the Home location much more frequently than at any other location. Regardless of where the animal was, the endpoint of the trajectory much more closely predicted the animal's future physical path than it did the previous physical path.

Because the rats are well-trained, they are familiar with the arena, and we can assume that approximately all possible physical trajectories through the arena have been taken at one point or another. We can also assume that these trajectories have been strengthened through various forms of replay. The design does ensure that some trajectories are novel in at least one sense. For the first 19 trials on a given day, the trajectory from a given away location back to the home location, has never been previously rewarding.

Under the hippocampal and simulation hypotheses, how might we account for the over-representation of away trajectories that end at the home location? As noted earlier, there is no topographic relationship between physical cell locations in the hippocampus and spatial locations in the physical environment, so it is not possible to map a novel trajectory between two place fields in the hippocampus. However, under the hippocampal hypothesis,

a previously established trajectory could be traversed. The fact that the home location is over-represented as the end point of away trajectories suggests that somewhere in the brain the animal "knows" that home is potentially rewarding. The authors looked for, but did not find, significant firing of the place cells representing the home location prior to the trajectory events (Pfeiffer, 2015). Although the place cells representing home are not firing, perhaps they have been activated at some level below threshold, but higher than the background activation level. A mechanism of gradient descent towards more activated locations might then lead to the selection of trajectories toward home at levels significantly above chance. Upon reaching home, the gradient would be zero, and the trajectory would stop.

Under the simulation process, I propose that some process generates candidate orientations to be explored, and passes them to a motor program, which executes while suppressing activation of motor neurons. I assume that the execution of the motor program generates self-motion signals, (*efference copy*), driving activity in the path integration system. Updates to the animal's simulated position in MEC in turn drive activation of the corresponding place cells in the hippocampus, generating the observed trajectory.

The lack of representation of the home location prior to the trajectory also poses a constraint on the simulation hypothesis. If the animal is guided by a vector between the current location and the target location (McNaughton, Knierim, & Wilson, 1995), we might expect the target location to be activated as part of calculation of the path to be followed, prior to simulated movement along that path. Perhaps the simulated movement is guided only by an orientation, rather than by a vector to a target location. The choice of orientation might be influenced by a similar process as proposed for the hippocampal hypothesis; perhaps the home location has been activated at some level below threshold. In addition to the possibility of detecting this gradient in the hippocampus, perhaps it is also plausible to detect a gradient in MEC. If neurologically plausible, this seems as if it would simplify the process of reading the target orientation, as MEC provides a reliable spatial metric against which to estimate.

Evidence that differentiates between the two hypotheses comes from studies of the emergence of place fields in trajectories that represent locations prior to physical exploration of those locations. Ólafsdóttir et al. (2015) placed rats in a T-maze, where the animal was initially confined to the stem by a barrier. The arms of the T had never been explored, but were visible through the barrier. Randomly, one of the arms of the maze was cued as a goal by placing food at the end of the arm, such that the animal could see the food as it was placed. After the goal was cued, the animal was allowed to rest, after which the barrier was removed and the animal was allowed to explore both arms of the maze. Trajectories observed during SWRs in the rest period included place fields that were later found on the arms of the maze, even though the animal had no previous experience on those arms. The arm that had been cued as the goal was represented in these trajectories at a much greater rate that the non-goal arm, suggesting that its rewarding character had motivated the preplay of trajectories during the rest period, presumably as a planning mechanism. Trajectories were observed both to and from the goal arm, and in forward and reverse directions.

These findings are not compatible with the hypothesis that novel trajectories are generated by gradient descent of previously established place field sequences in CA3, simply because there has been no previous opportunity to express place fields for the novel locations. The authors suggest that perhaps pre-configured but previously unused cell assemblies in CA3 are activated during the SWR trajectories to represent the as-yet unvisited locations. By virtue of strengthened connections during these SWR trajectories, these same assemblies are re-used during physical exploration later. Although this may be plausible under some models of hippocampal function, this suggestion is not consistent with the spatial attractor model discussed above, as the allocation of place cells to place fields is not endogenous to CA3. Rather, movement across the currently active chart in MEC is hypothesized to activate connected cell assemblies in CA3, generating new place fields.

If we accept the spatial attractor model discussed earlier, the study findings are not compatible with the hypothesis that the SWR trajectories originate in CA3. Accepting the spatial attractor model, the simulated action hypothesis, on the other hand, does account for the allocation of new place fields in trajectories prior to physical experience. Having seen food placed in an as yet unexplored location, the animal might estimate the direction of the food in relation to its location on the current active chart in MEC. During the rest period, the animal might simulate moving in the direction of the remembered food, generating a sequence of updates in MEC, and a corresponding sequence in CA3, some of which involves previously unallocated place fields. These place fields are associated with their corresponding chart locations, and are then re-activated during later physical exploration. The equal proportions of trajectories in the reverse direction from the goal might be due to a credit assignment mechanism (D. J. Foster & Wilson, 2006) to update the cortex with the learning from the simulated motion that the target direction is likely to be rewarding.

Simulated movement on the current chart to explore locations on a visible path seems a plausible mechanism, but the original reports of preplay in novel environments require a more general explanation to be compatible with the spatial attractor model. Dragoi and Tonegawa (2011) recorded from CA1 in mice during resting periods before their first encounter with two novel environments. In one case, a barrier made one arm of an L-shaped track inaccessible until the animal had become familiar with the remaining arm, at which point the barrier was removed. The trajectories were recorded during quiet waking periods while the rat was in the familiar arm, prior to barrier removal. In the second case, the animal was introduced to a single length of linear track directly from a resting box in the same room, without any prior exposure to the track. In both cases, the temporal sequence in which place fields fired during the trajectories was significantly correlated with the sequence in which the place fields were physically traversed once the animal was introduced to the new environment, in comparison to sequences that were the result of randomly shuffling the order of place cells in the novel

environments. For the L-shaped maze, the place fields in the familiar arm were stable before and after the removal of the barrier, indicating that global remapping had not occurred.

How might these findings be accounted for under the spatial attractor model? The study does not indicate that the animal could see through the barrier in the L-shaped maze. The sleep box for the second case had high walls so that the novel track to which the animal was later transferred was not visible (Figure 2.7). In the absence of visual cues about the locations that are to be later explored, what might be the mechanism for allocation of cell assemblies to locations? The authors suggest that the new cell assemblies might be generated by activation during SWRs of existing connections to cells that are silent in the familiar environment. These activations might then make it more probable that the same connections are activated when physical exploration occurs. Under the spatial attractor model, however, a different mechanism suggests itself. During quiet awake SWRs, perhaps the animal generates random trajectories uniformly across the current chart in MEC via simulated movement. Some of these trajectories would later be found to intersect with the place fields of the novel track environments, producing the observed correlations.

It is difficult to estimate the plausibility of this proposed account from the study data, because for the first test case, the analysis is confined to "spiking events" (highly correlated with SWRs), where at least four place fields are detected that match the place fields later found in the novel environment or in the familiar environment. There is no quantification of the number of SWRs that did not meet this criterion. Under the spatial attractor model, these excluded SWRs might have represented trajectories that randomly covered the rest of the environment represented by the current chart. For the second case, it is not clear whether the definition of spiking events is limited to those that include a place field in the novel track, or whether it includes any event where multiple place cells were activated in a short period preceded and followed by silence. Approximately 16% of the spiking events had a significant correlation with physical trajectories; one might speculate that at least some

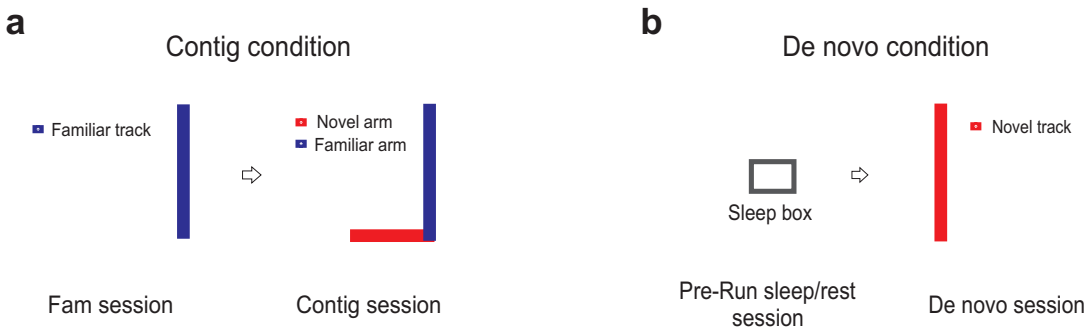


Figure S2 Experimental design a, Experimental apparatus for Contig condition. Left, a linear track (blue) to which the mice were familiarized. Right, an L-shaped track consisting of the familiar arm (blue) and a contiguous novel arm (red). After the mice were familiarized to the linear track, the barrier at one of the two ends was removed so that the mice could explore the entire L-shaped track freely. **b,** Experimental apparatus for the De novo condition. Left, sleep box in which mice were kept prior to being exposed to running on a linear track for the first time. Right, the novel track (red) on which mice had the first run session on a linear track (de novo run session).

Figure 2.7: Design of familiar and novel environments in Dragoi and Tonegawa (2011). Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Nature ("Preplay of future place cell sequences by hippocampal cellular assemblies", Dragoi, G., and Tonegawa, S.), (2011)

of the uncorrelated spiking events could be found to represent other spatial trajectories in the environment. The distance from the rest box to the linear track is not specified, so it is difficult to estimate the proportion of the entire environment that typical place fields overlapping the linear track might represent. Note that as seen in other experiments, e.g., Pfeiffer and Foster (2013), we would expect the trajectories to show a start- or end-point bias towards the animal's current location.

Some review is appropriate of the spatial attractor model, to better understand how I suppose it could accommodate the lack of visibility of portions of the environment. Upon entry to a novel environment, the spatial attractor model suggests that medial entorhinal cortex establishes a new chart by allocating random but stable grid coordinates and spacing. This initial allocation does not depend on external cues, just on the current state of the path integration system. Any movement over the grid, including simulated movement, causes the activation of directly connected place cells in CA3 and CA1, establishing place fields for the current grid coordinates. Simulated movement over the grid can take place immediately upon initial establishment, in the absence of sensory information about the places to be associated later with particular grid coordinates. The only requirement is that the environment be perceived as stable. If the animal were disoriented between the rest period and the exploration period, global remapping would occur, breaking any relationship between the place fields established by trajectories generated by simulated movement during rest, and the place fields established during physical exploration. Next, some salient cue is used to anchor the head direction system to the current coordinates. This cue is input from lateral entorhinal cortex, reflecting sensory input. Additional sensory cues define the qualities of place fields as they are encountered. I suppose that in this experiment a new chart is allocated upon entry into the laboratory, and the chart is anchored to some distal cue in the laboratory. Placement in the rest box enables the animal to do some limited exploration and associate the box walls with the place fields of the box. I further suppose that during quiet awake SWRs within the box the animal simulates moving through the larger environment of the laboratory outside the rest box, thereby establishing place fields for the simulated trajectories. Although the track is introduced later, it overlays portions of the environment over which simulated motion has already taken place. Place fields overlapping the track may already have been established. When introduced to the track, the animal explores its local features, probably leading to significant rate remapping of the pre-established place fields, but the grid coordinates of the place fields should be unchanged.

An experiment to test the plausibility of the spatial attractor model and simulated action hypothesis account would be to place a rest box in the center of a larger open arena, with which the animal has had no previous experience. On the expectation of random exploration of the current chart through awake SWR trajectories during the rest period, I would expect those trajectories to overlap significantly with the place fields later expressed during exploration of the open arena. Absent features of the laboratory that appear especially salient, I would expect that the spatial distribution of the trajectories would be roughly uniform, covering much of the arena prior to its physical exploration.

A somewhat analogous approach was taken in a later study by the same authors (Dragoi & Tonegawa, 2013). In this study, rats rested initially in a sleep box, and were later moved to novel environments, first, a linear track, and in a later variation, three tracks configured in a U-shape. Preplay was found in each case; in particular, preplay showed an independent set of place fields for each side of the U-shape. In this study, spiking events were defined as epochs of activity from at six pyramidal cells with < 100ms interspike intervals, preceded and followed by at least 100ms of silence. Under this definition, from 6-7% of the spiking events were significantly correlated with one of each of the tracks but not the other two, which is broadly consistent with the hypothesis that the animal might simulate random movement through the environment, in the absence of any additional information, thereby generating a uniformly distributed set of trajectories prior to physical exploration. Extrapolating more or less linearly, the authors suggest that approximately 15 tracks might be independently represented by trajectories prior to environment exploration. The authors suggest that the absence of visual information about the novel tracks during the rest period rules out the possibility of "mental travel" (their terms) as a mechanism to generate these trajectories, but the explanation given above suggests that their interpretation of the spatial attractor model is too limited.

Another basic assumption of the simulation hypothesis is that most of the circuitry that supports physical movement will be involved in simulated movement. Brandon, Bogaard, Andrews, and Hasselmo (2012) found that head direction cells in the postsubiculum do not show replay in SWS or REM sleep, or during SWRs in either SWS or quiet active periods. This is unexpected under the simulation hypothesis. In a later study in mice, however, Peyrache, Lacroix, Petersen, and Buzsaki (2015) found evidence of head direction activation during SWRs. They recorded from two areas known to have head direction cells, postsubiculum (PoS) and anterodorsal thalamic nucleus (ADn). Their results supported their main contention, that head direction cells function as a system and maintain a coherent state even during times of little sensory input, e.g., during SWS and REM sleep. They also detected changes in head direction during SWS, at an approximately tenfold level of temporal compression, roughly consistent with other findings of temporal compression during SWS. These results are consistent with the simulation hypothesis. The authors attributed the difference between their findings and those of Brandon et al. (2012) to smaller sample size in the earlier study, or to an overly wide (500 ms) window for Bayesian reconstruction.

2.10 Communication between regions

Having reviewed behavioral evidence that is consistent with the simulated motion hypothesis, it is time to review direct evidence on the communication between brain regions. The hypothesis of simulated motion supposes that at least some active SWRs do not originate in the hippocampus, but are triggered by processes in motor areas, in turn updating medial entorhinal cortex and then hippocampus. This presupposes a communication mechanism between these areas. Fries (2005) proposed that effective communication between cortical areas relies on *coherence* between groups of neurons (*assemblies*) in different areas. Coherence means that the groups are oscillating at the same frequency and are phase-locked.

This enables outputs from one neuronal group to arrive at the other group at a time when sensitivity to input is maximized. If we suppose that at least some SWRs are triggered outside hippocampus, the coherence hypothesis suggests that we should look for assemblies oscillating at the same frequency and phase at the first step upstream, in medial entorhinal cortex. Most of the early evidence contradicts this supposition.

The original reports of sharp wave ripples suggest that they appear to originate in CA3 (Buzsáki et al., 1983; Buzsáki, 1986). Later, in studies of SWR events in freely behaving rats during quiet waking periods, Chrobak and Buzsáki (1994, 1996) found that the SWRs propagate to the subiculum and parahippocampal region. The subiculum showed generally increased activity; the other structures showed increased activity only in the deep layers (layers five and six). As the deep layers of the parahippocampal formation, including the entorhinal cortex, are the output pathway from the hippocampus, this is consistent with the view that SWRs are a communication from the hippocampus to the rest of the cortex. In support of this, Olafsdottir, Carpenter, and Barry (2016) found direct evidence of coherent communication on the outbound path from CA1 to deep layers of MEC, the first step toward communication to the rest of the neocortex. Chrobak and Buzsáki (1994, 1996) found no evidence, however, that the SWRs in the deeper layers coincided with activity in the superficial layers (layers two and three), which are the input pathways to the hippocampus. The authors suggest a general pattern of responsiveness to theta oscillations in superficial layers and responsiveness to SWRs in deep layers. The absence of inputs from layers two and three during SWRs contradicts the simulation hypothesis.

There is some evidence that is consistent with the hypothesis that information from EC is triggering SWRs in CA3. Although approximately 95% of the time, Chrobak and Buzsáki (1996) observed ripples in EC simultaneously ($\pm$5ms) or with small delays (5-30 ms) from SWRs in CA1, perhaps 5% of the time entorhinal ripples preceded those in CA1.

In their study of *in vitro* activation of the subiculum in mice, Norimoto et al. (2013) found that a subset of neurons in the subiculum were consistently activated within 200 ms of SWRs in CA1. Measurement was done of LFPs in CA1, while spiking activity was measured in subicular neurons using functional multineuronal calcium imaging (fMCI) to capture transient calcium elevations that accompany an action potential. The distribution of firing of the neurons that tracked CA1 SWRs was bi-modal; most of the neurons followed the CA1 SWRs, but some preceded the CA1 SWRs. After severing the border between CA1 and the subiculum, the distribution became uni-modal; little firing followed the CA1 SWRs; most of it preceded the CA1 SWRs. The authors speculate that these activations are due to layer three input to the subiculum from entorhinal cortex. Alternative input sources could be the perirhinal and prefrontal cortices, although these are smaller in extent than EC inputs. Finally, there are subcortical inputs to subiculum; as these were not preserved in the slices that were studied, the authors discount them as possible sources. The study's findings imply that if EC can generate SWRs to the subiculum, it could also generate SWRs to the hippocampus.

Although the evidence is equivocal that there is communication from EC to CA3 in the high frequency range of SWRs, perhaps the communication is possible at another frequency range. Colgin et al. (2009) found two distinct frequency ranges that appear important in communication between EC and the hippocampus, and within the hippocampus. "Fast gamma", roughly 65-140 Hz, has a significant level of synchronization between MEC layer three and CA1. There is also some synchronization in these frequency ranges between CA3 and CA1, which the authors speculate may be due to synchronization between layers two and three in MEC, as well as outputs from layer two to CA3. "Slow gamma", roughly 25-50 Hz, shows a significant level of synchronization between CA3 and CA1. Slow and fast gamma occur during different phases of theta. Fast gamma occurs predominantly during theta troughs, when the probability of LTP in CA1 is maximal, suggesting that place-related firing in MEC may be updating memory in CA1. Slow gamma occurs predominantly during

the descending portion of the theta cycle, perhaps reflecting a memory retrieval process between CA3 and CA1.

There is some overlap between the gamma oscillation frequencies observed in Johnson and Redish (2007) and the low gamma frequencies (25-50 Hz) discussed earlier in the context of communication between EC and CA3. This overlap suggests a possible commonality between awake SWRs, and vicarious trial and error behavior during exploration under the theta oscillation. Perhaps both are forms of simulated motion and are driven by the same underlying process. This suggests a testable prediction, that the trajectory communicated between EC and CA3 during active SWRs and during vicarious trial and error is communicated at the same low gamma frequency.

Although this does not directly address communication between EC layer two and CA3, it directs our attention to gamma frequencies. In a followup to their 2013 study, Pfeiffer and Foster (2015) studied the timing of simulated movement within quiet waking SWRs recorded in CA1, both in an open arena and on a linear track. They found that the simulated movement was not uniform over time. Rather, there was an alternation between periods of no movement, followed by abrupt displacement, occurring at cycles of roughly 30 to 40 Hz. This coincided with increases in power in the slow gamma range of 25-50 Hz, which is also the frequency range of communication between CA3 and CA1 during periods of theta oscillation. The immobility and movement were phase-locked to the gamma oscillation, and the movement occurred at the phase of lowest activity. The authors suggested that this might be evidence of the autoassociative mechanism at work. Under their interpretation, the periods of high activity could constitute successive "sharpening" of the activity bump at the current location (attractor), while the periods of low activity might indicate a lowering of thresholds, enabling the transition from one attractor to another. Apart from its main argument supporting the autoassociative hypothesis of CA3 functioning, this study also suggests that the principal information in an awake SWR is being communicated at the low

gamma frequencies. Although the authors do not address this, we might speculate that the activity at the high frequencies of the ripples are an artifact of internal dynamics within the hippocampus, while the information as to the sequence of locations (attractors) being traversed is being communicated at low gamma frequencies. This in turn suggests that we might look for a communication at low gamma frequencies between EC layer two and CA3. That hypothesized communication might in turn trigger the sharp wave ripple within CA3 as the mechanism to carry the sequence information forward within the hippocampus and then back out to the cortex. I know of no evidence in support of this hypothesis, but it suggests a testable prediction, that recording from MEC layer two and CA3 simultaneously during awake SWRs should show signs of coherent communication between the areas, possibly at the SWR frequencies, or perhaps more likely, at low gamma frequencies.

The most direct evidence contradicting the simulated motion hypothesis comes from O'Neill et al. (2017). They studied rats navigating a T-maze, recording from electrodes in CA1 and in superficial MEC (sMEC) – layers two and three, which project to CA3 and CA1. Training the animals in an open arena and in a task in the maze enabled them to identify positional firing patterns for the units from which they were recording. They then analyzed High Synchrony Events (HSE), periods of significantly higher activity across multiple units, for CA1 and sMEC separately, looking for trajectories across the environment closely related in time. The study did find trajectories in both environments, but did not find much temporal relationship between them. Finding trajectories in sMEC is evidence that is compatible with the simulated motion hypothesis. But we would expect to see a temporal relationship between the trajectories in sMEC and the hippocampus, where trajectories in hippocampus closely trail those in sMEC. That is not what the study found. In Figure 2.8 we see examples of trajectories generated during HSE's in sMEC, which have no obvious relationship to the events in CA1 at the same time. An analysis of coherence between sMEC replay and CA1 activity, and between CA1 replay and sMEC activity showed coherence at chance levels, suggesting that the HSE's in the two areas were occurring independently.

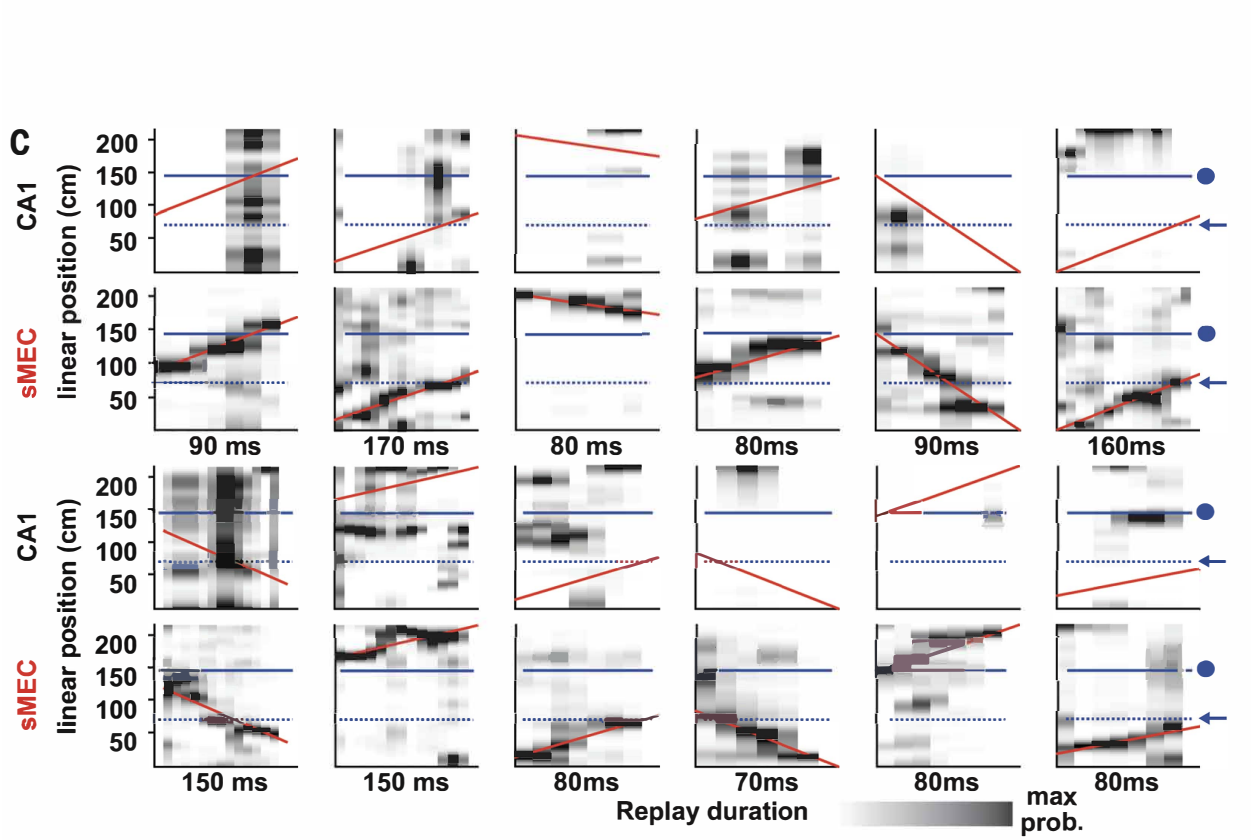


Figure 2.8: sMEC and CA1 position reconstruction during sMEC HSEs. From O'Neill, J., Boccara, C., Stella, F., Schonenberger, P., and Csicsvari, J. (2017). "Superficial layers of the medial entorhinal cortex replay independently of the hippocampus". Science, 355 (6321), 184-188. Reprinted with permission from AAAS.

Recent evidence that is consistent with the simulated motion hypothesis comes from Yamamoto and Tonegawa (2017), who studied mice running on a long track (6.3 meters), where the mice received a reward at both ends. Previous studies (Davidson et al., 2009) have found that replay for long tracks can take the form of multiple SWR events, representing successive sections of the track. In this study, the authors looked for events consisting of one, two, or three sequences in close temporal proximity. Recording from both CA1 and MEC layer three, which projects to CA1, the authors observed the mice in two behavioral states, quiet waking periods while on the track, and periods of slow-wave sleep in a rest box both before and after the sessions on the track. Under initial normal conditions, the authors observed alternating SWRs in CA1 and bursts of activity in MEC (Figure 2.9) during quiet waking periods, but not during sleep periods. During the quiet waking periods, the MEC activity preceded the CA1 activity, as would be expected of preplay, under the simulated motion hypothesis. During the rest periods, however, the alternation was not seen, which would be consistent with replay initiated from the hippocampus.

These behavioral differences were reinforced by the subsequent phases of the study, where the authors made two types of intervention into normal communications to CA1. An optogenetic intervention involved injection of a virus whose activity is triggered by light from an implanted fiber. When light through the fiber is turned on, the virus interferes with communication between MEC and hippocampus. The second intervention involved strains of transgenic mice where input from pyramidal cells in MEC to CA1 is inhibited. Both interventions interfered with the pattern noted during normal quiet waking periods, but not with the pattern noted during normal slow-wave sleep. For the optogenetic intervention, during quiet waking periods, the number of ripple bursts dropped significantly and the proportion of ripple bursts shifted toward a greater number of single bursts. The characteristics of ripple bursts during slow-wave sleep were unchanged. For the transgenic intervention, during quiet waking periods, many of the CA1 ripples were spatially fragmented across multiple regions of the track, and the spatial coverage of the track was smaller. The proportion of

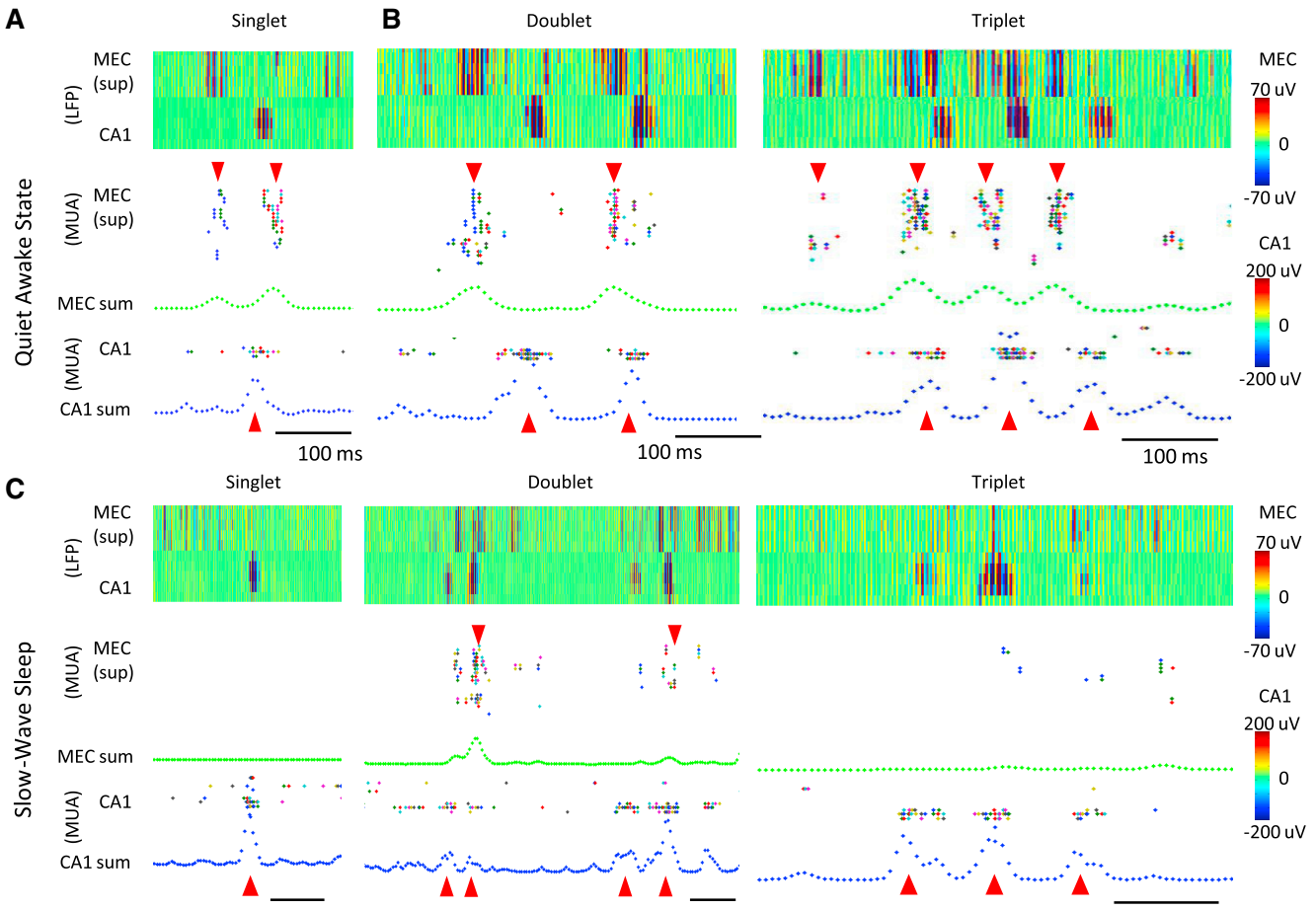


Figure 2. Quiet Awake-Specific Ripple Burst-Associated Alternating Burst Activities in MEC-CA1 Network

(A) An example of singlet sharp-wave ripple. Top panel: color-coded ripple band LFPs of superficial MEC and dorsal CA1 cell layer. Middle panel: associated superficial MEC MUA with elevated burst activities (red downward arrowheads). Green dotted trace is smoothed sum of MEC MUA (STAR Methods). Bottom panel: detected CA1 MUA and its smoothed sum (blue dotted traces) with elevated ripple activities (red upward arrowhead).

(B) Same as in (A): examples of doublets and a triplet, interregional burst interaction during quiet awake.

(C) Same as in (A): examples of non-interregional ripple bursts during slow-wave sleep.

Figure 2.9: Alternating MEC Layer 3 and CA1 activity. Reprinted from Neuron, Vol 96-1, Yamamoto, J., and Tonegawa, S., "Direct medial entorhinal cortex input to hippocampal CA1 is crucial for extended quiet awake replay", Pages 217-227, Copyright (2017), with permission from Elsevier

ripple bursts also shifted toward a greater number of single bursts. During slow-wave sleep, however, the spatial coverage and ripple burst proportion were unchanged.

As a comparison, the optogenetic technique was then used to block communication between CA3 and CA1. This intervention lowered the number of ripples in CA1 significantly both in quiet waking periods and in slow-wave sleep. Of the lower number of ripples, very few showed a replay sequence. This suggests that CA3 input is needed for normal ripple functioning, regardless of behavioral state.

Taken together, the Yamamoto and Tonegawa (2017) findings are consistent with the simulated motion hypothesis. As expected under simulated motion, during quiet waking periods activity is seen in MEC just prior to activity in CA1, including in multiple sequences covering much of the track. Also as expected, the interventions significantly interfere with this pattern. The pattern of activity during slow-wave sleep, on the other hand, is consistent with Complementary Learning System theory, where CA3-initiated replay propagates through CA1 and then elsewhere in the brain, without input from MEC.

Overall, the evidence on communications from MEC to hippocampus during quiet waking periods is equivocal.[7] In addition, Trimper et al. (2017) raise methodological concerns about recent studies that look for correlations between firing in hippocampus and MEC. Because grid cells fire in a repeating spatial pattern, the authors show that certain analytical methods can give rise to spurious correlations in firing across the regions.

Some form of communication from MEC to hippocampus during quiet waking periods is the central neuroscientific prediction of the simulated motion hypothesis. For purposes of this study, I make the simplest assumption, that the communication exists and is direct

[7]Note that both O'Neill et al. (2017) and Yamamoto and Tonegawa (2017) study communication from MEC to CA1, but the simulated motion hypothesis focuses on CA3, for reasons of model simplicity. Layers two and three of MEC project to CA3 and CA1, respectively, and SWRs cascade from CA3 to CA1, so findings regarding communication between MEC and CA1 make strong implications about communication between MEC and CA3.

and immediate. The disagreement in recent studies suggests that a revised account may turn out to be necessary. Such a revised account is unlikely to be addressable by SMS without modifications. The simplest model of an autoassociative network in CA3 uses a binary matrix, with values of zero or one. Such a simple model cannot accommodate likely complexities, such as the dynamics that generate SWRs, or the concept of sub-threshold activation, where input to neurons gives them an increased probability of firing without reaching the threshold to fire. The modifications that would be necessary to address such complexities are discussed in the last chapter, as part of future work, as well as in Appendix D.

Chapter 3

Petri nets and X-nets

3.1 Overview

Petri nets are a formalism that has been used to model and analyze a wide variety of concurrent processing systems. They enable the modeling of the impact of asynchronous events on ongoing processing. One application has been to model actions or behaviors, as parameterized motor routines, originally termed executing schemas (x-schemas) (Narayanan, 1997; Bailey, 1997); more recently termed *X-nets* (Narayanan, 2010).

The X-net model is useful in computational models of language (Doubleday, Trott, & Feldman, 2017), enabling a coherent integration of the asynchronous processing of language input and motor activity. In this work, X-nets are used to model the physical behavior of a rat in the arena. The interaction of that behavior with the animal's internal representation of that behavior in the hippocampal formation is the subject of the next chapter.

The X-net representation suggests that motor activities can be usefully modeled as a hierarchy of routines, where higher-level routines provide parameters for lower-level routines.

Events occurring at one level can be propagated to the other levels to change the course of processing. Interactions with components external to the X-nets (e.g., other cortical systems) can also be a source of parameters and events. This characterizes the use of X-nets in the current work, as part of the larger Simulated Motion System. The use of Petri nets has an additional advantage in separating some of the processing of the system into well-structured programs that can be analyzed and visualized in isolation, and in various combinations.

3.2 Petri nets

The Petri net formalism models concurrent processing as a set of *places*[1] and *transitions*, connected by directed *arcs*. An *inbound* arc connects a place to a transition; an *outbound* arc connects a transition to a place (see Figure 3.1). A place contains an integer number of *tokens*, zero or more. When a place has at least one token, it is *marked*; the vector of token counts associated with all the places in a Petri net is its state, or *marking*.

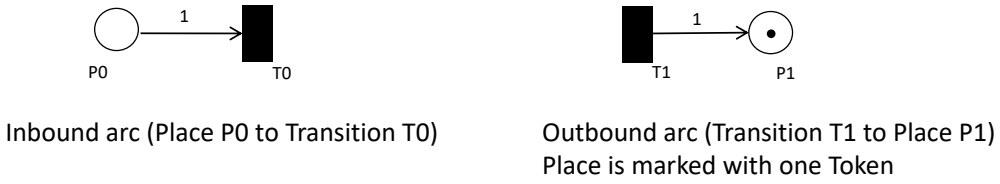


Figure 3.1: Inbound and outbound arcs

Arcs are of two types. A *normal* arc may be either inbound or outbound, and has an integer-valued *weight* of one or more, and consumes or produces tokens in the number of its weight. Alternatively, a weight may be determined by evaluating an *expression*. In this work, the only expression used is one that evaluates the number of tokens currently occupying some place in the Petri net, and sets the arc weight to that value. *Inhibitory* arcs are inbound

[1]This should not be confused with places in the sense of physical locations, as generally used elsewhere in this work.

arcs only, and prevent the firing of a transition when their associated places are marked (see Figures 3.2 and 3.3).

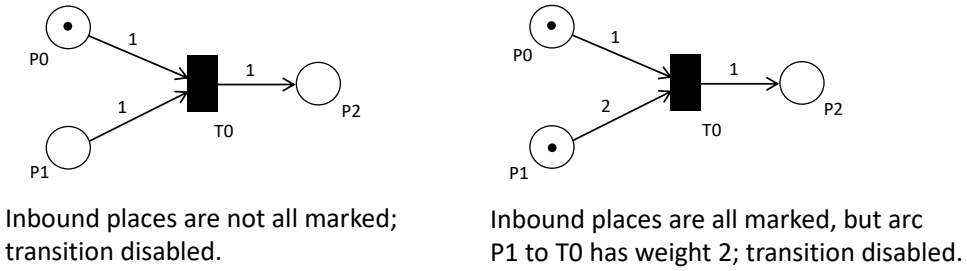


Figure 3.2: Disabled transitions (normal arcs)

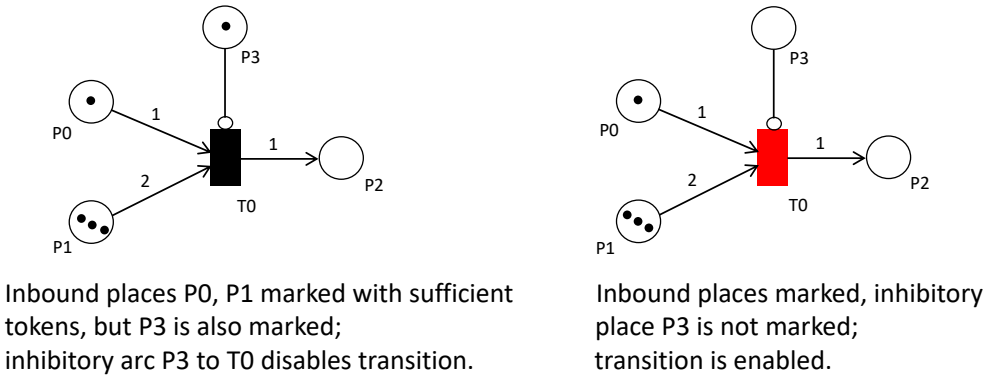


Figure 3.3: Inhibitory arc disables transition when its place is marked

A transition is enabled for firing when all the pre-conditions defined by its inbound arcs are met. Specifically, the pre-conditions are that the number of tokens in each inbound place for a transition is at least equal to the weight of its corresponding normal inbound arc, and no place associated with any inhibitory arc inbound to the transition, is marked.

Firing a transition consumes tokens in the amount of the weight of each inbound arc from its corresponding place, and then produces tokens in the amount of the weight of each outbound arc, in its corresponding place (see Figures 3.4 and 3.5).

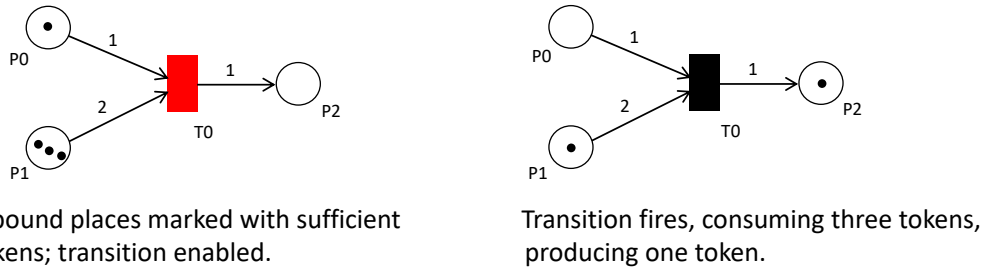


Figure 3.4: Enabled transition fires, consuming and producing tokens.

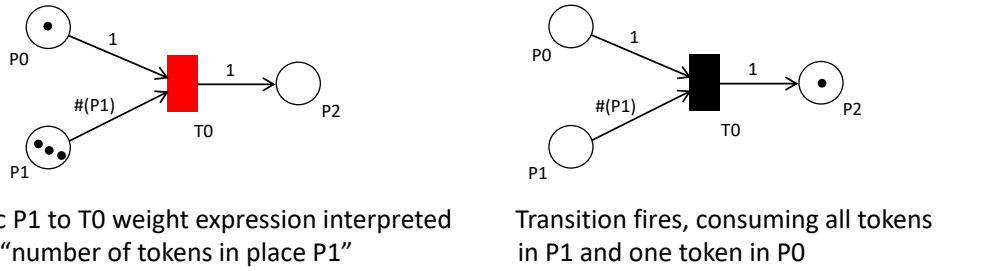
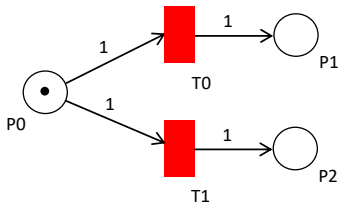


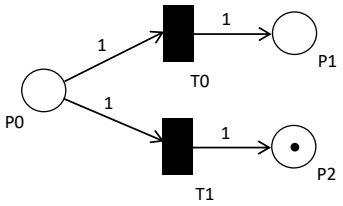
Figure 3.5: Variable arc weight expression consumes all tokens in referenced place.

Transition firing is stochastic; if multiple transitions are enabled, one is selected at random to fire (see Figure 3.6).

The basic Petri net formalism corresponds well to neural mechanisms. A Petri net place and its inputs and outputs could be interpreted as a single neuron, or perhaps more plausibly, as a population node. Places can accumulate tokens, similar to a node accumulating input activations. The firing of a node could be modeled as a transition firing. An arbitrary number of inputs to and outputs from a node's firing could be represented by arcs from and to places. The strength of a connection between two nodes could be modeled as the weights of the arcs between them and an intervening transition; these weights govern the number of tokens to be consumed by firing.



Both transitions are enabled.



Firing is random. Transition T1 fires.
Both transitions now disabled.

Figure 3.6: Among multiple enabled transitions, one is selected at random for firing.

There are areas in which Petri nets diverge from neural mechanisms. Unlike more realistic neural network models, inhibitory arcs are binary; their inputs are not summed. If the place attached to the arc is marked, the target transition is prevented from firing, regardless of the status of the excitatory inputs. Also, in the implementation used here, Petri net places accumulate integer token counts; continuous token quantities are not supported. Neither of these problems is a major concern for SMS; the primary use of Petri nets is as control mechanisms, implementing simple logic functions. For components where what is required is modeling of a neural network whose outputs are defined by the interaction of many weighted connections, Matlab networks will be used.

Although stochastic Petri nets can explicitly model time through temporal delays in the firing of transitions, the Petri nets used in SMS do not make use of this capability. Instead, transitions fire immediately, once their preconditions are met. This means that SMS does not attempt to model the complex temporal dynamics of neural systems, e,g., sub-threshold activations. Time is measured discretely in SMS, and the interaction between the components as modeled is not affected by sub-threshold dynamics.

The Petri net formalism is an abstraction of Markov processes. This mathematical foundation makes it possible to do various formal analyses of the behavior of a Petri net, e.g., to analyze whether a particular state of a Petri net is reachable. This work makes no use of

these methods, primarily because the behavior of the Petri nets in isolation is of little interest. The Petri nets are part of a larger system, and the integration with the external Matlab system removes many of the constraints that make analysis possible. The reachability of a given state is affected by the behavior of the Matlab portion of the system. It is not possible to determine from analysis of the Petri nets alone, whether and under what conditions a given state will be reached. Finally, although possibly of theoretical interest, the objectives of such forms of analysis are not relevant to the goals of this study.

For purposes of this work, perhaps the most relevant correspondence between Petri nets and neural mechanisms is the decentralized and asynchronous mode of Petri net execution. There is no centralized control of execution; transitions fire randomly when they are enabled and continue to do so until there are no more enabled transitions. As implemented here, Petri net execution is asynchronous from the processing that models other cortical subsystems. There is coordination at each time step between the Petri nets and the other subsystems, which is roughly analogous to the coordination across distance performed within the brain, mediated by distant areas firing at the same frequency. But within a time step, the course of firing of individual transitions is local to the Petri net.

3.3 X-nets for SMS

As suggested above, Petri nets serve as a useful intermediate-level abstraction for the representation of actions. The Petri nets used in SMS take the form of X-nets, which are designed to represent the various states of an action. These states have a standard semantics, originally inspired by linquistic *aspect* (Narayanan, 1997; Doubleday et al., 2017), which enables us to distinguish between actions that are ongoing, or just completed, or about to begin. The standard action semantics supports interrupting actions, suspending them, cancelling, etc., (see Figure 3.7). Narayanan (2010) suggests that in their modeling of the structure

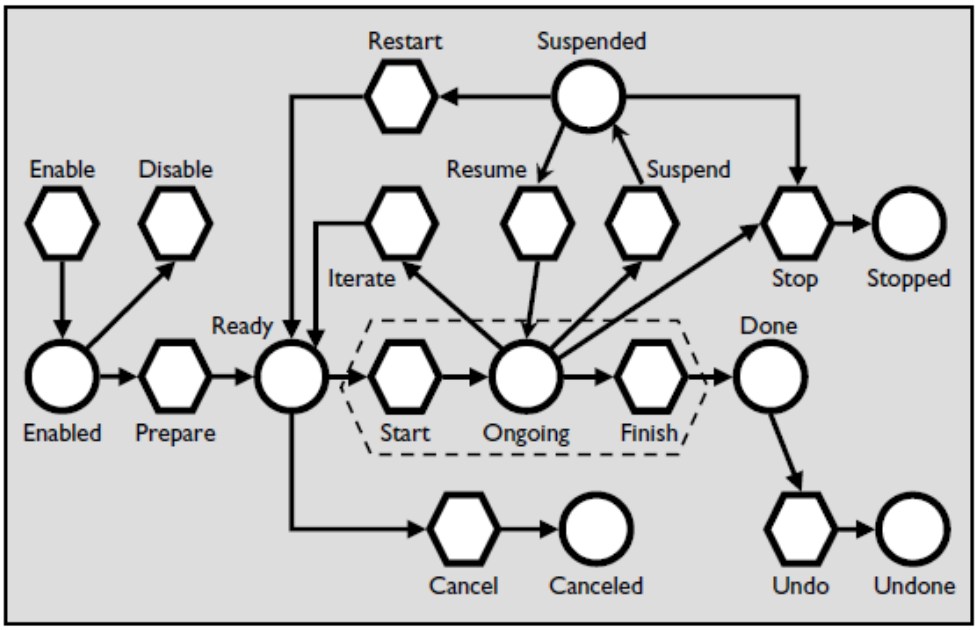


Figure 3.7: Event graph (controller) depicting the structure of actions. Hexagons may either be single transitions, or other Petri nets. From Narayanan (2010)

of actions in humans, X-nets might correspond to processing in the prefrontal cortex. For purposes of SMS, I do not attempt any analysis of which cortical structures are involved. In the computational model, I label the component that drives the simulated rat's motion "MortorCortex"; this is only meant to suggest "cortical motor processing", not any detailed interpretation of the actual components involved.

Within SMS I make use of a subset of the standard action semantics. Normally, actions progress through the states Enabled, Ready, Ongoing and Done, but actions may sometimes be interrupted, in which case they are Stopped. These will be the basic states of the behaviors that are modeled in SMS, *run*, *turn*, *move*, and *navigate*. Parameters are passed to the X-net

in the form of marking one or more places. The hierarchical structure of action is reflected in the hierarchy of these X-nets, where *navigate* invokes *move*, which in turn invokes either *run* or *turn*.

3.4 Petri net extensions to support composition and external systems

The Petri net and X-net portion of SMS is built on extensions written by the author to PIPE V5, an open source Java Petri net editor and debugging environment[2]. PIPE was extended to provide support for composition of multiple Petri nets, interfaces to and from external systems, and for execution by the external system through an API[3]. For reference, these are called the "hierarchical net extensions".

PIPE is composed of several components. The underlying logic that implements Petri net behavior is implemented in the PIPECore project. PIPE also has a graphical user interface (GUI) component, an environment for presenting and editing a visual representation of one or more Petri nets. The PIPE GUI relies on PIPECore to drive behavior of Petri nets when they are executing, e.g., when they are being debugged.

The hierarchical net extensions written for this work have been implemented in PIPECore, but no corresponding support has yet been added to the PIPE GUI to enable the visual manipulation of these extensions. To clarify the explanations of the various extensions, I have used some visual conventions in the figures, and these are re-used in the Petri net figures

[2] `https://github.com/sarahtattersall/PIPE`

[3] Current code is here: `https://github.com/sarahtattersall/PIPECore/tree/hierarchical-nets`, and is documented here: `https://github.com/sjdayday/PIPECore/wiki`. At some point, the extensions will be integrated as a new release of PIPE.

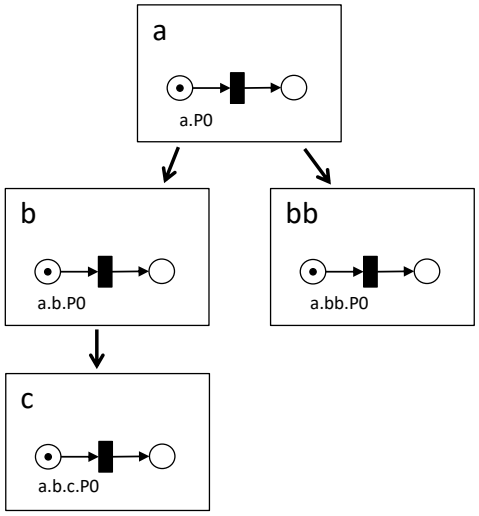
in Chapter 4, on the SMS computational model. These visual conventions have not yet been implemented in the PIPE GUI; that is future work.

The absence of visual editor support for the hierarchical net extensions means that it is cumbersome to maintain and debug the hierarchy of Petri nets used in the study, but it has no effect on the processing to generate the study results. The extensions were designed in part to enable Petri nets to be executed separately from the use of the PIPE GUI, which had previously always been required. This separation of logic from user interface is what makes possible the straightforward integration of Petri nets with external systems (Matlab, in the case of this study).

3.5 Hierarchical net extensions

3.5.1 Include hierarchy

Decomposing a problem into a hierarchy of smaller problems is a general computational approach. The hierarchical net extensions support the composition of large Petri nets from multiple small Petri nets. This enables each smaller Petri net to be programmed and debugged in isolation. The mechanism for composition of Petri nets is the *include hierarchy*, where a top-level Petri net may include one or more other Petri nets, each of which may in turn include other Petri nets. Each Petri net in the include hierarchy has a name; to avoid ambiguity, that name is qualified by the names of the Petri nets higher in the hierarchy. For example (see Figure 3.8), the top-level (or root) include might be named "a", and include two child Petri nets, "b" and "bb". Included Petri net "b" might have one child, "c". The parent of "b" and "bb" is "a"; the parent of "c" is "b". The fully-qualified name of "c" is "a.b.c"; the fully-qualified name of "bb" is "a.bb".



Include hierarchy: "a" is the top-level Petri net; "c" is the lowest level. The order of Petri nets at the same hierarchy level is not significant ("b" and "bb" may be included in either order). For uniqueness, the hierarchy of names is prefixed to the names of places and transitions.

Figure 3.8: Include hierarchy

3.5.2 Executable Petri net

The PIPE system has two modes of operation, one where Petri nets are edited, and a second where Petri nets are executed, one or more steps at a time. The addition of the include hierarchy concept makes this distinction more explicit. An individual Petri net is edited and persisted in its own file. This enables it to be tested and maintained independently[4]. When multiple Petri nets are added to an include hierarchy, however, they must be merged into one Petri net for execution. The combined Petri net is called an *executable Petri net*, and it supports execution and analysis. The executable Petri net contains the components from all of the Petri nets in the include hierarchy. Each component name is prefixed using the

[4]An include hierarchy is persisted in a separate XML file, naming each of the included Petri nets and identifying where their individual files are found.

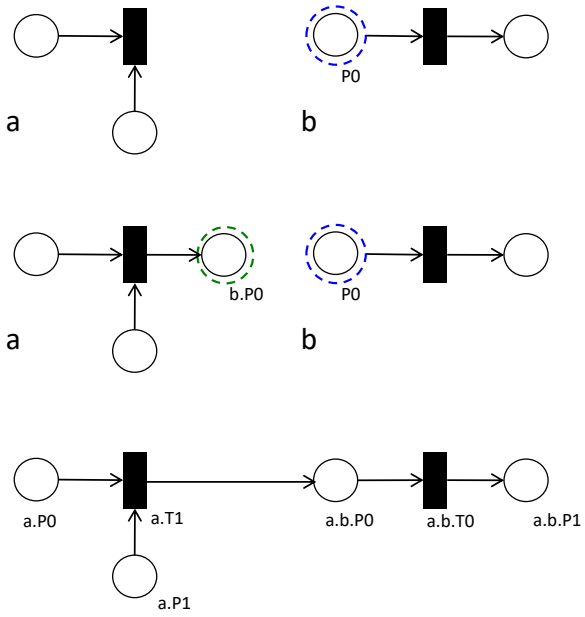
fully-qualified name of its Petri net in the include hierarchy. From the example above, place P0 in Petri net "c" would have name "a.b.c.P0" in the executable Petri net.

3.5.3 Merge places

The example include hierarchy above suggests that the Petri net at level "a" needs services provided by the "b" and "bb" Petri nets, and "b" in turns needs services provided by "c". In this composition of components, "a" marks one or more places in "b", which triggers processing in "b". This may result in marking other places in "b", which signal completion of processing in "b", and which may in turn trigger subsequent processing in "a". To encapsulate the function in "b" as much as possible, we define a subset of places in "b" as constituting the *interface* for "b". The typical semantics of the interface is that one or more of the places are entry points, whose marking starts processing in "b", and one or more of the places are exit points, whose marking returns values to the caller, "a". In Figure 3.9, place P0 in "b" is an entry point for processing from the perspective of "a". P0 is added to the interface of "b", and is designated a *home place*. This makes it available to be used by "a", as an *away place*. To this point, both "a" and "b" are separate Petri nets, and can be maintained and debugged separately. When an executable Petri net is generated from the include hierarchy, the home and away places are merged – the away place b.P0 drops out, and the arc that was connected to it is reconnected to the home place, a.b.P0. Home and away places are *merge places* – the mechanism by which Petri nets are separately maintained, and later merged for execution.

3.5.4 External system integration through the Runner API

Integration of Petri net execution with external systems was made possible by the creation of the *Runner* application programming interface (API). The Runner API enables other



Top: Petri net "a" includes "b". Place P0 is in the interface of "b", and may be accessed by "a". P0 in "b" is a home place.
Center: b.P0 is added to "a" as an away place.
Bottom: In the executable Petri net, away places drop out, and arcs are re-drawn to the corresponding home places.

Figure 3.9: Merge places

systems to request and control the execution of an executable Petri net. Runner is a Java API; the external system must be able to invoke Java programs[5][6]. The Runner API supports the following functions:

- load a Petri net from a file
- start Petri net execution
- mark a place in a Petri net
- subscribe to notifications of a change in marking of a place in a Petri net

[5]Extending the Runner API to be available across a network is another opportunity for future work

[6]Much of Matlab is written in Java, and there are mechanisms in Matlab to enable the execution of other Java programs. This made it straightforward to integrate Petri net processing with the other processing in SMS, through the Runner API.

– pass an object to be made available to an external transition when it fires

– subscribe to notifications of global Petri net execution events, including starting or stopping execution, and firing a transition

The primary implementation of the Runner API is the PetriNetRunner Java class. A PetriNetRunner is a single-threaded process, i.e., it runs one executable Petri net at a time. If execution of multiple Petri nets is required concurrently, one PetriNetRunner is required for each executable Petri net. The PetriNetRunner may be run asynchronously, in a separate thread. Various synchronization mechanisms are available, discussed below.

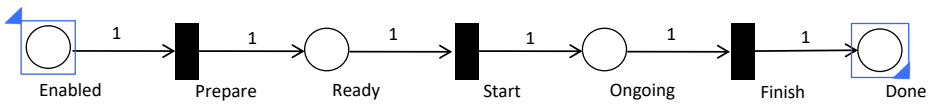
To initialize processing, the external system creates a PetriNetRunner and passes it the name of a file to be processed. The PetriNetRunner opens the Petri net or include hierarchy defined in the file, and creates an executable Petri net. The external system may then use the functions of the Runner API to execute and interact with the executable Petri net.

3.5.5 External interfaces

The concept of an interface for a Petri net also supports integration with an external system. Two mechanisms have been implemented to enable this integration. First, the external system can access *external places*. The external system can listen for changes in the marking of a given place, and can request an update to the marking of a place. Second, an *external transition* can be configured, which when fired, passes control to the external system for processing, prior to firing any further transitions. For some purposes, these two mechanisms are equivalent.

3.5.6 External places

An external place is one that has been added to the interface of its associated Petri net, and designated as available for external access. The external system may request that the marking of an *external input place* be rewritten with one or more tokens. The external system may also listen for updates to an *external output place*, such that the external system will be notified whenever the marking of the place changes (see Figures 3.10,3.11,3.12). This can be made more specific, such that notification is interpreted as boolean; when marked, the place is "true", and when unmarked, the place is "false". External places are the mechanism used by SMS to integrate the processing of the Matlab components with the Petri nets.

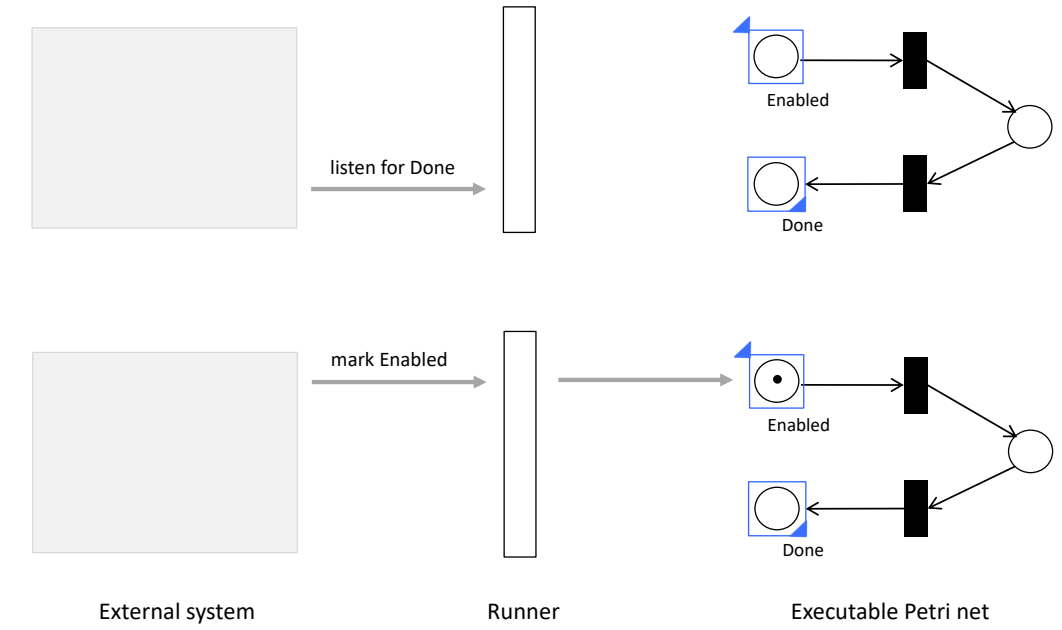


Standard X-net semantics, with externally accessible places. Enabled is an input-only place that can be marked by an external system. Done is an output-only place that can be read by an external system.

Figure 3.10: External places

3.5.7 External transitions

The other integration mechanism for external systems is an external transition, which provides a mechanism to execute Java code whenever the external transition fires, after token updates have been completed (see Figure 3.13). The external transition has access to the executable Petri net, and is optionally given a context passed from the external system through the Runner API, consisting of an instance of an arbitrary Java class. External transitions may also mark places in the executable Petri net, through a subset of the Runner API.



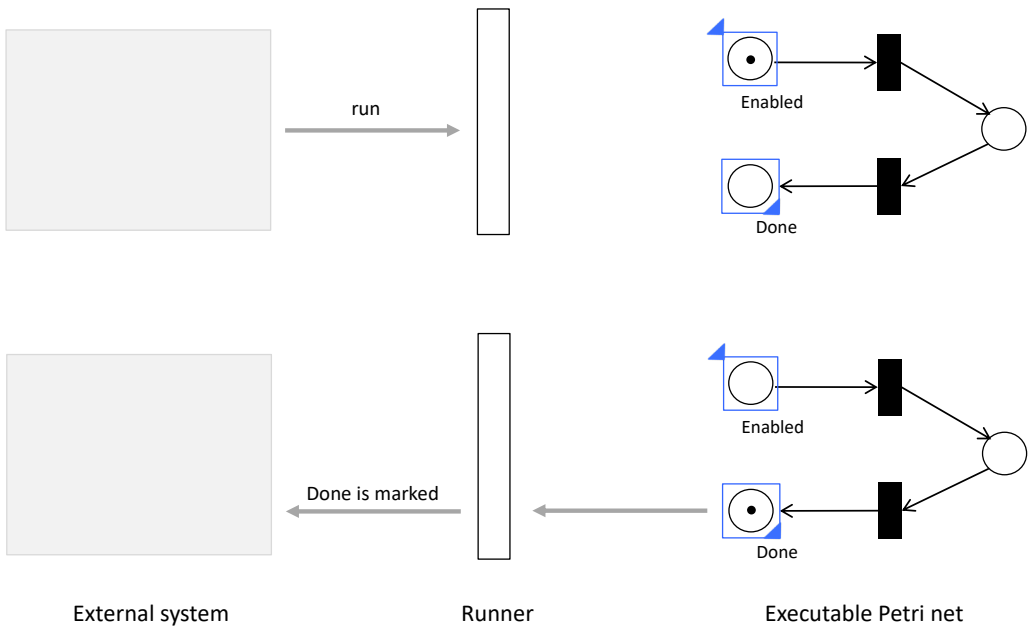
Top: external system requests to be notified of marking changes to external output place Done
Bottom: external system requests that external input place Enabled be marked

Figure 3.11: External system uses Runner to set up processing

External transitions are a useful way to share parameter values between the Petri net and the external system, as in Doubleday et al. (2017), but were not used for SMS.

3.5.8 Petri net persistence

The final extension to PIPE was in the area of Petri net persistence. Petri nets are saved in an XML format called PNML (Petri Net Markup Language). PIPE-specific extensions were made to PNML to support reading and writing include hierarchies, and external places and transitions.

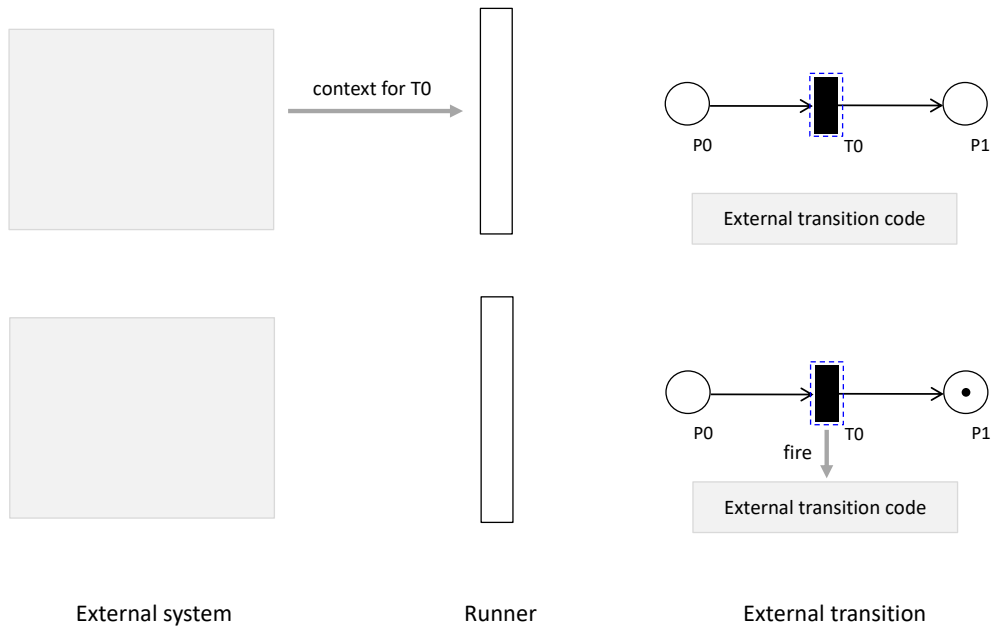


Top: external system requests Runner begin executing the Petri net
Bottom: Done is marked; Runner notifies external system

Figure 3.12: Execution begins, and external system is notified when external output places are marked.

3.6 Synchronization of Petri net processing with the external system

The integration of Petri net processing with an external system can be synchronized at several levels. The simplest approach is for the main processing thread of the external system to instantiate a PetriNetRunner whenever Petri net execution is needed, and run the Petri net until it exits, i.e., until there are no more transitions that are enabled for firing. This amounts to sequential processing, where control passes from the external system to the PetriNetRunner, and then back to the external system when, and only when, the PetriNetRunner exits. There might be simple cases where this is useful, but it is not an



Top: T0 is an external transition, associated with code provided by the external system. The external system may provide a context to be passed to T0 at time of firing.
Bottom: the external transition code receives control during the firing of T0, after token processing, with access to an optional context and the executable Petri net.

Figure 3.13: External transition

appropriate model for SMS, where physical behavior and an internal representation both need to be updated concurrently.

The second synchronization approach is to run two threads concurrently, one for the external system, and one for the PetriNetRunner. For the purposes of SMS, this better models the interaction of physical behavior and the internal representation. During an initial phase of testing of the integration of the SMS system, behavior consisted of many individual Petri nets being run, while internal representations of the results of each execution were simultaneously being calculated. Although this was an improvement, the overall system behavior was not predictable at a detailed level, due to race conditions between the two threads.

The necessary synchronization between the two threads was achieved by implementing an explicit acknowledgement between the PetriNetRunner and various listeners in the Matlab system. So, for example, if the external system is listening for place Done to be marked, execution will not proceed in the executable Petri net until the Done listener sends an acknowledgement to the PetriNetRunner. This enables the external system, in this case, to ensure that the internal representation is updated appropriately following completion of each unit of the Petri net processing that models the corresponding physical behavior. The external system does not listen for every event in the Petri net, just those that are important for coordination with the external system. In between such events, the precise order of transition firing within the Petri net is not significant. This level of synchronization can be interpreted as roughly modeling synchronization in the brain, where detailed processing happens locally within a subsystem, with high-level coordination between physically distant subsystems being mediated by coherent firing at predictable frequencies.

3.7 Petri net summary

This completes the description of the extensions to PIPE, which make possible the integration of Petri nets with an arbitrary external system. In the next chapter I will describe SMS, including how Petri nets that model spatial navigation are integrated with the rest of the system.

Chapter 4

Computational Model

4.1 Overview

In this chapter I review a computational model of spatial navigation in rats during physical and simulated motion, termed the Simulated Motion System (SMS). SMS models the physical motion of an animal in a rectangular arena, and the internal representation of this motion in various areas of the hippocampal formation, principally the entorhinal cortex and hippocampus. Periodically, the animal rests, entering a quiet waking state, during which it simulates motion in the arena, starting from its current physical position.

The model is composed of a number of components, each of which is relatively simple. The combination of components, however, exhibits behavior which is complex, and which demonstrates some of the behavior consistent with the neuroscientific predictions of the simulation hypothesis. Specifically, as the simulated animal moves, a composite component, the HippocampalFormation, coordinates the interaction of several sub-components. These sub-components model portions of the entorhinal cortex (EC) and hippocampus. The components that model the EC maintain an internal representation of the animal's head

direction (HeadDirectionSystem), two-dimensional position in space (GridChartNetworks), and relationship to salient visual cues (LecSystem). This set of representations are used to create inputs to the PlaceSystem, which models the hippocampus. In the PlaceSystem, each location in the arena is associated with a vector, a Place ID, which is then communicated back to the HippocampalFormation, and associated with the current state of the originating components (HeadDirectionSystem, GridChartNetworks, and LecSystem).

This dynamic works the same way whether the animal's motion is physical or simulated. The effect is that the animal "explores" the arena through a combination of physical and simulated motion. At each location, the animal attempts to retrieve a place ID from the place system. If no place ID is retrieved, the location is novel, otherwise it is recognized. After simulated motion during a quiet waking period, the animal attempts to physically retrace the first run of the simulated motion. As it does so, locations that were treated as novel during simulated motion are now treated as recognized during physical motion. This models the main prediction from Chapter 2, which is that inputs from the entorhinal cortex during simulated motion drive the assignment of novel place populations in the hippocampus. That is, some of the trajectories that are observed in the hippocampus during quiet waking periods are driven by the animal's simulated motion through space. The outputs from the entorhinal cortex components at each step in the animal's physical or simulated run activate different, sometimes novel, populations of place cells in the place system.

As discussed in Chapter 2, much activity within the hippocampus is thought to originate there, i.e., to have an endogenous source. An example is the replay of past experience (Wilson & McNaughton, 1994). The Pfeiffer and Foster (2013) results, suggest that some hippocampal activity is qualitatively different, i.e., "preplays" future activity. This makes it plausible to ask whether this activity also has an endogenous source, or might have an exogenous source. An endogenous source might be considered the default hypothesis, e.g., as suggested by Dragoi and Tonegawa (2011), that endogenous processes within the hippocam-

pus make populations of cells available for use in representing novel locations. This work suggests the alternate hypothesis, that the preplay trajectories have an exogenous source. I suggest that these trajectories are the byproduct of simulated motion originating in other cortical areas, and projected into the hippocampus from the entorhinal cortex. Some recent findings (Chapter 2) strongly suggest that the hippocampus is not immediately retracing the trajectories being projected by the entorhinal cortex (although other findings do support the simulated motion hypothesis). Although alternative mechanisms could be envisioned, SMS does not reflect such an alternative. Rather, it currently implements the simple hypothesis of immediate communication between entorhinal cortex and hippocampus.

SMS demonstrates the simplest version of the central hypothesis of this study, regarding simulated motion. There are many empirical findings from the neuroscience of the hippocampal formation that have not been addressed. As the software is open source and fairly stable, the system offers many opportunities for future work, to be discussed in Chapter 7. Appendix A discusses engineering considerations that should help make future work productive.

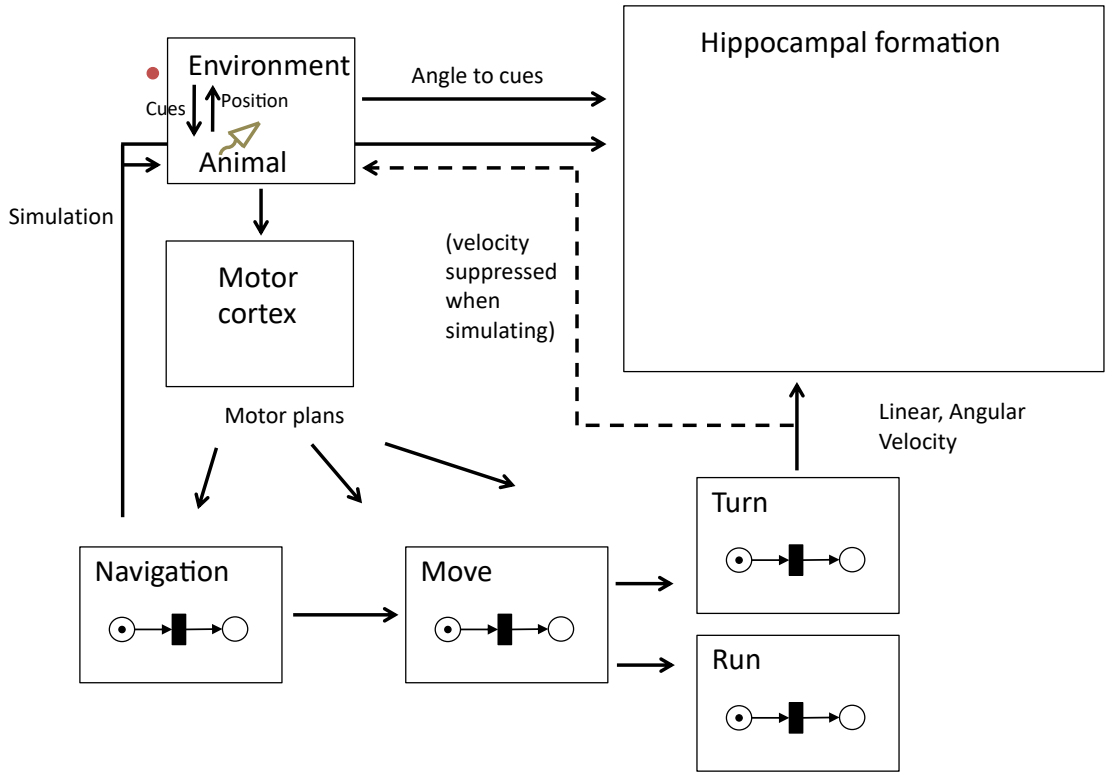


Figure 4.1: System data flow

4.2 Simulated Motion System

SMS consists of a system of software components modeling various neural subsystems, driven by a model of rat behavior implemented as Petri nets. The system is implemented in Matlab, Petri nets, and Java, and is maintained as open source: `https://github.com/sjdayday/simulated-motion`. The model simulates an animal's behavior as it explores an open arena. The data flow of the system is depicted in Figure 4.1. The subsystems modeled in most detail are those in the hippocampal formation; the data flow in this area is depicted in Figure 4.2.

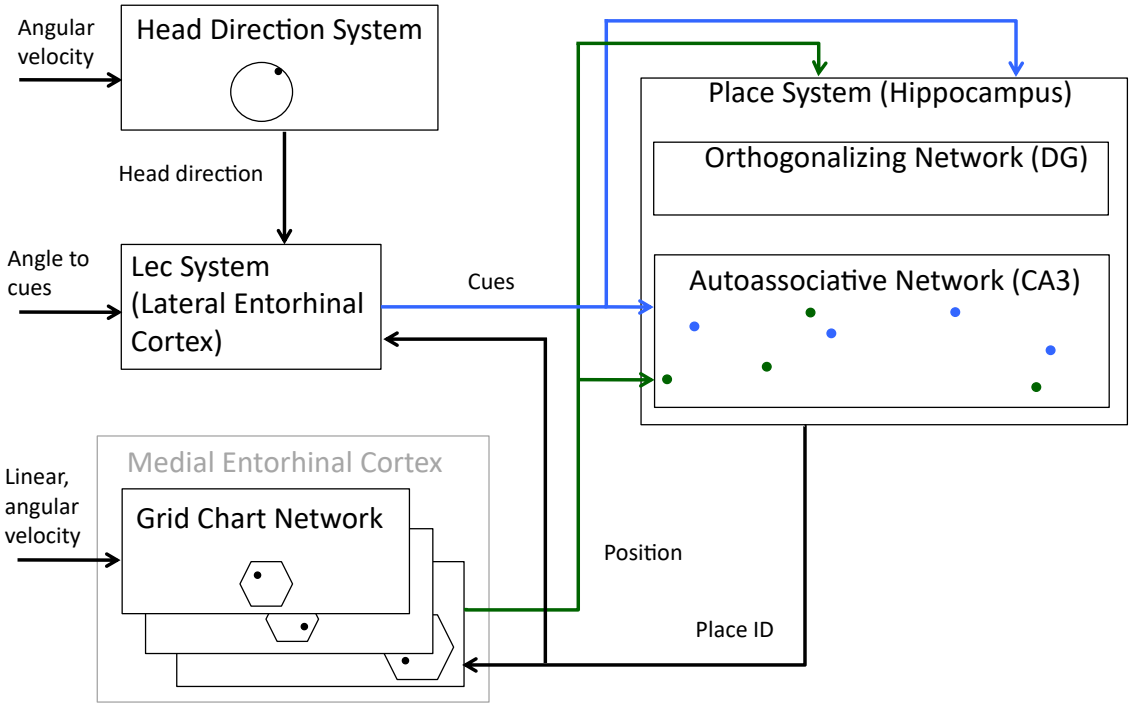


Figure 4.2: Data flow within the hippocampal formation

4.2.1 Components

The components of the system are modeled at a minimal level of detail, to keep the system tractable. The component name is meant to be suggestive of functions of one or a group of related neural subsystems. Components within the hippocampal formation are modeled in more detail, closer to their neural counterparts of the entorhinal cortex and hippocampus. The Environment and Animal components represent the physical arena and the physical rat behaving within that arena; the remaining components represent some aspect of neural functioning. The components are implemented as Matlab classes, and the overall behavior of the system is driven by Petri nets, integrated with the Matlab classes through the extensions described in Chapter 3.

Each component has a number of parameters, which govern the details of its behavior. No attempt has been made to find an optimal set of parameter values, beyond a set that enables the basic behaviors of the overall system to be demonstrated. Optimizing the parameter values is one of a number of opportunities for future work.

The components are described below, followed by a discussion of assumptions and the processing of the model. Component names are the names of the corresponding Matlab classes[1].

Environment: The Environment represents the physical arena within which the Animal moves. It consists of a set of walls and visual cues. The first cue added is designated as the most salient cue; this designation does not change. No attempt is made to model how salience is determined. The Environment calculates the angle from the Animal to the cues, and makes this information available to the Animal. (These calculations are performed in the Environment class for programming convenience; corresponding calculations in real animals are presumably processes in cortical areas not modeled in SMS.)

[1] `https://github.com/sjdayday/simulated-motion`

Animal: The Animal represents the physical rat, moving within the Environment. The Animal's physical position is updated as the result of the execution of motor plans. The Animal receives information from the Environment as to the distance and orientation to walls and cues. The Animal has *whiskers*, which detect when the distance to a wall is within some threshold.

MotorCortex: The MotorCortex generates and executes motor plans, termed Behaviors. The lowest level Behavior is a Turn or straight Run in the Environment, for some units of angular or horizontal displacement. In each time period, the animal executes a Turn or a Run for unit distance. The Turn or Run may be physical or simulated. The MotorCortex has two modes of operation. In normal mode, the linear or angular velocity from the current time period is sent to the Animal, resulting in a change in the Animal's physical position. In simulated mode, the velocity signal is suppressed, and the Animal does not move. In either mode, however, the linear or angular velocity is sent to the HippocampalFormation for processing, analogous to efferent copy. The sequence of Turns or Runs and transitions between normal and simulated mode is controlled by a higher level Behavior, called Navigate. Navigate generates a random sequence of Move Behaviors; a Move is either a Turn or Run, in either normal or simulated mode. When the Animal's whiskers detect proximity to a wall during physical motion, the current Turn or Run is interrupted, and the next Move is to turn away from the wall.

Petri nets: Petri nets are used to model a Behavior, and to model key events and processes of the overall system. The implementation of Petri nets makes use of prior work by the author, described in Chapter 3. As outlined above, there are four Behaviors modeled as Petri nets: Run, Turn, Move, and Navigate. Each of these has a corresponding Matlab class, which is driven by state changes in the Petri net to update the position of the Animal in the Environment, and to update the MotorCortex and other cortical components.

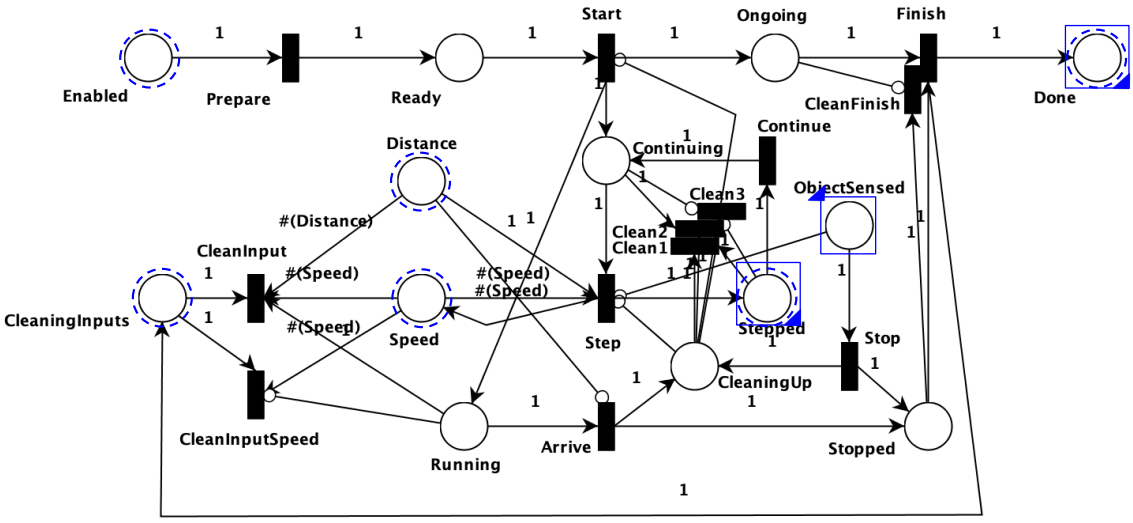


Figure 4.3: Run Petri net

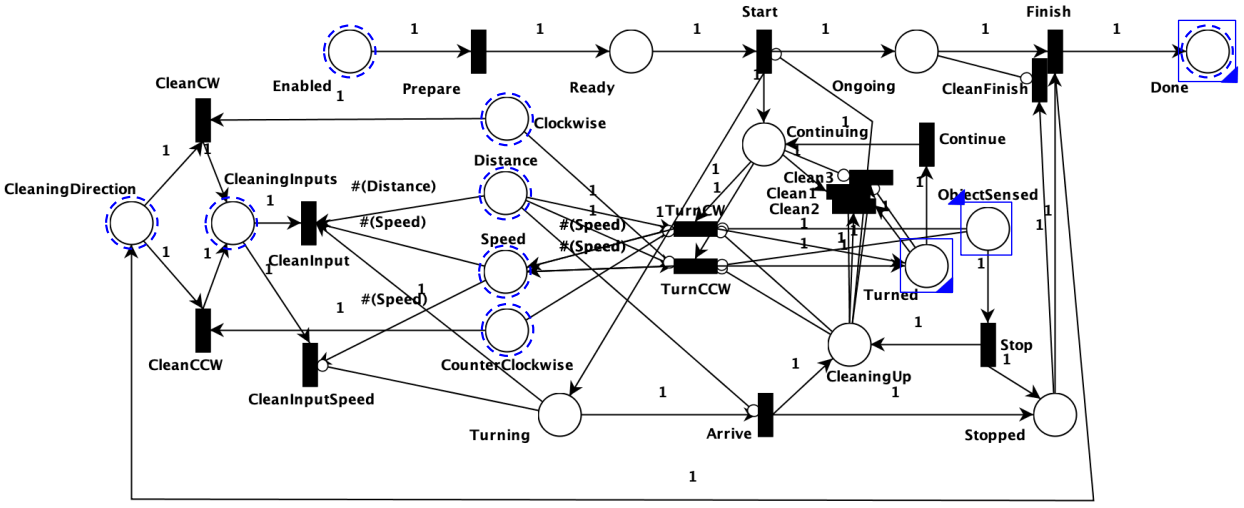


Figure 4.4: Turn Petri net

run: models a straight run until a distance is reached, or until an object is sensed (whiskers touching a wall). Accepts parameters Distance and Speed (Figure 4.3).

turn: models turning either clockwise or counterclockwise until a distance is reached, or until an object is sensed (whiskers touching a wall). Accepts parameters Distance, Speed, Clockwise or CounterClockwise (Figure 4.4).

move: models motion, either turning or running. Accepts parameters Distance, Speed, Clockwise or CounterClockwise, Run or Turn, and invokes *run* or *turn* (Figure 4.5)

navigate: models random navigation, consisting of a sequence of Moves defined by MotorCortex. When a move is Done, MotorCortex is notified, and determines the next move. Each

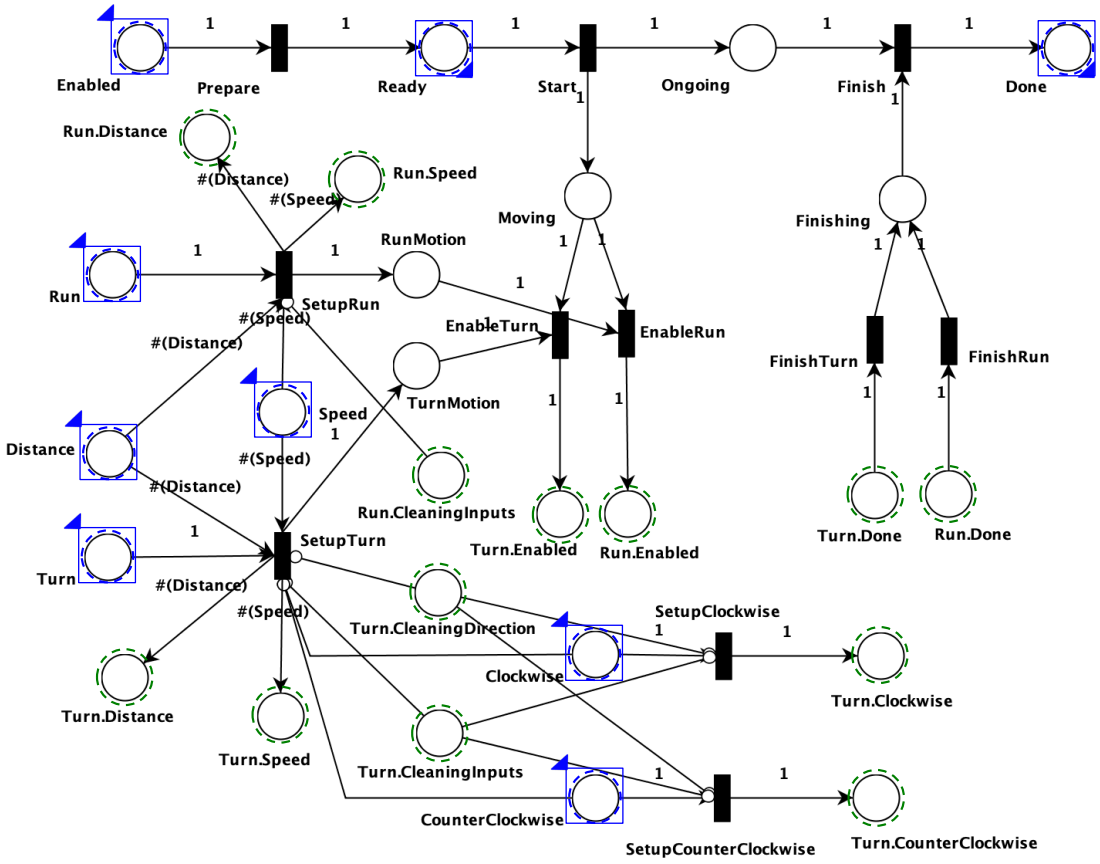


Figure 4.5: Move Petri net

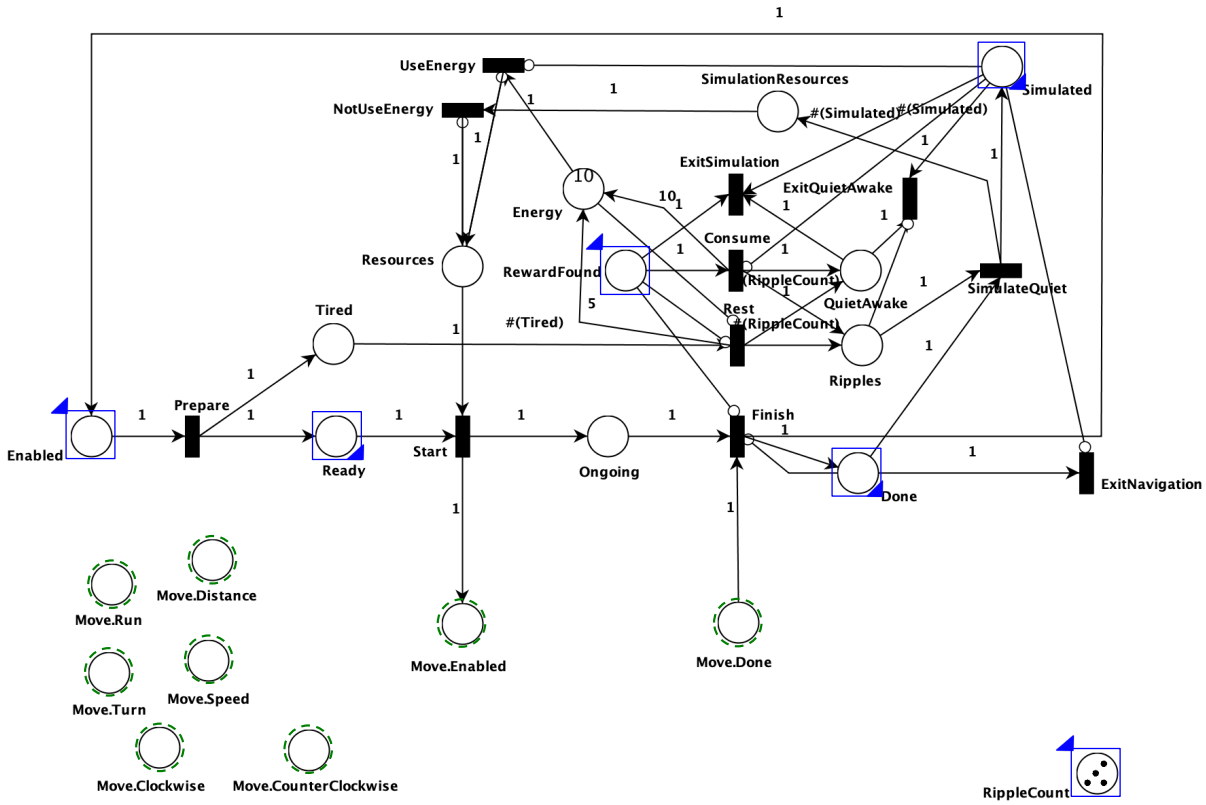


Figure 4.6: Navigate Petri net

move consumes Energy. When Energy is depleted, the animal is Tired and Rests, entering a QuietAwake state, where subsequent moves are Simulated. The number of simulated moves is the number of Ripples. Resting replenishes Energy, and the animal resumes physical motion. The first moves upon exiting simulation are to retrace the first Run (including any prior Turns) that was executed while the animal was in Simulated mode. This sequence is tracked by MotorCortex, which sets the moves accordingly. Once a simulated run (if any) is retraced, the animal returns to random navigation. Random navigation continues until a total number of steps is reached, at which point MotorCortex no longer creates new moves, and the navigation stops. Accepts parameters Turn or Run, Distance, Speed, Clockwise or CounterClockwise, and RippleCount, and invokes *move* (Figure 4.6).

The interface from Matlab to the execution of Petri nets is a PetriNetRunner, which instantiates an include hierarchy of the four Petri nets, with Navigate at the top of the hierarchy, then Move, and then Run and Turn. The PetriNetRunner marks places in the Petri nets as requested by the Matlab classes. Matlab classes also register through the PetriNetRunner as listeners for changes to the marking of particular places, and are notified by the PetriNetRunner when such events occur. The PetriNetRunner executes the Petri net until there are no transitions that can fire, at which point the PetriNetRunner exits. Because the MotorCortex updates the Petri net with the parameters for another move when the previous move is done, execution continues indefinitely, until the MotorCortex stops creating new moves.

HippocampalFormation: The Hippocampal Formation is an umbrella component that corresponds to the entorhinal cortex and hippocampus. It sends and receives information to and from the Animal and MotorCortex and distributes it appropriately among the following components: HeadDirectionSystem, LecSystem, multiple GridChartNetworks, and the PlaceSystem.

HeadDirectionSystem: The HeadDirectionSystem receives an angular velocity signal from the HippocampalFormation and maintains a representation of the Animal's head direction

relative to the head direction in the previous time period. The system implements continuous attractor dynamics for a single dimension as in Trappenberg (2010), following the head direction model from Skaggs et al. (1995), and the implementation is from Zilli (2012). The network is a vector, interpreted as a ring of cells, which comprises the head direction system. In this study, this number of cells is either 30 or 60. The cell number corresponding to the maximum activation of the network is interpreted as the current head direction, and this is output to the LecSystem and the GridChartNetworks.

GridChartNetwork: The GridChartNetwork receives linear and angular velocity signals from the HippocampalFormation and maintains a representation of the Animal's two-dimensional position on a single chart of the environment. The grid is configured as a torus in a matrix of five rows and six columns, for a total of 30 cells per grid. The system implements continuous attractor dynamics for two dimensions as in Trappenberg (2010), roughly following the model from Samsonovich and McNaughton (1997). The activation pattern takes the form of a hexagonal grid, as found in the medial entorhinal cortex. The model of grid activation is adapted from Guanella, Kiper, and Verschure (2007), as implemented by Zilli (2012). The Guanella model makes a simplifying assumption that is not biologically plausible, in that the weights that drive the attractor dynamics are recalculated in each time step and directly include the velocity signal. As a consequence of this simplifying assumption, however, the model generates repeatable maximum activation levels, with low error rates, which makes it sufficient for the purposes of this system. The grid cell with the maximum activation level is interpreted as representing the animal's current position on that grid.

In this implementation, there is no single component that corresponds to the medial entorhinal cortex (MEC). Rather, there are multiple GridChartNetworks, and their separate positional outputs are combined by the HippocampalFormation into a single vector as the output of the MEC. The MEC output is a binary vector of length 30 times the number of grids, where the number of activated positions (value of one) in the vector is the number

of grids. For ease of discussion, this MEC output vector is represented as the indices of its non-zero positions. For example, [22 36] would represent the output from two grids, where the cells of maximum activation are 22 on the first grid, and 6 on the second grid.

The MEC output is input to the PlaceSystem for its processing. The Place ID that is output from the PlaceSystem is then associated with the current position on each of the grids. This association supports the later retrieval of the grid position from a Place ID, for each of the GridChartNetworks. If the maximum activation of each of the grids following this retrieval is the same as the maximum activation when the Place ID was first created, the Animal has successfully recalled the place.

The number of GridChartNetworks is determined by multiplying two parameters. The grid orientation models the observation that different grid cell populations are offset from one another by various angles, such that their activations move in different directions. The grid gain models the observation that grid cell populations vary in spatial scale, increasing in discrete steps moving from dorsal to ventral locations in the MEC. The study results use two grid orientations, and from one to three grid gains, resulting in two to six GridChartNetworks.

LecSystem: The LecSystem models some aspects of the lateral entorhinal cortex (LEC), providing spatial inputs to the PlaceSystem. In the LEC, non-spatial inputs are also included, but these are not modeled in SMS. The LecSystem receives the vector of cues from the Environment, and the current head direction from the HeadDirectionSystem, and calculates a *canonical* view of the cues. For purposes of creating a single representation of a place within the PlaceSystem, the animal's current head direction is rotated to orient directly towards the most salient cue, and all other cues are rotated accordingly. Although such processing is neurologically plausible, it is likely that multiple cortical areas are involved, and it may be that LEC only receives the output of processing that happens elsewhere. Creation of the canonical view enables the encoding of multiple cues in a single vector as the output of LEC, which is then input by the HippocampalFormation to the PlaceSystem. The most salient cue

is always at orientation 0, because the canonical view represents the animal pointing directly to the most salient cue. The orientations to the remaining cues are then represented as the relative head directions at which each cue would appear to the animal in its canonical view. Results in this study were generated using two cues, the most salient cue and one other.

The Place ID that is output from the PlaceSystem is then associated with the current canonical view in the LecSystem. This association supports the later retrieval of the Animal's head direction when it is in the position of the canonical view. If the maximum activation of the HeadDirectionSystem following this retrieval corresponds to the maximum activation when the Place ID was first created, the Animal has successfully recalled its canonical head direction. This recall process is performed when the Animal is *orienting*. When a rat orients, it looks around its environment for salient cues. This is done when an animal is first placed in an environment to help the animal decide whether this is a known or novel environment. Orienting is also done periodically as the animal moves through the environment, which might serve the function of re-calibrating the animal's head direction system. In SMS, orienting means that the animal turns to point directly at the most salient cue, and then uses the resulting LEC output to attempt to recall a corresponding Place ID. From this Place ID the LecSystem then attempts to retrieve the head direction that was originally associated with it, thereby re-initializing the HeadDirectionSystem to its state at that time. Orienting is supported in SMS (see Appendix A), but is not integrated into the full system from which the study results were generated.[2]

PlaceSystem: The PlaceSystem is an umbrella component that models some functions of the hippocampus. It is comprised of two components, the OrthogonalizingNetwork and the AutoassociativeNetwork, corresponding roughly to the dentate gyrus and CA3. For purposes

[2]The canonical view with two cues, as used in this study, is the angle between the two rays from the animal's position to the cues. The set of points in the arena where this angle is constant is an arc. In future work where orienting is fully integrated into SMS, three or more cues would be used to limit ambiguity when orienting. With three cues not on the same line, the animal's position is the intersection point of the three rays defined by the two angles from the salient cue to the other two cues.

of this system, separate models of CA1 and subiculum are not needed. The PlaceSystem receives input from the LecSystem and the GridChartNetworks, retrieves or generates Place IDs, and sends these Place IDs to the HeadDirectionSystem, the LecSystem, and the GridChartNetworks, via the HippocampalFormation. Its only other processing is to initially connect the OrthogonalizingNetwork and the AutoassociativeNetwork together. An important simplifying assumption is that the PlaceSystem only maintains binary representations; all information is encoded as either one or zero. The interpretation is that a given population node is either active or inactive, and this implies that the model cannot accommodate sub-threshold activation.

OrthogonalizingNetwork: The OrthogonalizingNetwork implements one aspect of the processing of a model of the dentate gyrus, to orthogonalize inputs from the LecSystem and the GridChartNetworks. For example, when representing adjacent physical locations, lower error rates will result if the overlap in bits representing the locations is minimized. The OrthogonalizingNetwork establishes a random but stable connection between the input vectors received from the LecSystem and the GridChartNetworks, and the output vector to the AutoassociativeNetwork. Under parameter control, the OrthogonalizingNetwork may or may not perform processing to make the output representation more sparse than the input vectors. The most sparse representation is one where the outputs from MEC and LEC are each hashed to generate a single integer. The resulting pair of integers are used as offsets into the AutoassociativeNetwork to control which synapses are created.

AutoassociativeNetwork: The AutoassociativeNetwork models some aspects of CA3, interpreted as a Hebb-Marr autoassociative network, as defined in McNaughton and Nadel (1990). The underlying data structure of the network is a binary square matrix, where each point may take the value zero or one. The dimension of this matrix is N x N, where N is the length of the input vector to the OrthogonalizingNetwork, i.e., the sum of the lengths of the MEC and LEC input vectors.

The inputs from the OrthogonalizingNetwork are treated analogously to *detonator* synapses, i.e., their firing causes the nodes to which they are connected in the AutoassociativeNetwork to fire with probability one. These inputs from the OrthogonalizingNetwork originated in the LecSystem and GridChartNetworks. Analogously to the perforant path connections from entorhinal cortex to hippocampus, the same inputs are also directly sent to the AutoassociativeNetwork (see Figure 4.2). On this second path the role of these inputs is equivalent to projections to the *modifiable* synapses of the Hebb-Marr model, where they undergo Hebbian learning when firing in conjunction with the firing of inputs from the OrthogonalizingNetwork pathway. As implemented in SMS, these synapses are given a weight of one upon first firing, and do not decay.

The AutoassociativeNetwork operates in two modes. When encoding, both the detonator and modifiable pathways are active, strengthening synaptic connections. When recalling, the detonator pathway is suppressed, and only the signal along the modifiable pathway is active. At this point, the third synaptic type in the Hebb-Marr model becomes involved, that of inhibitory synapses, to control the output pattern that is retrieved from the AutoassociativeNetwork. The inhibitory synapses implement the equivalent of integer division, ensuring that the only output nodes in the AutoassociativeNetwork that fire are those where the number of active inputs is at least equal to the number of strengthened synapses of the original input pattern. This constitutes an implementation of point attractor dynamics. If no pattern is retrieved, the inhibition level is lowered on successive attempts to a threshold, until the inputs match some subset of a stored pattern. If no pattern is ever retrieved by the time the threshold is reached, this serves to indicate novelty. The threshold parameter for this study is set to two, meaning that at least two of the inputs along the modifiable path must match existing synapses to retrieve a pattern.

Any retrieved pattern is interpreted as a Place ID. A Place ID is a binary vector whose length is the sum of the lengths of the MEC and LEC inputs. As a shorthand transcription,

a Place ID (as well as the MEC and LEC input vectors) is written as the indices of the non-zero positions in the vector. For example, in a sparse representation, a Place ID might be [89 131]. The point attractor dynamics may be sufficient to retrieve the complete Place ID given only partial input, e.g., MEC input corresponding to [89], or LEC input corresponding to [131]. Finally, following memory index theory, the Place ID is output to the other components in the HippocampalFormation, where it is stored in association to current activation patterns.

4.3 Model Assumptions

There are a number of important simplifying assumptions made in SMS whose implications should be made explicit.

- States in CA3 are modeled in the AutoassociativeNetwork component as binary; an individual node is activated or not activated. This means that concepts that depend on sub-threshold activation cannot be modeled.

- The only output from the PlaceSystem is a Place ID, which is stored by other cortical areas in association with the current local cortical state, so that local state may be retrieved later. This is an implementation of memory index theory. One immediate function this enables is continuous attractor dynamics in the HeadDirectionSystem, LecSystem and the GridChartNetworks. Although hippocampal outputs might drive other processing, that is not provided for here.

- As implemented, motion is discrete, not continuous. The animal moves a single step at a time, generating outputs from MEC and LEC and creating a Place ID, which is then associated with the current activation in MEC and LEC. Only then is the next step taken. This has the advantage of enabling the inputs and outputs across time

periods to be associated with low error rates. This assumption might be supported by observing that rat physical motion is frequently discontinuous, with small starts and stops. The evidence of trajectories during quiet waking periods also does not show continuous motion. Pfeiffer and Foster (2015) suggest the observed discontinuities in SWR trajectories might have the effect of lowering the error in autoassociative dynamics, which might support my assumption. Finally, this assumption plays an analogous role in SMS to that hypothesized to be played by a visual saccade (although not in rats) in visual information processing, resetting the visual system such that there is a clear delineation between two successive images.

– As discussed in the description of the LecSystem, I assume that the PlaceSystem stores a canonical local view, consisting of orientation to a cue, associated with a given place field. The canonical local view assumes that the animal's head direction is oriented directly towards the most salient cue, and all other cues are rotated accordingly. This has the advantage of eliminating duplicative information in the place system, representing all possible orientations toward cues. It also supports a notion of object constancy, such that the animal recognizes that there is a single object, not a different object for each possible orientation and distance. This assumption is supported by findings from Navratilova, Hoang, et al. (2012) that in an arena, place fields appear not to have an orientation, i.e., firing rates are similar in all directions. (In a linear track environment, however, differential firing rates develop over time from the initial uniform pattern, suggesting that the two directions develop an independent salience.)

  Note that this assumption has another effect, which is that new Place IDs are only allocated during runs, not turns. Whenever the animal moves to new cartesian coordinates in the arena, the outputs from both MEC (the GridChartNetworks) and LEC (the LecSystem, which will calculate a new canonical view of the salient cue) will change, causing a new Place ID to be created. If the animal turns its head at the existing coordinates, this updates the HeadDirectionSystem, but does not change the

canonical view of the salient cue. Therefore, the LEC output will be unchanged, and since the cartesian coordinates are also unchanged, the MEC output is also unchanged, such that the Place ID remains the same.

– I assume that some neuromodulatory process drives two different modes of processing in the PlaceSystem:

  – *encoding*: novel places are stored, driven by detonator synapses (DG granule cells), and associated to MEC and LEC inputs

  – *recall*: known places are retrieved from MEC and LEC inputs; firing of detonator synapses is suppressed.

  At each step, the PlaceSystem first attempts to recall a Place ID; successful recall constitutes "recognizing" the place. Subsequently, the MEC and LEC inputs create or update a Place ID.

– During simulated motion, the angular and linear velocity resulting from Turn or Run are used to update the activation of the HeadDirectionSystem and GridChartNetworks. This reflects the hypothesis that even simulated motor processing is tracked by those two systems, which track a relative position at least in part from internal self-motion information. On the other hand, I assume the LecSystem does not generate a simulated canonical view of cues. The canonical view calculation can be interpreted as a shortcut equivalent of calculation that probably takes place at least in part in visual cortex. Since the animal is not physically moving, presumably sensory cortical input is unchanged, making calculation of an updated canonical view of cues an implausible assumption. This has the effect of limiting the input to the PlaceSystem during simulated motion to just the MEC output vector; the LEC output vector is zeros, and therefore does not drive the creation of new synapses. This means that any Place IDs created during simulated motion will be partial, representing only the MEC inputs. If the animal retraces the same path after returning to physical motion, then such Place

IDs will be updated to add the LEC input. For example, a sparse simulated Place ID might be [89]. If the same place is revisited during physical motion, it might be assigned Place ID [89 131], reflecting the same MEC inputs with the addition of new LEC inputs.

– At the end of each simulated run, the animal is in the same physical location, and its physical head direction is also unchanged, although the activation in the GridChart-Networks reflects the simulated motion. In addition, new Place IDs have been created in the PlaceSystem. I assume that there is a mechanism for the animal to reset its simulated position back to its physical position.[3] In SMS this process is called *settling*, as in "settling back to where I was". (Settling is a general mechanism in SMS, although in this study, this is the only circumstance where it is integrated into the system's behavior. In the context of managing the proliferation of Place IDs, discussed below, the literature uses the term *fixing position* for an equivalent process.)

– Each period of simulated motion begins from the animal's current physical position and head direction. This means that the system does not generate trajectories that begin at locations remote from the current position. Such trajectories could be done as part of future work (see Appendix D).

[3]As humans, we say "remember where I am" when our thoughts take us somewhere else.

4.4 Model Processing

Processing begins with ExperimentController, a class that enables the model to be run with different pseudo-random input seeds, and different parameters. Each such combination is termed a scenario, and each scenario generates a detailed debugging file and a csv file of statistics for later analysis. A scenario parameter governs the number of time steps the scenario runs. A time step corresponds to a unit motion, either turn or run, in simulated or real mode.

For each scenario, the ExperimentController instantiates an Environment with a 2 x 2 arena, and an Animal with the assigned parameters. The Animal is placed in the arena at location [1,1], facing orientation 0 on the unit circle (pointing "east").[4] The Animal in turn instantiates its internal cortical components, the MotorCortex and HippocampalFormation. The HippocampalFormation in turn instantiates the HeadDirectionSystem, the LecSystem, the GridChartNetworks, and the PlaceSystem.

The MotorCortex begins random physical navigation by creating a PetriNetRunner with the navigate Petri net. The *navigate* Petri net includes the three other Petri nets, *move*, *run*, and *turn*. The four Petri nets are combined to make one executable Petri net. Places in the executable Petri net are uniquely identified by prefixing their names with the names of their position in the hierarchy. For example, the Enabled place in the navigate Petri net is named Navigate.Enabled, while the Enabled place in the run Petri net, at the bottom of the hierarchy, is named Navigate.Move.Run.Enabled.

The MotorCortex establishes a set of listeners for key events in the executable Petri net, and registers them with the PetriNetRunner. The two most important high-level events are associated with marking the Navigate.Done and Navigate.Simulated places. When Navigate.Done is marked, it indicates that the current Move is complete, and MotorCortex should

[4] In more extensive future work, the animal would be placed at a random location in the arena.

generate the next Move. When Navigate.Simulated is marked, going from zero to one tokens, MotorCortex puts multiple components into *simulated* mode. When Navigate.Simulated is cleared, going from one or more to zero tokens, MotorCortex takes multiple components out of simulated mode, and resumes physical motion.

The MotorCortex also establishes listeners for two other key events, which drive the bulk of the cortical processing in the system, those associated with marking the Navigate.Move.Run .Stepped and Navigate.Move.Turn.Turned places. Each of these drives motion of the Animal, and updates the cortical representation of that motion in the HippocampalFormation, detailed below.

The MotorCortex randomly generates a Move, and requests that the PetriNetRunner marks places in the executable Petri net corresponding to the places of the Move: Navigate.Move .Distance, Navigate.Move.Speed, and either Navigate.Move.Turn or Navigate.Move.Run. If the move is a Turn, Navigate.Move.Clockwise or Navigate.Move.CounterClockwise are marked. At the beginning of scenario execution, Navigate.RippleCount is marked with the number of Ripples that will control the number of simulated Moves in each quiet waking period. Finally, Navigate.Enabled is marked; this enables the Navigate.Prepare transition for firing, and thus starts Petri net execution.

Control of MotorCortex processing is governed by the interaction of the executable Petri net and an internal finite state machine called NavigationStatus. NavigationStatus has one standard method, nextStatus(), which is invoked whenever the listener for the Navigate.Done place is dispatched. The nextStatus() method determines what move is appropriate under the current NavigationStatus and sets up that move by requesting that the corresponding places be marked in the executable Petri net. Finally, the current status then updates the NavigationStatus of the MotorCortex to the NavigationStatus that will be appropriate once the new move completes. This processing manages the logic of various transitions, for example:

- Normal physical navigation to simulated navigation
- Simulated navigation to retracing the first simulated run (including any prior simulated turns)
- Normal physical navigation to turning away from a wall when whiskers detect proximity

As transition firing progresses in the executable Petri net, either of two places is eventually marked, Navigate.Move.Run.Stepped, or Navigate.Move.Turn.Turned, corresponding to a unit step or unit turn, respectively. The corresponding listener invokes the Animal run() or turn() methods. The angular and linear velocity of the animal is calculated and updated in the HippocampalFormation. If the animal is not in simulated mode, its new physical position in the arena is calculated.

Next, the step() methods of the ExperimentController and the Animal are invoked. The ExperimentController step() method collects statistics for reporting. The Animal step() method in turn invokes the step() method in the HippocampalFormation. HippocampalFormation step processing has the following sequence:

1. *HeadDirectionSystem*: a non-zero angular velocity will be used to update the activation function in either a clockwise or counterclockwise direction, possibly resulting in a new value for the cell corresponding to the maximum activation.
2. *GridChartNetworks*: the updated head direction is combined with any linear velocity to calculate a cartesian velocity vector, which is applied to each GridChartNetwork. This is used to update the activation function of each GridChartNetwork, possibly resulting in a new value for the cell corresponding to the maximum activation for each GridChartNetwork. An mecOutput binary vector is built that is the length of the number of cells in a single GridChartNetwork, multiplied by the number of grids. For example, the default size of a GridChartNetwork is 30 cells; four grids results in a

vector of length 120. In each 30 cell segment, all values are 0 except for the position corresponding to the cell with maximum activation, which is set to 1. For four grids, mecOutput will have four vector positions with value one, for input to the PlaceSystem.

3. *LecSystem*: the updated head direction is used to calculate the canonical view head direction. That is the activation that would result if the animal's current physical direction was rotated to point to the most salient cue, and its current internal head direction was then rotated by the same amount. Then the current head direction activation is copied from the HeadDirectionSystem to the LecSystem, and rotated by the same amount, leading to a canonical view head direction that is identical to that of the HeadDirectionSystem, with the canonical offset. An lecOutput binary vector is built that is length of the number of head direction cells, and the cell that corresponds to the canonical view maximum activation is set to 1. The lecOutput vector is then input to the PlaceSystem.

4. *PlaceSystem*: The mecOutput and lecOutput vectors are concatenated as a single binary vector and presented to the PlaceSystem as input. First, an attempt is made to recall a Place ID from the AutoassociativeNetwork using the input vector. (The attempt to recall a Place ID does not affect subsequent processing as used for this study's results; it is only used for fixing position, as discussed in Section 4.3.) Regardless of whether the place is recognized, the PlaceSystem is next updated with the input vector, possibly creating a new Place ID or overwriting an existing Place ID. In either case, the Place ID is output back to the HippocampalFormation where it is associated with the current maximum activation in the LecSystem, and the GridChartNetworks. Finally, depending on the current NavigationStatus in the MotorCortex, the Place ID may be logged as part of tracking a simulated run or physically retracing a previously simulated run.

As random navigation proceeds, eventually the system will enter simulated mode, and begin to make simulated moves. As noted in the Model Assumptions (Section 4.3), each simulated move takes the animal's simulated position or head direction away from its physical position or physical head direction. Following each simulated run, the HippocampalFormation settles back to the current physical Place ID, which is saved upon entering simulated mode. Settling is accomplished by retrieving the GridChartNetwork activation that was associated with the current physical Place ID, and using that to update the GridChartNetwork activation repeatedly until the cell of maximum activation stops changing (hence the term "settling"). Usually, the updated activation results in a cell of maximum activation that is the same as the cell of maximum activation at the time the original Place ID was created, such that the animal's simulated position is the same as it was when it entered simulated mode. The MotorCortex then randomly generates the next simulated move.

The settling process is different for turns, because the animal's current head direction is not associated with a Place ID. Instead, as discussed in the Model Assumptions, what is associated with a Place ID is the canonical head direction. Therefore, we don't have a simple way to retrieve the head direction associated with the last physical turn. Consequently, the approach taken is to do a simulated reverse turn at the end of simulated motion, before returning to physical motion. The distance and direction of the turn are the opposite of the sum of the turns taken during simulation. This returns the HeadDirectionSystem to something approximating its prior state upon entry to simulated motion.

4.5 Managing the proliferation of Place IDs

This section addresses the state of efforts to deal with one of the current weaknesses of SMS, which is managing the proliferation of Place IDs in the PlaceSystem. This is the subject of

several of the internal results to be discussed in Chapter 6, and the area that offers the most immediate opportunity for improvement in the model.

As SMS currently works, each step of a run that results in a change to the maximum activation of one or more of the GridChartNetworks or a change in the canonical view of the LecSystem may result in the creation of a new Place ID in the PlaceSystem. If the MEC and LEC inputs are identical to those presented in some previous step, the previous step's Place ID will be updated, resulting in no change. As a practical matter, this happens relatively rarely, because without using the settling or orienting mechanisms, the system drifts out of calibration fairly quickly, such that retracing the same physical path after some time steps is unlikely to result in the same Place IDs being retrieved.

The AutoassociativeNetwork used to model CA3 has the virtue of simplicity, as its binary structure supports the Hebb Marr attractor dynamics directly. Because synapses do not decay, however, a Place ID, once created, is never "forgotten". This means that as the animal explores, the number of activated synapses (those with value one) in the network will continue to increase. As the number of activated synapses approaches 50% of the total synapses, the number of retrieval errors becomes unmanageably large, and the network is said to be saturated.

Three strategies that might be taken to managing the proliferation of Place IDs are as follows, in ascending order of change to the overall system.

1. *sparseness*: make the Place ID representation as sparse as possible to slow the rate of saturation.

2. *fixing position*: use the settling mechanism described above to avoid creating a new Place ID when a Place ID is retrieved that is sufficiently "close".

3. *continuous synapse values*: implement the synapses in the AutoassociativeNetwork as continuous in [0,1], and enable synapses to weaken to the point of "forgetting".

As will be discussed in the results, the level of saturation seen in the PlaceSystem did not interfere with obtaining the primary result of this study, so this problem has not yet had to have a final resolution. Secondary results will address the value of the first strategy of maximizing sparseness. The next step would be to implement the second strategy, of fixing position. It seems likely that this would be sufficient to stabilize the saturation levels. The third strategy would be a major change to SMS; it would add significant complexity to the system, while also adding to the system's power, by being able to model sub-threshold dynamics in the PlaceSystem. I have spent no time exploring this option, so have no further detail to add.

Maximizing the sparseness of the representation in the network is the most basic strategy for limiting saturation in Hebb-Marr networks, and is explored in detail in Chapter 6. In SMS, this has considerations from the perspective that models detonator synapses from the OrthogonalizingNetwork, and from the perspective that models modifiable synapses from the perforant path directly from MEC and LEC.

– Increasing the sparseness of the outputs from the OrthogonalizingNetwork should help. The initial implementation of the OrthogonalizingNetwork did not increase sparseness; typically there were four inputs from GridChartNetworks and three from the LecSystem, resulting in seven detonator synapses firing. A later implementation used a hash on each of the MEC and LEC inputs, resulting in one detonator synapse firing for the MEC input, and one for the LEC input. The effect of these two implementations was tested via the sparse Place ID parameter, discussed in Chapter 6.

  I suggest that reducing the representation further, from two values to one value, is undesirable for two reasons. Maintaining separate inputs from MEC and LEC enables

pattern completion to retrieve a Place ID when either input is later presented. When the animal is placed back in a previously explored arena, the orienting process may result in LEC output that is the same as previously encountered. This partial input should be sufficient to retrieve the original Place ID, which is then used to reset the GridChartNetworks to their previous activations, through the settling process.

Pattern completion works in the opposite direction when physically retracing previous simulated trajectories. During simulated motion, we have only inputs from MEC, as discussed in the Model Assumptions. This creates a one-digit Place ID, reflecting only the MEC inputs. When the place is later revisited, the Place ID is retrieved from the MEC inputs, and the Place ID is updated with the addition of the LEC inputs, while the MEC portion of the Place ID is unchanged.

– Changing the number of inputs and the size of the input vector for the modifiable synapses will also affect saturation levels. Decreasing the granularity of the inputs from MEC and LEC should increase the size of place fields. That is, for sufficiently small absolute changes in linear or angular displacement there should not be changes in the cells corresponding to maximum activation of the GridChartNetworks and HeadDirectionSystem. Chapter 6 includes some analysis of the impact of changing parameters to affect place field size, by varying the number of GridChartNetworks and the number of cells for the HeadDirectionSystem. Note that the parameter for variation of the number of GridChartNetworks was the spatial gain, so doubling the number of GridChartNetworks from two to four would not be expected to have a linear impact on the granularity of place fields. The additional GridChartNetworks have a larger gain, and are therefore less sensitive to small absolute changes in linear displacement. The number of cells in a single GridChartNetworks could also have been varied, e.g., to nine rows by ten columns (90 cells, a more granular structure), but was left constant at five rows by six columns (30 cells).

Increasing the size of place fields has a potentially offsetting consideration, depending on whether the implementation uses sparse Place IDs, or not. If sparse Place IDs are used, and the unit distance of linear or angular displacement is big enough to generally trigger a change in the corresponding MEC or LEC outputs, the mathematics of saturation favors increasing granularity (i.e., decreasing the size of place fields), rather than decreasing granularity. For example, if the number of GridChartNetworks increases linearly, from two to four or six, the size of the MEC input vector increases linearly by a factor of 30 times the delta, but the capacity of the PlaceSystem network grows as the square of this increase.

The increase in the number of new synapses created for a novel place ID differs depending on whether sparse Place IDs are used or not. If sparse Place IDs are used, the number of detonator synapses used is unchanged when the number of GridChartNetworks increases, as the hash will always only have two outputs, so the increase in the number of new synapses is only linear. This is turn suggests that saturation levels will drop. On the other hand, if sparse Place IDs are not used, an increase in the number of GridChartNetworks will cause an increase in both the EC inputs and in the detonator synapse inputs from the OrthogonalizingNetwork, so the increase in the number of new synapses is as the square of the increase in the number of GridChartNetworks. For example, if the number of GridChartNetworks goes from four to six, the number of new synapses attributable to MEC inputs (putting aside the LEC inputs) will go from 4^2 to 6^2. In this case, since both the number of new synapses created and the size of the PlaceSystem grow as the square of the increase in the number of MEC inputs, the expected effect on saturation is not as clear. This is discussed further below, in Chapter 6.

Once having decided on an approach to optimizing the sparseness of the AutoassociativeNetwork, the second strategy for addressing saturation is to implement a mechanism for fixing

position. This means using the settling mechanism that has been described earlier to avoid creating a new Place ID when sufficiently close to an existing Place ID. "Close" means attempting to retrieve a Place ID with a high threshold for matching, so that most of the MEC and LEC input values match an existing Place ID. For example if there are four grids and one value from LEC, then matching on four of five inputs might mean that one of the grids does not match. This is likely to represent a place field that is quite close to the animal's current position, as most of the grids are the same and the canonical head direction unambiguously puts the animal in the same area of the arena. On the other hand, depending on the granularity of the GridChartNetworks, there could be multiple physical locations in the arena that have the same four grid values. If the match is for all four grid cells, but not LEC, the match would be spurious, because the different canonical views mean the physical place fields are far apart. If a close Place ID is retrieved, then settling would reset both the GridChartNetworks and the LecSystem to the maximum activation that had previously been associated with the retrieved Place ID. This would avoid creating a novel Place ID. As the arena is increasingly explored, the probability of retrieving a close Place ID should increase, and the number of novel Place IDs should level off, putting a limit on the saturation level of the AutoassociativeNetwork.

The fixing position mechanism has some neurological plausibility; we observe that after the animal has explored the arena, the number of place fields does not increase indefinitely. Place fields can change in size and shape if cues change or the arena itself changes, but that is not the case for the simple scenarios of SMS.

Some unit testing has been done for this settling approach to managing saturation. At some level, we can therefore say SMS supports this approach, but it has not been integrated into the behavior of the full system.

This concludes the basic review of SMS, at a level of detail sufficient to explain the questions addressed in the study results. Those interested in making use of SMS will find additional detail in Appendix A.

Chapter 5

Methods

5.1 Overview

The Simulated Motion System (SMS) was executed over a limited parameter space to generate data reflecting different aspects of model behavior. In this chapter I explain these parameters and their expected impacts on the behavior of the model in three areas. Once the parameters have been detailed, I explain how the model was executed and data were gathered for analysis.

First, the parameters shed light on the model's support for the hypothesis that simulated motion could generate some of the trajectories that are observed in the hippocampus. As the animal explores a novel environment, during quiet waking periods it simulates motion to places in the environment. This will result in the creation of new Place IDs or reactivation of existing Place IDs in the Place System. Upon exiting the quiet waking state and resuming physical motion, the animal attempts to physically retrace the trajectory that was just simulated. The specific behavior to be measured is the percentage of time the animal physically retraces a trajectory that has just been traversed under simulated

motion. In support of the simulated motion hypothesis, the successive activation of multiple Place IDs during a simulated run would provide an economical account of the generation of trajectories that predict the animal's subsequent physical motion. An additional implication is that this mechanism would support the hypothesis of an exogenous source for the activation of populations of cells in the hippocampus to represent novel places.

Second, the parameters should affect the efficiency with which the animal explores the arena. The Pfeiffer and Foster results establish the fact of simulation – rats simulate their future motion – but do not suggest what the function of such simulated motion might be. One simple theory is that simulation is a way to explore a novel environment at lower energy cost, than by physical motion alone. Under this theory the more simulation the animal does, the lower the total energy cost of exploration. Simulated exploration doesn't provide as much information as physical exploration, but it does provide some information, and so has some value.

Lastly, the parameters affect an important internal behavior of the model, the rate at which the memory in the PlaceSystem becomes saturated. If we accept the hypothesis that CA3 functions as an auto-associative network, it becomes important to minimize the rate at which that network becomes saturated, to minimize errors in retrieving Place IDs. Analyzing the effect of several parameters on the saturation rate should provide a sanity check on the overall behavior of SMS. Demonstrating that the actual behavior of the model is consistent with what would be expected from its design, should give additional confidence in the overall behavior of the system.

5.2 Model parameters

To measure its various behaviors, SMS was executed under 36 scenario sets, where a scenario set is defined by a unique combination of parameters. Under each individual scenario a new animal is created and placed in the arena to explore randomly for 3000 time steps under the control of the Navigate Petri net. A scenario set consists of 20 scenarios, where each scenario starts with a different seed for the pseudo-random number generators for Matlab and for the PetriNetRunner (see Appendix A). The value of 3000 time steps was chosen empirically, as roughly sufficient time for the animal to explore about 90% of the arena.

SMS has a large number of parameters, whose potential combination creates a vast parameter space. To make at least a preliminary analysis manageable, two large classes of parameter were left unchanged, i.e., not chosen for variation. The models of individual cortical regions – HeadDirectionSystem, GridChartNetwork, and to a lesser extent, OrthogonalizingNetwork and PlaceSystem – each have a number of parameters that govern their behavior in isolation. Some of these parameters correspond to the variables in the underlying equations that comprise the essence of the model. Apart from some ad hoc manual testing and some automated unit tests (see Appendix A) these parameters were left largely unchanged, once the initial behavior of the component was verified. For example, the current parameters for HeadDirectionSystem enable it to update its activation to respond to an angular velocity signal in a predictable manner. So, if the animal physically turns pi/30 radians clockwise, the HeadDirectionSystem maximum activation will move clockwise by approximately the same amount. Although these internal HeadDirectionSystem parameters could be further tuned, their current behavior is sufficient for the purposes of their function in the overall system, so they have been left constant.

Another large class of parameters being left unchanged, governs the logic of the behavior of SMS. These are discussed in further detail in Appendix A, but at a high level, they represent

Table 5.1: SMS model parameter space, defining 36 scenario sets.

Parameter	*Values*
sparsePlaceId	0, 1
ripples	2, 4, 6
grids	2, 4, 6
headDirectionCells	30, 60

different modes of possible interaction among the individual components. To a significant extent, they represent the evolution of the software over time, as individual components went from working in isolation to working in concert with other components. These parameters currently enable the system to work as an integrated whole, to generate the results discussed below. Changing these parameters would generally cause the system to behave in an older manner, with less functionality. In some cases the system would behave in a newer manner, representing individual component behavior that has not yet been tested for its overall impact. An example of this are the parameters governing "settling" (see Chapter 4), which has not been integrated into the overall system behavior, and is an area for future work.

To address the aims of this work, four parameters were chosen to vary. These address the neuroscience predictions and the internal behavior of the system around saturation levels in the PlaceSystem. Although these are not the only parameters that could bear on these questions, they do allow useful conclusions to be drawn. Table 5.1 summarizes the parameter space explored through the 36 scenario sets. At 20 scenarios per scenario set, this generated 720 model runs.

5.2.1 Ripples

The Navigate Petri net models the process where the animal navigates randomly and rests periodically. During these quiet waking periods, the animal simulates possible future motion. Each simulated motion corresponds to a *ripple*, a reference to the sharp wave ripples that

coincide with the trajectories observed in the hippocampus. The number of ripples is the number of motions, either turns or runs, that the animal will make during one quiet waking period. The range of ripples used in the scenario sets is two, four, or six. As the number of ripples drives the amount of simulated motion, we would expect this parameter to affect the demonstration of the simulated motion hypothesis, as well as the efficiency with which the animal explores the arena.

5.2.2 Grids

Having multiple GridChartNetworks models the observation that different populations in the MEC seem to function independently in tracking the animal's position, at different orientations and spatial gains. This has the effect of defining an animal's position with greater granularity. For example, in a single 5 x 6 grid, there are 30 potential cells that can represent the animal's current position. Adding a second grid at a different orientation means that the animal's position can be represented as the combination of the positions of the two grids, giving 900 potential positions that can be represented.

As explained in Chapter 4, the other model parameter that governs the number of grids is the spatial gain. In rats, the gain increases in discrete steps by a factor of approximately 1.42, moving from dorsal to ventral locations in MEC. The spatial resolution drops at each step increase in the gain, meaning that for a grid with a higher gain, the animal has to move further physically before the maximum activation of the grid changes. The number of gains is the specific parameter that was varied in the scenario sets, with a range of one, two, or three. The number of orientations was fixed at two, and the combination of orientation and gain gives a total number of grids as two, four, or six. The number of unique positions that can potentially be represented is 30^{grids}.

The effect of increasing the number of grids is both to increase the granularity of the position signal sent to the PlaceSystem and to increase the size of the PlaceSystem itself. The auto-associative memory of the PlaceSystem is a square matrix, whose dimension is calculated directly from the length of the MEC output vector (plus the length of the LEC output vector, nLEC). Two grids generate an MEC output vector of length 60, and a PlaceSystem of size $(60 + nLEC)^2$, while four grids generate an MEC output vector of length 120, and a PlaceSystem of size $(120 + nLEC)^2$. An example output from two grids could be [22 36], representing maximum activation of 22 for the first grid, and 6 for the second grid. An example output from four grids could be [22 36 89 100], adding a maximum activation of 29 for the third grid and 10 for the fourth grid.

At a high level, we would expect two areas of impact from changing the number of grids. First, the number of grids would affect the saturation level of the PlaceSystem. This effect is a combination of two factors moving in opposite directions, so the expected direction of the effect is not immediately obvious. An increase in grids increases the size of the PlaceSystem, which will lower the level of saturation. But each increase in grids also increases the number of synapses created for each new place by at least one (see Section 4.5, and the discussion of sparse Place IDs, below), so increases the level of saturation. Secondly, given that any attempt to retrace a previous path will be subject to some error, an increase in granularity might decrease the likelihood of exactly retracing a previous path or recalling a previously visited place. There is no obvious reason to expect that the number of grids would have an impact on the efficiency of exploration.

5.2.3 Head direction cells

The number of head direction cells controls the resolution with which the animal's head direction can be represented by the HeadDirectionSystem. This parameter is given two

values in the scenario sets, 30 and 60. In the original implementation from Zilli (2012), the number of head direction cells is 60. This makes it simple to think of the angular interval represented by a single cell as the size of a minute on a clock.[1] At this level of angular granularity, the system behaves quite reliably. At a practical level, this means that an angular velocity signal that drives the system n cells clockwise, will, with reversed sign, drive the system n cells counter-clockwise. As the number of cells falls below 30, however, the reliability of the system drops off. A unit velocity signal may result in moving one cell, or two cells, or zero cells. Rather than tune the functions that drive the system, I chose 30 as the practical lower limit on HeadDirectionSystem granularity. To ensure that the physical motion is consistent with the HeadDirectionSystem, the unit of angular motion is set to $\frac{2pi}{headDirectionCells}$. This means that the animal turns further during each unit turn when the number of head direction cells is 30 than when it is 60. As exploration is random, and one direction is as good as another, no attempt was made to analyze whether this has other consequences for the overall behavior of the system, although some ad hoc observations are offered in Chapter 6.

The number of head direction cells determines the length of the LEC output vector, which is therefore a binary vector of length 30 or 60, times the number of cues, minus one: $headDirectionCells(cues - 1)$. As discussed in Chapter 4, this represents the canonical view of the cues. The HeadDirectionSystem is adjusted as if the animal was pointing directly at the most salient cue, and one vector segment is given for the head direction offset for each cue other than the primary cue. For the scenario sets, the number of cues was two, so one vector segment is sufficient to represent the canonical view. This means that the length of the LEC output vector is just the number of head direction cells.

Just as for the number of grids, an increase in the number of head direction cells both increases the granularity of the position signal sent to the PlaceSystem, and increases the

[1] An old-style analog clock, with a circular face. (The digital age generates new metaphors, but makes old ones obsolete.)

size of the PlaceSystem. The size of the PlaceSystem is the sum of the lengths of the LEC and MEC output vectors, squared, so either $(30 + nMEC)^2$ or $(60 + nMEC)^2$.

Although the difference in granularity between 30 and 60 head direction cells is much lower than the difference in granularity between two, four, or six grids, we would expect analogous effects from a rise in the number of head direction cells as from a rise in the number of grids. That is, we would expect that the number of head direction cells would affect the saturation rate of the PlaceSystem, but in this case the effect should be negative. More head direction cells increases the size of the LEC output vector, and therefore increases the size of the PlaceSystem, lowering the saturation level. On the other hand, the canonical view is still only the angle to one cue, so the number of synapses required does not change, so the net effect on saturation level should be negative. Secondly, we might expect that an increase in head direction cells, and therefore granularity, might decrease the likelihood of exactly retracing a previous path or recalling a previously visited place. Just as for the number of grids, the number of head direction cells has no obvious impact on the efficiency of exploration.

5.2.4 Sparse Place IDs

The default representation of a novel place in the PlaceSystem requires matrix positions as follows:

$$positions = (grids + (cues - 1))^2 \tag{5.1}$$

That is, for each grid and for each cue after the most salient cue, there will be one non-zero input along the modifiable synapse pathway (the combination of MEC and LEC output), and one non-zero input along the detonator synapse pathway (the output from the OrthogonalizingNetwork), resulting in the creation of a number of new synapses that is the

square of the number of inputs. As discussed in Chapter 4, to lower the rate of saturation in the PlaceSystem, a sparse Place ID parameter was introduced. Sparse Place IDs use a hash to reduce the detonator synapse outputs from the OrthogonalizingNetwork to two, one representing the inputs from the MEC, and one representing the inputs from the LEC.

$$positions = 2(grids + (cues - 1)) \tag{5.2}$$

This means that the impact of increasing the number of inputs from EC, for example, by increasing the number of grids from four to six, remains linear. If there are two cues represented by LEC, the inputs along the modifiable synapse path go from 5 to 7, and the new synapses created go from $2 * 5 = 10$ to $2 * 7 = 14$.

We would expect that sparse Place IDs would have a negative effect on saturation levels. It's not obvious why sparse Place IDs should have an effect on either the efficiency of exploration or the ability of the system to retrace a previous path.

5.3 Model execution

In this section, details on how the model was executed are reviewed.

5.3.1 Time steps

Each scenario runs for 3000 steps, which means some combination of random unit turn or run motions, in either physical or simulated mode. An individual motion is of distance one to five units, and each unit, whether turn or run, is decremented from the 3000 total. Each scenario includes an average of about 10% more steps than 3000, because some steps are not counted towards the total. Steps that are not counted are those taken as the animal

is about to exit from simulated motion, to reverse any net simulated turns and return the animal's HeadDirectionSystem activation to approximately what it was upon entry to simulated motion. Also not counted are steps taken to physically retrace the animal's first simulated run upon exit from simulated motion.

The exact number of steps in a given scenario is not relevant to most of the results analysis, as the data were cut off for each scenario when the animal reached a threshold percentage in exploring the arena.

5.3.2 Initial conditions

For each scenario, the animal was placed at the center of the arena. This is point (1,1) in a square arena two units on a side, positioned in the first quadrant. The animal's physical head is pointed at zero on the unit circle ("east"). The HeadDirectionSystem and GridChartNetworks initial activation is set randomly, driven by the Matlab random number generator, whose initial seed is set randomly, but repeatably, for each scenario. No attempt was made to ascertain whether placing the animal initially at random locations in the arena would materially affect the results.

5.4 Log files

In this section, the information that was gathered from model execution log files is explained. These files were the source for the data used in the subsequent analysis.

Each scenario generates two files, whose names reflect the time the scenario began, and the file contents. The Matlab output, called the diary file, is a log of diagnostic information from both Matlab and the PetriNetRunner, useful for detailed trouble-shooting and tracing

of system behavior. The diary file was also used as the data source for Figure 6.1. The steps file is in csv (comma-separated values) format, containing one record per time step for the scenario. This is the source of the data that is analyzed below. The fields are as follows:

- *seed*: the random seed that was used to populate the Matlab random number generator, and indirectly, the Java random number generator for the PetriNetRunner
- *step*: the time step, as measured by the animal after each unit turn or run[2]
- *place ID*: the Place ID generated by the PlaceSystem after the animal's physical or simulated motion in this step
- *simulated*: one if the animal is in simulated motion during this time step; zero if the animal is moving physically
- *turn/run*: one if the animal turned in this time step; two if the animal ran
- *placeRecognized*: one if the animal recognized this place when it completed its motion; zero if the place was novel
- *retracedTrajectory*: one if the animal was physically retracing the first simulated run of its most recent period in simulated mode (retracing includes all simulated turns up to the first simulated run); zero otherwise.
- *successfulRetrace*: one if the Place ID generated by this time step is an exact superset of the corresponding Place ID in the simulated run that the animal is retracing. (If successful, the physical Place ID will be a superset of the simulated Place ID because the physical Place ID includes the LEC portion of the Place ID, while the simulated Place ID does not.)

[2]this is not the same time step as is measured by the scenario, as the animal tracks all steps/turns, regardless of whether they are counted toward the scenario total, as discussed above. Consequently, the number of steps taken by the animal is higher than the number of time steps defined for the scenario, on average, by about 10%. This is a confusing discrepancy, as discussed briefly in Appendix A.

– *gridSquarePercent*: the percentage of the overall arena that the animal has visited through either physical or simulated motion. The arena is divided into 100 equal squares, and visiting a square means that the animal's physical or simulated position falls within the square. The number of visited squares is totaled, then divided by 100 to give the gridSquarePercent.

– *saturationPercent*: the number of matrix points in the PlaceSystem with a value of one, divided by the total number of matrix points

– *sparsePlaceId*: the value of the sparse Place ID parameter for this scenario; either one or zero

– *ripples*: the value of the ripples parameter for this scenario; either two, four, or six

– *grids*: the value of the grids parameter for this scenario; either two, four, or six

– *headDirectionCells*: the value of the head direction cells parameter for this scenario; either 30 or 60

– *tag*: the git source control commit tag for the Matlab code and Petri nets for this scenario; these results were generated from tag "v1.0.7"[3]

– *pipeTag*: the git source control commit tag for the PIPECore Java code used by the PetriNetRunner; these results were generated from tag "pipe-core–2.0.0"[4]

[3] `https://github.com/sjdayday/simulated-motion/releases/tag/v1.0.7`
[4] `https://github.com/sarahtattersall/PIPECore/releases/tag/pipe-core--2.0.0`

Chapter 6

Results

6.1 Overview

In this chapter I analyze the results of running SMS over the parameter space defined in Chapter 5, to draw two conclusions about the behavior of the system and its implications for the hypotheses of this work. There is also a conclusion that reflects work in progress on the internal behavior of the system, to guide its future evolution. I begin with a comparison of the navigation behavior of the animal under SMS to the behavior of actual rats from Pfeiffer and Foster (2013). This is followed by the statistical analyses that support the three conclusions.

The results support the central hypothesis of this work, that simulated motion provides a plausible mechanism by which some trajectories in the hippocampus could be generated. Specifically, upon exiting a quiet waking period, the animal is able to physically retrace the trajectory of its first simulated run from the previous period, at a level significantly greater than chance. To the observer, a trajectory in the hippocampus predicts the animal's

subsequent physical motion. A corollary of this account is the suggestion of an exogenous source for the activation of populations of cells in the hippocampus to represent novel places.

The results support a second conclusion, about the function of simulated motion for the animal. One possible value of simulated motion might be to enable the animal to explore the environment at a lower energy cost than the cost required to physically navigate everywhere in the environment. Simulated motion provides less information than physical motion, due to the lack of sensory input, but the information provided has some value. The results are consistent with the suggestion that one possible value of simulated motion is greater efficiency in exploring the environment. In discussing future work in Chapter 7, I will offer an additional suggestion about the utility of simulated motion.

Finally, the results are consistent with the earlier discussion (Chapter 4) of factors that affect the level of saturation within the PlaceSystem. The results confirm the impact of sparse Place IDs on lowering the saturation rate of the PlaceSystem. This provides additional confidence in the overall behavior of the system, by demonstrating that its actual behavior is consistent with what we would expect from its design. Nevertheless, although sparse Place IDs reduce the saturation rate, the saturation rate continues to rise as the animal explores. This motivates future work, to incorporate fixing position ("settling") as a next evolution in the system. This should limit the steady creation of new Place IDs and thereby avoid the eventual exhaustion of PlaceSystem capacity.

The function of the statistical analyses that follow is to give numerical characterizations of the behavior of the SMS model, and to act as a high-level sanity check on the implementation of the model. One objective of a quantitative assessment is to increase our overall confidence that the system works roughly as intended. A second objective is to enable us to assess whether the behavior of the system moves in predictable directions when we vary important parameters. Unlike many statistical analyses, which attempt to estimate some parameters that are important to the questions under examination, in this case the system being ex-

amined is just software, and has no unknown parameters that require estimation. Although in principle a software system can be examined in unlimited detail, as a practical matter we rely primarily on testing to verify system behavior (see Appendix A). Once the system appears to be running more or less correctly, it is then useful to use a statistical analysis to assess whether its behavior seems to be in the right ballpark. I am therefore less concerned about the specific values that are estimated for the impact of model parameters, than I am to assess their signs and relative magnitude, in light of what the logic of the system would lead us to expect. The technique used is ordinary least squares linear regression, which is sufficient to analyze signs and rough magnitudes. If precision in parameter estimation were important, I would also use non-linear regression (for example, see Figure 6.4 for a case where a non-linear model should be more accurate), but a linear model suffices for these purposes.

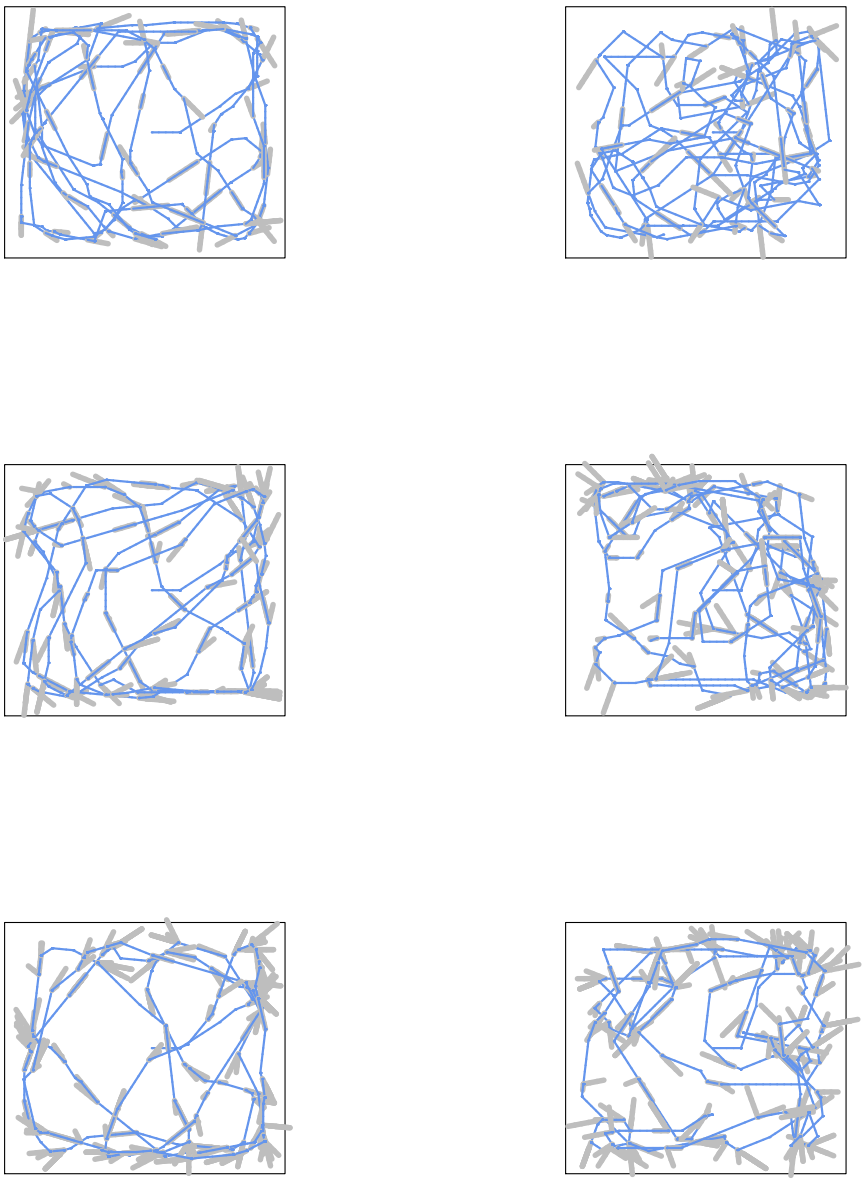


Figure 6.1: SMS behavior. Top to bottom: two, four, six ripples. Left to right: 60, 30 head direction cells. Other parameters: Sparse Place IDs, six grids. Gray lines depict simulated motion; blue lines depict physical motion. Black square: arena walls.

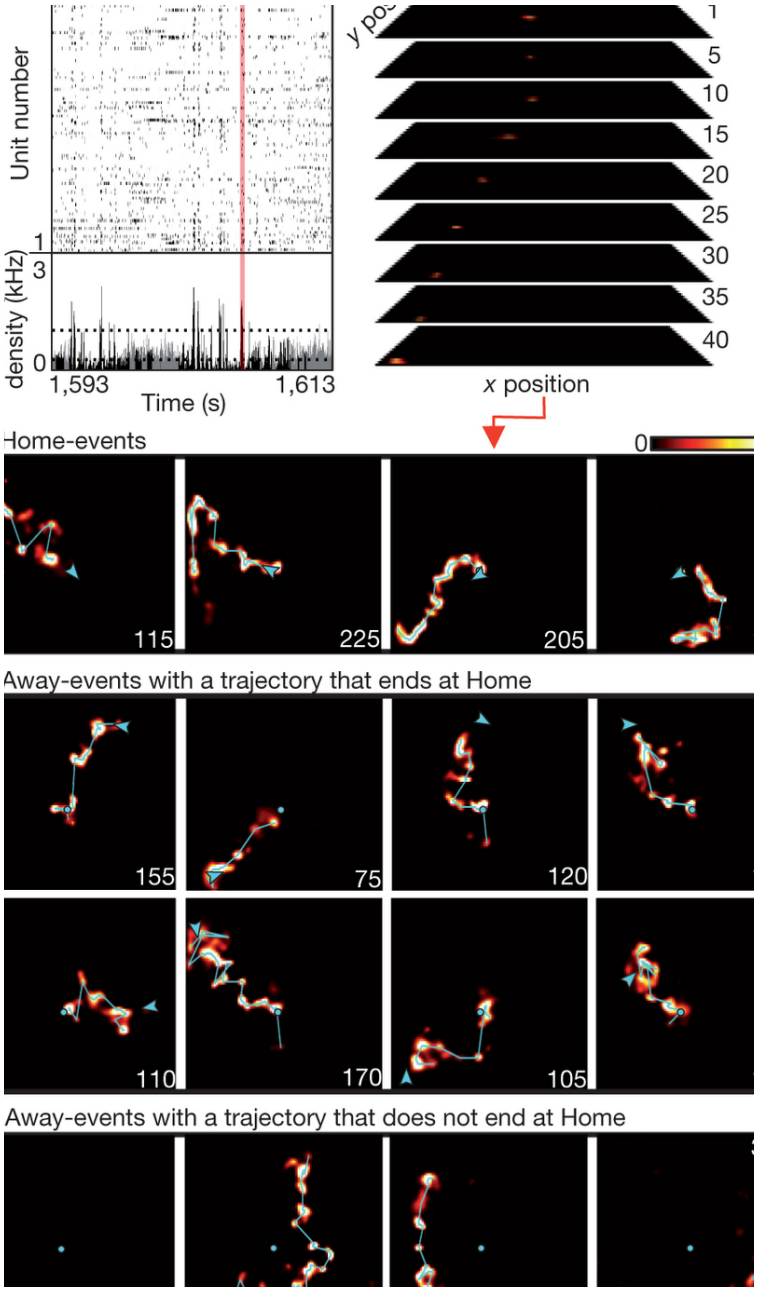


Figure 6.2: Trajectory events: **a**, Raster plot (top) and spike density (bottom) of simultaneous unit activity for R1,D1 for representative epoch. Periods of immobility denoted in black. Dashed lines represent candidate event detection threshold. **b**, Position posterior probabilities in selected frames for the candidate event in **a**. **c**, 16 representative events (of 274) for R1,D1, decoded and summed across time. Values indicated by colour bar. Event duration (in ms) in right corner. Cyan circle, Home well. Cyan line, peak probability for each timeframe. Cyan arrowhead, position and head direction of rat at time of event. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Nature ("Hippocampal place-cell sequences depict future paths to remembered goals", Pfeiffer, B. E., and Foster, D. J.), (2013)

6.2 Model navigation behavior

In a typical SMS scenario run, the animal explores the arena randomly, with some combination of physical and simulated motion (Figure 6.1). The trajectories observed in the Pfeiffer and Foster (2013) study (Figure 6.2) differ from the SMS simulated trajectories in several ways:

- The observed trajectories may begin or end at the animal's current location, or the entire trajectory may be remote from the current location. In contrast, the SMS simulated trajectories always begin at the animal's current location.
- The observed trajectories frequently do not proceed in a straight line, but seem to consist of multiple segments that may be at angles to one another. In contrast, the SMS simulated trajectories consist of straight runs.
- The observed trajectories may begin at an angle different from the animal's current head direction. In contrast, the SMS simulated trajectories always begin from the animal's current head direction, although the first simulated motion may consist of a turn. This has the effect of making subsequent runs proceed at an angle different from the current head direction; such trajectories can resemble the observed trajectories.

Focusing more closely on the trajectories generated by SMS in Figure 6.1, I note the following:

- As one would expect, the physical trajectories (blue lines) are confined within the walls of the arena. The simulated trajectories (gray lines) sometimes extend beyond the walls of the arena. As explained in Chapter 4, this is because simulated motion proceeds without sensory input. In SMS, sensory input is primarily modeled by the LecSystem, which processes visual input in the form of angles to salient cues. During simulated motion, output from LecSystem is suppressed, so only the self-motion information from

the GridChartNetworks is input to the PlaceSystem. The other source of sensory input is the animal's whiskers, which detect the proximity of walls. In the absence of physical motion, there is no signal from the animal's whiskers. Due to the absence of sensory input from either of these sources, the animal's trajectories are not blocked by the arena walls. This pattern of "exploring" locations that are not physically accessible to the animal is consistent with various experimental findings ((Ólafsdóttir et al., 2015; Dragoi & Tonegawa, 2011, 2013).

- In a number of cases, under all the parameter settings, the physical trajectory appears to closely retrace the simulated trajectory (blue line overlays the gray line). This will be analyzed in detail below.

- Overall, the combination of physical and simulated trajectories appears to cover the arena fairly uniformly. There is a difference between the left and right columns in that the angles at which connected trajectories proceed are, on average, more acute in the right-hand column. This is a consequence of the head direction cell parameter. As discussed in Chapter 5, when head direction cells is 30, the animal turns through twice the angular distance as when head direction cells is 60. The effect of this is not directly analyzed further, but to the eye, the more gradual turns in the left column seem to enable the animal to more closely follow the walls of the arena, which may enable it to more efficiently explore the arena. In a number of cases in the right column upon approaching a wall the animal turns abruptly away, which may tend to make it take longer to complete its exploration. Roughly, the left column seems to show more exploration along the walls, while the right column shows more exploration in the interior of the arena.

Having made some characterization of SMS physical and simulated behavior, I turn now to the statistical analyses.

6.3 Analysis of retraced trajectories

The central neuroscientific hypothesis of this work is that simulated motion provides a plausible mechanism by which some trajectories in the hippocampus could be generated. During quiet waking periods, the trajectory data from Pfeiffer and Foster (2013) suggest that the animal is simulating future motion toward other places in the environment. I have suggested that the mechanism for this could begin with unspecified processes in the neocortex that generate a plan to move in some direction. This triggers execution of a corresponding motor routine to turn or run in simulated mode, i.e., without sending a signal to the motor neurons. Efferent copy of this simulated activity in the form of angular and linear velocity signals continues to flow to the entorhinal cortex. The angular and linear velocity signals cause the grid cell networks in the medial entorhinal cortex (MEC) to update. The processing of multi-dimensional inputs to the lateral entorhinal cortex (LEC) includes both spatial and non-spatial information, but there has been little study of its activity during quiet waking periods, so I have not modeled a contribution from LEC to hippocampal processing.

The updated grid cell networks in MEC send updates to the hippocampus, both directly to CA3, and indirectly through the dentate gyrus, causing populations in CA3 to be activated. As the simulated motion continues, multiple populations will activate in turn in the hippocampus. This would manifest itself to observers of hippocampal activity as a trajectory across the environment. The elapsed time of the trajectory could be much accelerated by the suppression of the feedback loop through the motor neurons to the animal's musculo-skeletal system and back. Under simulation, the same motor routine could execute at a much faster speed, generating the rapid trajectories we observe in the hippocampus. If we observed simultaneously in the MEC, we would also expect to see similar rapid trajectories through MEC, as in the findings of Yamamoto and Tonegawa (2017).

The simulated motion system (SMS) is a computational model of these processes. Making no assumptions about how and where the motor plans originate, the MotorCortex generates the parameters for random motions. The Navigate Petri net include hierarchy models the execution of these motor plans as turns or runs, as well as the transition into and out of simulated mode.[1] At each physical or simulated motion step, the angular or linear velocity is sent to the GridChartNetworks, HeadDirectionSystem, and LecSystem. The GridChartNetworks update their activation and send the updated information to the PlaceSystem. As I don't make an assumption about LEC processing during simulated motion, the LecSystem only sends information to the PlaceSystem during physical motion, and not during simulated motion. Based on the inputs from the GridChartNetworks and the inputs from the LecSystem (during physical motion only), the PlaceSystem generates an updated Place ID. A sequence of these updated Place IDs corresponding to the steps of a simulated run constitutes the equivalent of a trajectory during a quiet waking period.

In the Pfeiffer and Foster experiment, the trajectory data were gathered only after the animal went through prolonged training in the environment, such that all places in the environment were previously known. Other experiments, however, suggest that place fields may be allocated during quiet waking periods for physical locations prior to the animal having an opportunity to explore those locations (Dragoi & Tonegawa, 2011; Ólafsdóttir et al., 2015). When later given the opportunity to physically explore those locations, the previously activated place fields are re-activated. This is consistent with the simulated motion hypothesis.

The SMS scenarios model the process of an animal randomly navigating in a novel arena. As the animal rests periodically, it simulates motion to random locations, some of which

[1]The Petri nets do not explicitly model time; their logic is unaffected by the speed at which they execute. Their actual elapsed execution time is a function of the speed at which the Matlab components, representing the animal's other cortical components, respond after each unit of motion. Although physical and simulated motion in SMS do not currently vary significantly in elapsed time, additional delay to model physical motion could be introduced into the Petri nets without much altering the logic of the system.

are novel. These novel locations will be represented by new Place IDs in the PlaceSystem. Previously visited locations will be represented by updated Place IDs (although, as discussed previously and below, due to calibration errors, new Place IDs are often created). Upon exiting simulation mode, the animal's first physical motion is to attempt to retrace the first run (and any preceding turns) of the just-completed simulation period. If as part of this physical motion the animal re-activates the same Place IDs as during its simulated motion, the system has demonstrated behavior equivalent to what has been observed experimentally for novel places (Dragoi & Tonegawa, 2011; Ólafsdóttir et al., 2015). Regardless of whether a retraced place is novel or previously known, this behavior is also consistent with the Pfeiffer and Foster (2013) results.

Verifying that the system shows the ability to retrace trajectories is the simplest way to demonstrate behavior that is consistent with the simulated motion hypothesis. A fully-functioning system should also be able to demonstrate a number of other behaviors, as detailed in Appendix D. A subset of these behaviors includes:

- Upon placement back into a previously explored environment with unchanged cues, the animal orients, successfully retrieving a Place ID corresponding to the physical placement location, and the GridChartNetwork and LecSystem activations that correspond to the recalled Place ID.
- When in close physical proximity to a previously visited location, the animal shows sizeable and stable place fields.
- When the cues or the shape of a known environment are manipulated, the animal's place fields change shape or location in predictable ways.

Although some of these behaviors can be demonstrated by SMS in isolation (see Appendix A and D), they have not been integrated into the full system. The most immediate problem

is that the system drifts out of calibration quickly, so that the ability to recall places that were visited many time steps in the past is limited.[2] As currently implemented, however, this is not a problem for immediately retracing a trajectory, as drifting out of calibration is not a practical issue on such short time scales. Therefore, I will focus only on analyzing the ability to retrace trajectories, as demonstration of the support of SMS for the simulated motion hypothesis.

6.3.1 Results for retraced trajectories

Over the 720 scenario runs, the system successfully retraces potential trajectories an average of 63% of the time (s.d. 11.5%). Specifically, for the steps of the physical runs that immediately followed exiting simulation mode, 63% of the time the Place ID generated was an exact superset of the Place ID generated during the corresponding step of the first simulated run of the immediately preceding quiet waking period. As the Place ID in a simulated run is just generated by MEC output (not LEC output), an "exact superset" means that the cells of the GridChartNetworks with maximum activation are the same during this step of the retraced run as they were during the corresponding step of the simulated run. In addition, the Place ID of the physical run will have one component that reflects LEC input. For example, the Place ID from one simulated run step was [26 89 127 142 160 205], and during the attempted physical retrace, the Place ID was [26 89 127 142 160 205 *216*], where *216* reflects the additional LEC input.

This result reflects the accuracy of the model in attempting to retrace simulated trajectories. To test whether this is better than the results from random navigation, SMS logic was changed such that upon exiting the quiet awake state, the animal immediately resumes random navigation.[3] Over the same parameter space as the previous scenario runs, 72 sce-

[2]The implementation of a mechanism for fixing position (settling) should remove this limitation.

[3]`https://github.com/sjdayday/simulated-motion/releases/tag/v1.0.6-shuffle`

nario runs were executed to determine the average successful retrace percentage. Under the random navigation scenarios, the animal successfully physically retraced the simulated trajectories an average of 25% of the time (s.d. 11.1%). Even under random navigation the animal successfully retraces the simulated trajectories a quarter of the time, because simulated trajectories start from the animal's current position and head direction, and random navigation will begin with a run from the current position about half the time. The distance of the random run will likely vary from the distance of a previous simulated run, but at least the first step of a first run will closely match a previous run. Despite these areas of overlap, the deliberate attempt to retrace a simulated trajectory is significantly different from the random navigation (P-value: $2.2X10^{-16}$, Wilcoxon rank-sum test).

This result shows that most of the time SMS clearly behaves in a way consistent with the simulated motion hypothesis.[4] This is the primary result of this work. We can increase our understanding of this result by using ordinary least squares linear regression to analyze the separate impacts of the four parameters that were varied across the scenario sets, upon the successful retrace percentage. The equation to be estimated is:

$$successfulPercent_i = \beta_0 + \beta_1 sparsePlaceId_i + \beta_2 ripples_i + \beta_3 grids_i + \beta_4 headDirectionCells_i + u_i \quad (6.1)$$

From the logic of the model, we would not expect sparsePlaceId to have an impact on successful retrace percentage. For example, if MEC inputs are [10 34 71 103] and LEC inputs are [23] then the generated Place ID, without sparse Place IDs, would be five numbers (e.g.,

[4]The percentage of the time that SMS fails to successfully retrace the exact steps of the preceding simulated motion can be attributed to small differences that accumulate in the floating point matrices that represent the activation of the grid chart networks. The maximum value of each network is converted to a single integer value in [1,30], but the underlying floating point values change through the activation function from step to step and rarely repeat.

[2 38 44 79 139], where four numbers are the result of the MEC input, and one number is the result of the LEC input). With sparse Place IDs, the generated Place ID would be two digits (e.g., [56 94], where both MEC and LEC inputs are hashed, and one number for each input is the result). Regardless of the length of the Place ID output, the retrace will not be successful unless the MEC inputs ([10 34 71 103]) are successfully re-created at the appropriate step. If the MEC inputs are matched, then the Place ID outputs will match, regardless of whether they are in sparse Place ID format or not. In other words, the key question is the ability of the system to re-create the animal's position on the GridChartNetworks, not the format of the Place IDs.

We would expect ripples to have some negative impact on successful retrace percentage. The logic of the Navigate Petri net causes one of the ripples during the rest period to be "consumed" immediately upon entry into simulation mode. This means that when ripples = 2, there is exactly one turn or one run that is executed during one period of simulation. As turns do not change the animal's Place ID, they are ignored for purposes of calculating the successful retrace percentage. So, at low levels of ripples, retracing the animal's path only involves a single run. As the number of ripples increases, however, some simulated runs will be preceded by simulated turns. Also, at the conclusion of the simulation period, there is a turn that acts as a net adjustment to the animal's head direction, to return it to its approximate value as of the entry into simulation. These turns introduce some error into the animal's head direction system. When the animal later attempts to physically retrace the same path, any head direction error that has accumulated through multiple turns has some probability of affecting the exact direction in which the run takes place. This may cause the maximum activation of one or more of the GridChartNetworks to differ from its previous activation. The differences are usually minor; e.g., if there are four GridChartNetworks, the activations of two or three of them are likely to match, but one error will cause the generated Place ID to be different and the retrace to be unsuccessful. For a multiple step trajectory, it

is frequently the case that some steps will successfully retrace and others will fail, indicating these types of small mismatches.

We would expect the number of grids to have a sizeable negative impact on successful retrace percentage. As explained above, each increase in grid gain adds two grids, and greatly increases the granularity of the place representation. Any small errors in the system will therefore have an increased probability of leading to a discrepancy in maximum activation in one or more GridChartNetworks, causing a different Place ID to be generated, and the retrace attempt to be unsuccessful.

Similarly, we would expect the number of head direction cells to have a negative impact on successful retrace percentage, for the same reason – an increase in granularity as the number of head direction cells increases. The effect should be more modest than the impact of the grids, as the difference in granularity between 30 and 60 head direction cells is only a factor of 2, while the difference in granularity between two grids and six grids is several orders of magnitude.

Table 6.1 shows the estimated equation results for three regressions. The first regression estimates the population model, from Equation 6.1. As expected, the effect of the number of grids is negative, and sizeable; each additional grid lowers the successful percentage rate by about 5%. However, the magnitude and sign of the other coefficients are surprising. Although the impact of head direction cells is smaller in absolute magnitude, its impact is positive rather than negative. The impact of ripples is also positive rather than negative, and on the order of 1% per ripple. The most surprising result is the impact of sparsePlaceId, which was expected to be neutral, but is very strongly positive – on the order of 8%. Finally, the intercept is significant but has no interpretation in any of the regressions in these results, as, with the exception of sparse Place ID, zero is outside of the range of the parameters.

Table 6.1: Regression of Successful Retrace Percentage on Model Parameters

Regressor	(1)	(2)	(3)
sparsePlaceId	0.079***	0.078***	0.028***
	(0.004)	(0.004)	(0.006)
ripples	0.014***	-0.019***	-0.024***
	(0.001)	(0.003)	(0.003)
grids	-0.054***	-0.053***	-0.045***
	(0.001)	(0.001)	(0.001)
headDirectionCells	0.0003*	0.001***	-0.0002
	(0.0001)	(0.0002)	(0.0002)
steps		0.0002***	0.0002***
		(0.00002)	(0.00002)
saturationPercent			-0.654***
			(0.057)
intercept	0.732***	0.095*	0.319***
	(0.01)	(0.049)	(0.049)
Summary statistics			
SER	0.059	0.053	0.049
$\bar{R}^2$	0.737	0.789	0.821
n	720	720	720
*** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$			

These results first suggest possible logic errors in the model. After various double-checks against the logs and the program codes, no obvious logic errors were found. Two additional regressions were run to address possible omitted variable bias. As mentioned earlier, the number of time steps in a given scenario can vary from one run to the next. It is not obvious why this would be correlated with the successful retrace percentage, but adding steps to the model for the second regression leads to results that are more consistent with the expectations. The steps coefficient is significant, and has a sizeable impact; the addition of 300 more steps (10% more than the scenario definition) results in an increase in the successful retrace percentage of about 6%. As a side effect, the addition of steps causes the estimated coefficient for ripples to have its expected negative sign, and an impact of about -2%. The coefficient for head direction cells continues to be small and positive, and the coefficient for sparse Place ID continues to be strongly positive.

The addition of saturation percentage to the model in the third regression clarifies the situation. Saturation percentage is the percent of the auto-associative network in the Place System that is consumed, i.e., the percentage of the matrix points where the value is one. The mean saturation percent at the end of each of the 720 scenario runs was 22% (s.d. 6.7%). The coefficient for saturation percentage is negative and important, suggesting that if saturation rises by 10%, the successful retrace percentage will drop about 6.5%. The addition of saturation percent drops the size of the impact of sparse Place ID to about 2.8%, but also leads us to a suggestion as to why it continues to have a sizeable positive impact. The impact of head direction cells is now negative, as predicted, but very small and not significant.

As discussed at length earlier, and again later in these results, as the saturation percentage rises, errors in retrieval from the PlaceSystem will become more frequent. Managing the saturation percentage in the PlaceSystem is therefore a significant concern for the performance of the system across a number of dimensions. An average saturation rate of 22% suggests that the system is well on its way to saturation; as a rule of thumb, a 50% saturation level is unworkable, generating a very high error rate. Saturation in the PlaceSystem is only an indirect contributor to errors in retracing trajectories, however, because retracing trajectories is about writing to the Place System, not reading from it. To successfully retrace a step, the outputs from the MEC that are written to the Place System must be identical. But as the PlaceSystem becomes saturated, there are consequences for the downstream networks that receive its outputs, i.e., the GridChartNetworks and LecSystem. These components also have networks, which associate Place IDs to activation levels. These networks are continuous, and can accommodate more information than the binary network that supports the PlaceSystem, but errors in these networks do accumulate with utilization. The impact is felt during the "settling" process, when the animal finishes its simulated motions, and attempts to return to its prior activation, corresponding to its current physical position. That process will be subject to errors, which will accumulate in the network of each GridChartNetwork,

leading to a lower rate of success in retrace attempts. This also explains why the coefficient for sparse Place IDs continues to be important and positive; sparse Place IDs lower saturation in the Place System, but also lead to lower utilization in the downstream components, the GridChartNetworks and LecSystem.

A question that the model does not address is the tradeoff between larger and smaller levels of granularity. Greater granularity lowers the likelihood of successfully retracing trajectories, which might suggest that fewer grids is preferable. But fewer grids means greater ambiguity in the GridChartNetworks; multiple physical locations might generate the same MEC output, leading to an inability to determine which of multiple previously visited locations corresponds to the current physical location. An inability to unambiguously recall previously visited places has obvious disadvantages. At present, SMS does not allow us to model the advantages of granularity; this is another opportunity for future work. The internal behavior of the model in the area of network utilization in the PlaceSystem, GridChartNetworks, and LecSystem has a significant impact on the model's ability to successfully retrace trajectories. The internal behavior of PlaceSystem memory utilization will be explored further below. The data that were gathered in the scenario runs did not include metrics on the continuous networks of the GridChartNetworks and LecSystem, so additional detail on their performance is not available. The analysis suggests that performance of the model will benefit from improving its ability to manage saturation levels; this will be further confirmed by the saturation analysis below.

In summary, the analysis of successful retrace percentage is consistent with what was expected, once we include additional variables that capture the impact of utilization levels in the various memory networks. The Pfeiffer and Foster (2013) results show quiet waking trajectories that predict the animal's future motion. SMS generates similar behavior. The successful retrace percentage results show that the accuracy with which SMS can retrace trajectories is well above chance levels. Currently, SMS retraces trajectories because that is

what we observe experimentally, and because it demonstrates that the system can behave as we would expect under the simulated motion hypothesis. In the next analysis, I explore one way in which simulated motion in general might have value for the animal. In the discussion of future work in Chapter 7, I will offer a suggestion about how the model might demonstrate the specific value of retracing a simulated trajectory.

6.4 Analysis of exploration efficiency

In the previous analysis I attempt to assess the ability of the model to generate simulated trajectories that resemble those which we observe experimentally. This does not address why the animal uses simulated motion – what might be the value of these simulated trajectories? In this analysis I address how SMS supports the simple theory that simulated motion is a way to explore a novel environment at lower energy cost than by physical motion alone. The account is limited, as the model is conservative about the information available to the animal during simulated motion. As explained in Chapter 4, the inputs to the PlaceSystem during simulated motion are only from the GridChartNetworks, modeling the animal's motion through abstract space, driven by self-motion cues. The only sensory inputs that the model incorporates are the angles to visual cues from the LecSystem during physical motion. It is not obvious how to incorporate simulated visual inputs into the model, so the conservative assumption is to suppress inputs from the LecSystem during simulated motion. This means that the information that is gathered during simulated exploration of the environment consists only of the distance and orientation of a new location from the current location. It does not include how the cues will appear from the new location. It also does not include whether there is a wall that obstructs the path, as walls are detected by the animal's "whiskers" during close physical proximity. Finally, it does not include any of the other dimensions that

might characterize a location, e.g., whether it is potentially rewarding. These dimensions are outside the scope of the model, but are another area for future work.

Understanding that simulated exploration returns limited information, the presumed value to the animal is that it takes less energy to explore the environment through simulated motion than through physical motion.[5] For this analysis, I assume that each physical turn or step consumes one unit of energy, and that each simulated turn or step consumes zero units of energy. This is explicitly modeled in the Navigate Petri net, which consumes one abstract "Resources" token at each step, whose source in turn either does or does not consume "Energy" tokens.[6] Measuring the energy consumed during a single scenario run therefore consists of totaling the number of physical turns or steps taken.

Measuring exploration is done by sub-dividing the arena into a grid of 10 x 10 squares, whose sides are 0.2 units in length. Any physical or simulated motion whose position falls in a given grid square is treated as exploring that square. The number of squares explored to a given point in a scenario run, divided by 100, gives the percentage of the arena explored, termed the grid square percentage.

As each time step comprises one unit turn or step, whether physical or simulated, the exploration of the environment will be correlated with the time steps, as in Figure 6.3a. In each time step, a simulated step or turn can increase the amount of the environment explored at no additional energy cost. We will therefore expect that as the number of ripples increases, the animal explores the environment more efficiently, because the percentage of the total time steps spent in simulated turns and steps increases. We see examples of this in Figure 6.1, where the number and length of connected segments in gray increases from

[5]Simulated exploration also takes less elapsed time, as the observed hippocampal trajectories take much less time than physical trajectories. As this difference in elapsed time is not captured in the model, as discussed in the previous analysis (see Section 6.3), it is not included in the analysis.

[6]In the case of simulated motion, perhaps we might think of attention as a resource, but this isn't further modeled.

top to bottom, as the number of ripples increases. Figure 6.3b shows that the energy cost to reach a given percentage of the grid squares diminishes as the number of ripples increases.[7]

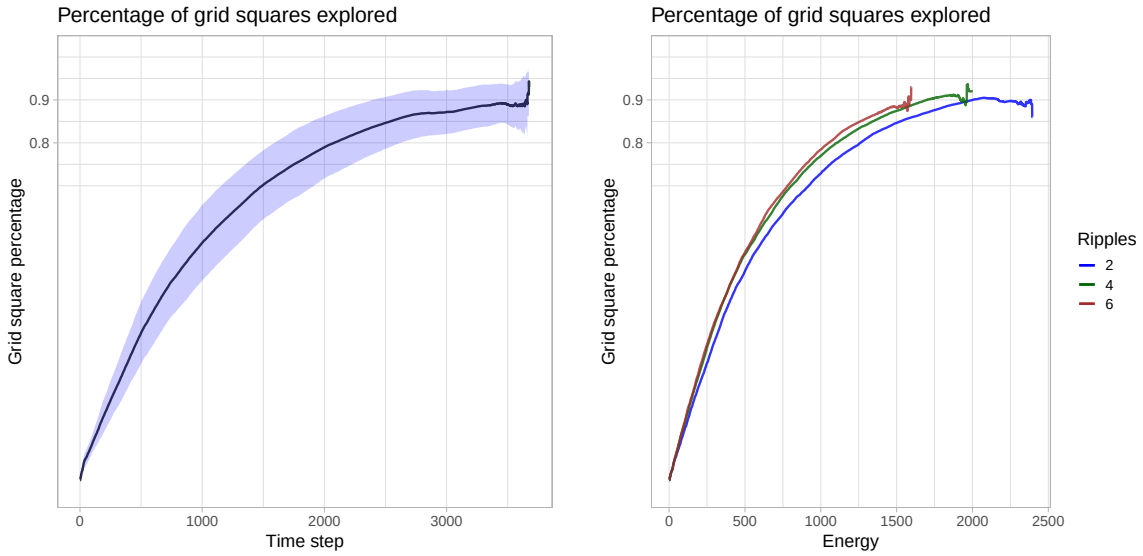


Figure 6.3: **a** (left): Percentage of grid squares explored by time steps, with standard deviation, n=720 scenarios. **b** (right): Percentage of grid squares by energy required, for each value of *ripples*, n=240 scenarios for each line.

6.4.1 Results for exploration efficiency

Exploration efficiency is the amount of energy used to explore a given percentage of the arena. We would like to estimate the effect of the model parameters on the energy required to achieve a given grid square percentage. The equation to be estimated is:

$$energy_i = \beta_0 + \beta_1 sparsePlaceId_i + \beta_2 ripples_i + \beta_3 grids_i + \beta_4 headDirectionCells_i + \beta_5 steps_i + u_i \quad (6.2)$$

[7] As ripples increases, the point at which all steps are exhausted for a given scenario happens at a lower energy level. This is because "energy" is only consumed by physical motion, but both physical and simulated motion consume the step allowance; as the number of ripples increases, the percentage of steps (energy) of the total steps allowance drops.

We would expect that the ripples parameter should have the greatest effect on energy expended. As the number of ripples increases, the number of simulated moves increases. This should increase the number of grid squares visited, at no energy cost. This should result in a sizeable negative coefficient for ripples, which is equivalent to saying that the amount of energy required to explore a given percentage of the arena should drop as the number of ripples rises, as seen in Figure 6.3b.

We would not expect the sparse Place ID parameter to affect the energy expended. The sparse Place ID parameter controls the size of Place IDs, which affects memory saturation and, as seen in the previous analysis, success in retracing trajectories. There is no obvious reason why it should affect the exploration efficiency.

We would also not expect the grids parameter to affect the energy expended. The grids parameter affects the granularity of place representations, but not the rate at which the animal can explore the environment.

Our expectation for the effect of the head direction cells parameter is analogous to the expected effect of the grids parameter, in regards to its impact on granularity. As the number of head direction cells rises, the granularity of the place representation increases. Again, this should not affect the rate at which the animal can explore the environment. There is a side effect of an increase in head direction cells, however, which might affect exploration efficiency. As explained in Chapter 5, a correspondence is maintained between a single head direction cell and a unit turn. The radians the animal moves in a physical unit turn are roughly equivalent to the fraction that a single head direction cell constitutes of the total head direction cells. For 30 head direction cells, a unit turn is $pi/15$ radians; for 60 head direction cells, a unit turn is $pi/30$ radians.

As explained in Chapter 4, turns do not change the current location, so do not directly contribute to the animal's exploration. New grid squares can only be visited when the animal

takes a step. Turns contribute indirectly to exploration by changing the animal's orientation for subsequent runs, possibly towards areas as yet unexplored. It is not obvious what the impact of this difference in angular distance per unit turn should be.[8] Figure 6.1 gives a suggestion about the effect, as discussed in Section 6.2. Although no numerical analysis was done, to the eye, it appears that when head direction cells is 60, the animal stays closer to the walls, rather than veering abruptly away from them. We might therefore expect that the effect of head direction cells would be negative; the energy required to explore the arena might drop somewhat with greater head direction cells. This effect would be approximately entirely due to the side effect of the angular distance of a unit turn, and would have no larger significance.

As the amount of exploration an animal can accomplish is presumably directly related to the number of time steps in which it explores, the model also includes the steps that were required to reach the threshold level of arena exploration. We would expect the coefficient to be positive; the more steps taken, the more energy expended.

To establish exploration thresholds, we observe that the animal explores at least 90% of the grid squares in 63.7% of the 720 scenario runs, and explores at least 80% of the grid squares in 98.3% of the scenario runs. For the runs that did not achieve the 90% threshold, the mean grid square percentage is 85.8% (s.d. 3.2%). For purposes of the analysis, I use the 90% threshold, and because a significant number of scenario runs fall short of the threshold, I redo the analysis using the 80% threshold, as almost all scenario runs reach this threshold. In Figure 6.3a, we see that the percentage of the arena explored rises with time steps. The rate of increase diminishes with time, as the portion of the arena that is still unexplored diminishes. The figure shows the 80% and 90% exploration thresholds that are used in the

[8]The idea that two turns of $pi/30$ radians cost twice as much energy as one turn of $pi/15$ radians is clearly not biologically plausible; it is an artifact of the model's construction. The different values for head direction cells were intended to enable the effects of place representation granularity to be explored. This aspect of the different values is a secondary side effect.

Table 6.2: Regression of Energy on Model Parameters and Time Steps

Regressor	(1)	(2)	(3)	(4)
sparsePlaceId	2.58	24.5**	2.09	23.6**
	(5.71)	(4.71)	(4.85)	(7.68)
ripples	-248***	-245***	-167***	-164***
	(2.33)	(2.56)	(1.87)	(2.03)
grids	1.70	-1.65	1.71	-1.16
	(1.75)	(1.44)	(1.45)	(1.65)
headDirectionCells	-2.84***	-2.15***	-2.20***	-1.59***
	(0.194)	(0.29)	(0.157)	(0.232)
steps	0.558***	0.549***	0.515***	0.503***
	(0.008)	(0.008)	(0.006)	(0.007)
saturationPercent		288**		297***
		(91.0)		(83.8)
intercept	1106***	1032***	835.2***	773.1***
	(24.6)	(33.7)	(13.9)	(22.3)
Summary statistics				
SER	76.65	76.17	63.24	62.73
$\bar{R}^2$	0.944	0.945	0.934	0.935
n	720	720	720	720
*** $p < 0.001$; ** $p < 0.01$; * $p < 0.1$; " " $p >= 1$				
Exploration threshold: 90% (1), (2); 80% (3), (4)				

analysis below. The unevenness beyond 3000 steps is an artifact of the limit on scenario execution (see Chapter 5).

Table 6.2 shows the estimated equation results for four regressions. The first and third regressions estimate the population model, from Equation 6.2. The first two regressions estimate the effects for a 90% threshold on percentage of the grid squares explored, and the last two regressions estimate the effects for an 80% threshold. The second and fourth regressions show the impact of adding saturation percentage, as was done for the previous analysis.

Overall, the results at the 90% and 80% thresholds are consistent with each other, so the fact that many scenario runs did not reach the 90% threshold does not affect the analysis.

As expected, the largest effect is from the ripples parameter, which has a strong negative effect on the energy cost. The energy cost to reach the exploration threshold decreases 8% for each additional ripple; exploring with six ripples is almost a third less expensive than exploring with two ripples. The absolute difference in the coefficients between the 90% and 80% thresholds (roughly -245 vs. -165) reflects the large number of additional time steps required to get to the higher threshold (see Figure 6.3a).

The number of time steps contributes directly to the energy cost, as predicted. On average, in roughly half the time steps the animal is moving physically, increasing its energy cost.

The saturation percent is clearly correlated with energy cost; as exploration continues, saturation rises steadily. It was added to the regression as a check against surprises, after noting its unexpected importance in the previous analysis. In this case, the coefficient of saturation percent is positive, as expected. The effect is on the order of raising the energy cost to reach the threshold of 28-30 for each 10% rise in saturation. But saturation percent has no obvious causal relationship to energy cost. As its contribution to the variation explained by the models ($\bar{R}^2$) is very small, the relationships are more clear if it is dropped from the regressions.

The sparse Place ID parameter has an effect only when saturation percent is added to the model. Its effect is modest, a rise in energy cost on the order of 1% when using sparse Place IDs. I don't have an interpretation for this effect; as in the case of saturation percent, there is no obvious causal relationship between sparse Place IDs and energy cost of exploration.

The grids parameter has no effect, as predicted. The effect of head direction cells is negative, as tentatively predicted. The impact of moving from 30 to 60 head direction cells is to decrease the energy cost to reach the threshold by roughly 2.5%. As discussed above, the change in granularity should have no effect on the efficiency of exploration. Although I have done no numerical analysis to confirm this, I speculate that this is a side effect of the

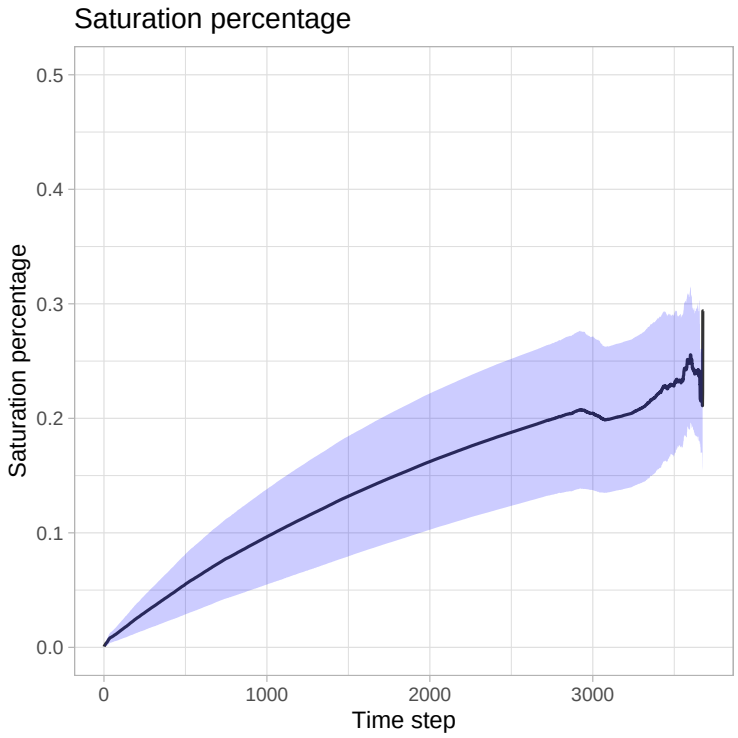


Figure 6.4: Saturation percentage in PlaceSystem by time steps, with standard deviation, n=720

difference in angular distance traversed with each unit turn. Figure 6.1 suggests that the smaller angular distance traversed with each unit turn when head direction cells is 60 may allow the animal to more thoroughly explore the area near the walls of the arena.

Overall, the analysis of exploration efficiency supports the suggestion that part of the value of simulated motion for the animal is lowering the energy cost to acquire information about the environment.

6.5 Analysis of PlaceSystem saturation

The saturation percentage is the percent of the points in the binary square matrix of the PlaceSystem whose value is one. In Figure 6.4 we see that saturation rises steadily over

time. The slope of the curve decreases steadily (convex upward) but saturation will clearly rise further for longer scenarios.[9] Although errors in retrieving Place IDs are not measured directly in these results, the error rate will rise steadily with saturation, motivating future work to stabilize the saturation level from the exploration of a single environment.

In this analysis I address the expected and actual impacts of the various model parameters on the saturation rate. The intent is to help confirm that the model is behaving as expected.

6.5.1 Results for PlaceSystem saturation

Saturation percentage is the percent of the PlaceSystem matrix points whose value is one. We would like to estimate the effect of the model parameters and time step on the saturation percentage. The equation to be estimated is:

$$saturation_i = \beta_0 + \beta_1 sparsePlaceId_i + \beta_2 ripples_i + \beta_3 grids_i + \beta_4 headDirectionCells_i + \beta_5 steps_i + u_i \quad (6.3)$$

In the analysis of retraced trajectories (see section 6.3), we had a preview of the interaction of saturation levels with the model parameters. As discussed in the earlier analysis, we would expect that sparse Place IDs would have a strong negative impact on the saturation level. That is, when sparse Place IDs are used, the level of saturation should drop.

We would expect the coefficient for head direction cells to be negative, because going from 30 to 60 head direction cells increases the size of the input vector from the LecSystem, and

[9] The changes in slope near and after time step 3000 are artifacts from the limitation of scenarios to 3000 time steps, as discussed earlier.

consequently the size of the PlaceSystem, without changing the number of matrix points activated at each step.

The impact of the grids parameter is less obvious, because increasing the number of grids both increases the size of the MEC inputs and therefore the size of the PlaceSystem, and, in the absence of sparse Place IDs, also increases the number of matrix points activated at each step. Both effects are as the square of the increase; going from four to six grids increases the size of the square matrix, and increases the number of inputs along two dimensions – directly from the MEC (the GridChartNetworks) and indirectly through DG (the OrthogonalizingNetwork).

The impact of increasing the number of ripples should be negative. From the perspective of the PlaceSystem, a Place ID is created regardless of whether it represents physical or simulated motion. But the assumption that there is no input from LecSystem during a simulated step means that the number of matrix points activated is smaller during simulated motion.

The effect of time steps should clearly be positive. As seen in Figure 6.4, saturation increases steadily over time. There is either a turn or a run at each time step, and because there is no mechanism to prevent the creation of novel places during a run (even if they are physically close to previously visited places), we would expect the steps coefficient to be positive.

Table 6.3 shows the estimated equation results for two regressions. The data used are the same as for the exploration efficiency analysis, i.e., as of the 90% and 80% thresholds in the exploration of the arena.

As expected, the effect from the sparse Place ID parameter is strongly negative, dropping the saturation percentage by about 7%. The effect of head direction cells is also negative, as expected. The size of the effect is larger than I would have guessed; moving from 30 to 60 head direction cells has almost as much effect as sparse Place IDs, dropping the saturation

Table 6.3: Regression of Saturation Percentage on Model Parameters and Time Steps

Regressor	(1)	(2)
sparsePlaceId	-0.076***	-0.072***
	(0.002)	(0.002)
ripples	-0.012***	-0.01***
	(0.0009)	(0.0008)
grids	0.012***	0.01***
	(0.0007)	(0.0006)
headDirectionCells	-0.0024***	-0.002***
	(0.00008)	(0.00007)
steps	0.00003***	0.00004***
	(0.000003)	(0.000003)
intercept	0.255***	0.209***
	(0.001)	(0.006)
Summary statistics		
SER	0.031	0.028
$\bar{R}^2$	0.779	0.785
n	720	720
*** $p < 0.001$; ** $p < 0.01$; * $p < 0.1$		
Exploration threshold: 90% (1); 80% (2)		

percentage by more than 6%. As expected, the ripples parameter also has a negative effect; each additional ripple lowers the saturation percentage by about 1%.

As expected, the coefficient for time steps is positive. Roughly, every 250-300 steps is associated with a 1% rise in the saturation percentage. Finally, the effect of the grids parameter is also positive. Each additional grid increases the saturation percentage by about 1%. The positive effect of activating additional matrix points apparently outweighs the negative effect of the increase in matrix size.

The analysis of saturation percentage suggests that the model is behaving as we would expect, under the influence of its various parameters. Despite the moderating effect of sparse Place IDs, the overall direction of the saturation percentage is steadily upward, at a pace that suggests the need for a mechanism to better manage saturation.

6.6 Results summary

Together, the statistical analyses of retraced trajectories, exploration efficiency, and PlaceSystem saturation suggest that SMS is behaving as expected. The model demonstrates a mechanism that could account for the trajectories we observe in the hippocampus during quiet waking periods that appear to simulate future motion. The model offers a simple suggestion of the possible value of simulated motion to the animal, that simulated motion provides some information about the environment to the animal at a lower energy cost. Finally, the internal behavior of the model leads to a steady increase in the saturation percentage. Although some parameters mitigate the increase, this is clearly not sustainable, and motivates future work to manage PlaceSystem network utilization.

Chapter 7

Conclusion

7.1 Summary

I have suggested that trajectories during quiet waking periods that preplay a rat's future motion are a form of simulated motion. Beginning in neocortex and propagating through the path integration mechanisms of entorhinal cortex, they are communicated to hippocampus to generate the observed trajectories. The current evidence in support of this theory isn't conclusive, particularly on the communication from MEC to hippocampus. This makes it likely that this simple story will need to be substantially modified, if not discarded altogether.

The first contribution of this work is a computational model that generates behavior similar to that observed experimentally. Relying heavily on previous computational models of hippocampus and entorhinal cortex, SMS integrates these into a system that generates simulated trajectories as part of its basic functioning. The system also generates results to support a simple account of the possible value of simulated motion to the animal, that simulation helps the animal explore its environment at lower cost than by physical motion alone.

The second contribution of this work is methodological. In the area of tools, the Petri net extensions that I have written make it possible to use Petri net modeling as part of more complex systems. SMS shows how the Petri net and X-net formalisms can be integrated with computational models from neuroscience to give a plausible overall computational account. In their visual simplicity and asynchronous execution, Petri nets are a useful way to abstract neural processing. X-nets are a natural way to model the motor routines that I suggest drive simulated action.

The other aspect of my methodological contribution is less tangible, but more fundamental. SMS is a working example of an approach to building systems that has proven very useful in many domains. SMS shows that approach at work in computational neuroscience. To borrow Complementary Learning Systems as a metaphor, too many computer programs are treated as if the problem was hippocampal "one-shot learning" – let's write one thing that demonstrates the concept, and then leave the software behind. If software folks have learned anything in the past couple of decades, it is to treat software much more like how the neocortex learns – developed continually through tiny incremental changes, driven by granular feedback at each step. Automated tests are the feedback mechanism for software, a way to make small changes, while avoiding the equivalent of catastrophic interference, where too much change too quickly causes the system to collapse. This is the only reliable way that we have come up with to increase the complexity and capability of a software system indefinitely, without it becoming so brittle and fragile that it has to be rewritten or abandoned. Although far from ideal (see Appendix A for details and shortcomings), SMS is a practical example of this approach. With the support of its small automated tests, SMS, or something simple like it, can continue to evolve, accounting for more and more findings, and continuing to generate suggestions about how the biological systems it models, might work.

7.2 Future work

The results outlined in Chapter 6 make clear that the next step in the evolution of SMS should be to implement a mechanism to limit the growth in memory utilization in the PlaceSystem, LecSystem, and the GridChartNetworks. Fixing position, or settling, as I have called it, is already implemented and tested in isolation; the next step is to integrate it into the overall functioning of the system. Although this is primarily an internal consideration, it should have one important behavioral consequence. The animal should show stable place fields of nontrivial size, rather than steadily allocating new place IDs for nearby physical locations.

Appendix D gives a detailed list of behaviors that the system currently supports, supports in isolated tests, or should be able to support in the future. These are given a very rough numerical order, reflecting my sense of how the system might best or most easily evolve to develop important capabilities. Here, I'll describe two of them in more detail.

7.2.1 Rewards

The demonstration that SMS successfully retraces simulated trajectories is missing an account of why this should happen. Currently, the animal may generate several simulated runs in a quiet waking period, but the system is hard-coded to select the first run for retracing. The purpose of this was to verify that the system has the basic capability of retracing a simulated trajectory, but like any good Hollywood actor, the simulated animal might well be asking, "so what's my motivation?"

A model is needed of a region to store memories of whether a given place is rewarding. A model of the amygdala would associate a Place ID with the presence or absence of a reward. Then, just as the models of MEC and LEC retrieve the state that has been previously

associated with a Place ID, the reward system would retrieve whether or not a place has been rewarding.

As an extension to SMS, this is only a moderate effort, but offers a simple story to demonstrate the possible value of simulated action. As suggested in Section 1.3, once a reward system is included, SMS should be able to perform low-cost "what-if" tests, without any computational support from neocortex. After exploring an environment, and determining that certain places have been rewarding, the animal could use simulated motion to decide what next to do. In simulating multiple trajectories, the animal would retrieve the rewards associated with each of the places it simulates visiting. If any of them is recalled as being possibly rewarding, the animal could select that trajectory to physically retrace.

The neocortex has multiple planning capabilities, and can infer much more complex patterns than the simple one described here. But the hippocampus is a very old structure, in evolutionary terms, and this story suggests that simulated action could be a useful way to guide future action with little or no assistance from neocortex, whose capabilities have evolved more recently. Evidence increasingly suggests that simulated action is ubiquitous in the thinking of humans, much of it unconscious and automatic. By adding a reward system, SMS would develop a capability that suggests that simulated action could be an old and basic mechanism for computation about the environment.

7.2.2 CA3 models

Adopting the model of CA3 as a binary matrix adheres to an old test-driven principle, TSTTCPW (The Simplest Thing That Could Possibly Work), and that has been productive for SMS so far. It is quite possible, however, that the lack of clarity in the evidence about the communication between MEC and hippocampus (see Chapter 2) reflects dynamics in the hippocampus that cannot be modeled by a binary matrix. The trajectories we want

to explain are just high-frequency ripples in sharp waves of relatively great magnitude that propagate through the hippocampus and well beyond. The dynamics that trigger those sharp waves aren't established, and it may be that signals from entorhinal cortex are insufficient to trigger them. In such a case, another more indirect mechanism of communication might need to be hypothesized, or the theory of simulated motion that I have proposed might need to be abandoned.

Another mechanism that I have not incorporated in SMS to date, is sub-threshold activation, where inputs to a neuron increase its activation, but not to the point where it fires immediately. For some period of time, such neurons have an increased probability of firing if additional activation is provided. Sub-threshold activation provides significant additional capability and complexity to neural processing, and it could be part of the communication story between MEC and hippocampus. As the current model of CA3 is binary – either a synapse is activated or it is not – it cannot be used to model sub-threshold activation. One modification could be to model the CA3 autoassociative network as a matrix of continuous values, where points above a certain threshold are interpreted as firing. This could enable modeling of sub-threshold activation, as well as the strengthening or forgetting of places over time.

Another modification could be to incorporate a spiking network model, which explicitly models the temporal dynamics of neurons, as well as their electrical effects. This might be required to model the generation and propagation of sharp waves. Such models are much more faithful to the complexities of neural processing, but their complexities mean that the integration into the larger system would take significantly longer. Maintaining the body of automated tests should allow such an integration to proceed in a methodical manner, preserving much of the system's current capability, while adding the new capability. Regardless of the approach, incorporating a new model of CA3 would be a major effort.

Bibliography

Bailey, D. R. (1997). *When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs* (Unpublished doctoral dissertation). UNIVERSITY of CALIFORNIA.

Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.

Bergen, B. K. (2012). *Louder than words: The new science of how the mind makes meaning*. Basic Books.

Bliss, T. V. P., & Lø mo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology*, *232*(2), 331–356.

Brandon, M. P., Bogaard, A. R., Andrews, C. M., & Hasselmo, M. E. (2012). Head direction cells in the postsubiculum do not show replay of prior waking sequences during sleep. *Hippocampus*, *22*(3), 604–618.

Brandon, M. P., Koenig, J., Leutgeb, J. K., & Leutgeb, S. (2014). New and distinct hippocampal place codes are generated in a new environment during septal inactivation. *Neuron*, *82*(4), 789–796.

Buzsáki, G. (1986). Hippocampal sharp waves: their origin and significance. *Brain research*, *398*(2), 242–252.

Buzsáki, G., Horvath, Z., Urioste, R., Hetke, J., & Wise, K. (1992). High-frequency network oscillation in the hippocampus. *Science*, *256*(5059), 1025–1027.

Buzsáki, G., & Moser, E. I. (2013). Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, *16*(2), 130–138.

Buzsáki, G., S., L. L.-W., & Vanderwolf, C. H. (1983). Cellular bases of hippocampal {EEG} in the behaving rat. *Brain Research Reviews*, *6*(2), 139–171. Retrieved from `http://www.sciencedirect.com/science/article/pii/0165017383900371` doi: http://dx.doi.org/10.1016/0165-0173(83)90037-1

Chrobak, J. J., & Buzsáki, G. (1994). Selective activation of deep layer (V-VI) retrohippocampal cortical neurons during hippocampal sharp waves in the behaving rat. *The Journal of neuroscience*, *14*(10), 6160–6170.

Chrobak, J. J., & Buzsáki, G. (1996). High-frequency oscillations in the output networks of the hippocampal–entorhinal axis of the freely behaving rat. *The Journal of Neuroscience*, *16*(9), 3056–3066.

Colgin, L. L., Denninger, T., Fyhn, M., Hafting, T., Bonnevie, T., Jensen, O., ... Moser, E. I. (2009, November). Frequency of gamma oscillations routes flow of information in the hippocampus. *Nature*, *462*(7271), 353–357. Retrieved from `http://dx.doi.org/`

10.1038/nature08573

Colgin, L. L., Leutgeb, S., Jezek, K., Leutgeb, J. K., Moser, E. I., McNaughton, B. L., & Moser, M.-B. (2010). Attractor-map versus autoassociation based attractor dynamics in the hippocampal network. *Journal of neurophysiology*, *104*(1), 35–50.

Couey, J. J., Witoelar, A., Zhang, S.-J., Zheng, K., Ye, J., Dunn, B., ... Others (2013). Recurrent inhibitory circuitry as a mechanism for grid formation. *Nature neuroscience*, *16*(3), 318–324.

Davidson, T. J., Kloosterman, F., & Wilson, M. A. (2009). Hippocampal replay of extended experience. *Neuron*, *63*(4), 497–507.

Deshmukh, S. S. (2014). Spatial and nonspatial representations in the lateral entorhinal cortex. In *Space, time and memory in the hippocampal formation* (pp. 127–152). Springer.

Diba, K., & Buzsáki, G. (2007). Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience*, *10*(10), 1241–1242.

Doubleday, S., Trott, S., & Feldman, J. (2017). Processing natural language about ongoing actions in ecg. In *2017 aaai spring symposium series.*

Dragoi, G., & Tonegawa, S. (2011, January). Preplay of future place cell sequences by hippocampal cellular assemblies. *Nature*, *469*(7330), 397–401. Retrieved from `http://dx.doi.org/10.1038/nature09633;http://www.nature.com/nature/journal/v469/n7330/abs/10.1038-nature09633-unlocked.html#supplementary-information`

Dragoi, G., & Tonegawa, S. (2013). Distinct preplay of multiple novel spatial experiences in the rat. *Proceedings of the National Academy of Sciences*, *110*(22), 9100–9105.

Eilam, D., & Golani, I. (1989). Home base behavior of rats (Rattus norvegicus) exploring a novel environment. *Behavioural brain research*, *34*(3), 199–211.

Ferbinteanu, J., & Shapiro, M. L. (2003). Prospective and retrospective memory coding in the hippocampus. *Neuron*, *40*(6), 1227–1239.

Ferbinteanu, J., Shirvalkar, P., & Shapiro, M. L. (2011). Memory modulates journey-dependent coding in the rat hippocampus. *The Journal of Neuroscience*, *31*(25), 9135–9146.

Foster, D. J., & Knierim, J. J. (2012). Sequence learning and the role of the hippocampus in rodent navigation. *Current opinion in neurobiology*, *22*(2), 294–300.

Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, *440*(7084), 680–683. doi: 10.1038/nature04587

Foster, T. C., Castro, C. A., & McNaughton, B. L. (1988). Influence of motor set on place-related hippocampal complex spike cell activity. In *Society for neuroscience abstracts* (Vol. 14, p. 396).

Frank, L. M., Brown, E. N., & Wilson, M. (2000). Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron*, *27*(1), 169–178.

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in cognitive sciences*, *9*(10), 474–480.

Fyhn, M., Hafting, T., Treves, A., Moser, M.-B., & Moser, E. I. (2007). Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, *446*(7132), 190–194.

Gothard, K. M., Skaggs, W. E., & McNaughton, B. L. (1996). Dynamics of mismatch correction in the hippocampal ensemble code for space: interaction between path integration

and environmental cues. *The Journal of neuroscience*, *16*(24), 8027–8040.

Guanella, A., Kiper, D., & Verschure, P. (2007). A model of grid cells based on a twisted torus topology. *International journal of neural systems*, *17*(04), 231–240.

Gupta, A. S. (2011). *Behavioral correlates of hippocampal neural sequences* (Unpublished doctoral dissertation). Citeseer.

Hafting, T., Fyhn, M., Bonnevie, T., Moser, M.-B., & Moser, E. I. (2008). Hippocampus-independent phase precession in entorhinal grid cells. *Nature*, *453*(7199), 1248–1252.

Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801–806.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory* (fifth ed.). New York, NY, USA: John Wiley & Sons, Inc.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554–2558.

Johnson, A., & Redish, A. D. (2007). Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. *The Journal of neuroscience*, *27*(45), 12176–12189.

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

Karlsson, M. P., & Frank, L. M. (2009). Awake replay of remote experiences in the hippocampus. *Nature neuroscience*, *12*(7), 913–918.

Ketz, N., Morkonda, S. G., & O'Reilly, R. C. (2013). Theta coordinated error-driven learning in the hippocampus. *PLoS computational biology*, *9*(6).

Knierim, J. J., Kudrimoti, H. S., & McNaughton, B. L. (1995). Place cells, head direction cells, and the learning of landmark stability. *The Journal of Neuroscience*, *15*(3), 1648–1659.

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, *20*(7), 512–534.

Lee, A. K., & Wilson, M. A. (2002). Memory of sequential experience in the hippocampus during slow wave sleep. *Neuron*, *36*(6), 1183–1194.

Leutgeb, J. K., Leutgeb, S., Moser, M.-B., & Moser, E. I. (2007). Pattern separation in the dentate gyrus and CA3 of the hippocampus. *science*, *315*(5814), 961–966.

Leutgeb, S., Leutgeb, J. K., Barnes, C. A., Moser, E. I., McNaughton, B. L., & Moser, M.-B. (2005). Independent codes for spatial and episodic memory in hippocampal neuronal ensembles. *Science*, *309*(5734), 619–623.

Leutgeb, S., Leutgeb, J. K., Moser, E. I., & Moser, M.-B. (2006). Fast rate coding in hippocampal CA3 cell ensembles. *Hippocampus*, *16*(9), 765–774.

Louie, K., & Wilson, M. A. (2001). Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron*, *29*(1), 145–156.

Maurer, A. P., VanRhoads, S. R., Sutherland, G. R., Lipa, P., & McNaughton, B. L. (2005). Self-motion and the origin of differential spatial scaling along the septo-temporal axis of the hippocampus. *Hippocampus*, *15*(7), 841–852.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, *102*(3), 419.

McNaughton, B. L., Barnes, C. A., Gerrard, J. L., Gothard, K., Jung, M. W., Knierim, J. J., ... Others (1996). Deciphering the hippocampal polyglot: the hippocampus as a path integration system. *The Journal of Experimental Biology*, *199*(1), 173–185.

McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., & Moser, M.-B. (2006). Path integration and the neural basis of the'cognitive map'. *Nature Reviews Neuroscience*, *7*(8), 663–678.

McNaughton, B. L., Chen, L., & Markus, E. J. (1991). "Dead reckoning," landmark learning, and the sense of direction: a neurophysiological and computational hypothesis. *Cognitive Neuroscience, Journal of*, *3*(2), 190–202.

McNaughton, B. L., Knierim, J. J., & Wilson, M. A. (1995). Vector encoding and the vestibular foundations of spatial cognition: Neurophysiological and computational mechanisms.

McNaughton, B. L., Leonard, B., & Chen, L. (1989). Cortical-hippocampal interactions and cognitive mapping: A hypothesis based on reintegration of the parietal and inferotemporal pathways for visual processing. *Psychobiology*, *17*(3), 230–235.

McNaughton, B. L., & Morris, R. G. M. (1987). Hippocampal synaptic enhancement and information storage within a distributed memory system. *Trends in neurosciences*, *10*(10), 408–415.

McNaughton, B. L., & Nadel, L. (1990). Hebb-Marr networks and the neurobiological representation of action in space. *Neuroscience and connectionist theory*, 1–63.

Mehta, M. R., Barnes, C. A., & McNaughton, B. L. (1997). Experience-dependent, asymmetric expansion of hippocampal place fields. *Proceedings of the National Academy of Sciences*, *94*(16), 8918–8921.

Mittelstaedt, M.-L., & Mittelstaedt, H. (1980). Homing by path integration in a mammal. *Naturwissenschaften*, *67*(11), 566–567.

Moser, E. I., Roudi, Y., Witter, M. P., Kentros, C., Bonhoeffer, T., & Moser, M.-B. (2014). Grid cells and cortical representation. *Nature Reviews Neuroscience*.

Narayanan, S. (1997). Embodiment in language understanding: Sensory-motor representations for metaphoric reasoning about event descriptions. *University of California, Berkeley: Unpublished doctoral dissertation*.

Narayanan, S. (2010). *Mind changes: A simulation semantics account of counterfactuals.*

Navratilova, Z., Giocomo, L. M., Fellous, J.-M., Hasselmo, M. E., & McNaughton, B. L. (2012). Phase precession and variable spatial scaling in a periodic attractor map model of medial entorhinal grid cells with realistic after-spike dynamics. *Hippocampus*, *22*(4), 772–789.

Navratilova, Z., Hoang, L. T., Schwindel, C. D., Tatsuno, M., & McNaughton, B. L. (2012). Experience-dependent firing rate remapping generates directional selectivity in hippocampal place cells. *Frontiers in neural circuits*, *6*.

Navratilova, Z., & McNaughton, B. L. (2014). Models of path integration in the hippocampal complex. In *Space, time and memory in the hippocampal formation* (pp. 191–224). Springer Vienna.

Norimoto, H., Matsumoto, N., Miyawaki, T., Matsuki, N., & Ikegaya, Y. (2013). Subicular activation preceding hippocampal ripples in vitro. *Scientific reports*, *3*.

O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental neurology*, *51*(1), 78–109.

O'Keefe, J., & Burgess, N. (1996). Geometric determinants of the place fields of hippocampal neurons. *Nature*, *381*(6581), 425–428.

O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, *34*(1), 171–175.

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. Clarendon Press Oxford.

O'Keefe, J., & Recce, M. L. (1993). Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, *3*(3), 317–330.

O'Keefe, J., & Speakman, A. (1987). Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research*, *68*(1), 1–27.

Ólafsdóttir, H. F., Barry, C., Saleem, A. B., Hassabis, D., & Spiers, H. J. (2015). Hippocampal place cells construct reward related sequences through unexplored space. *eLife*, *4*. doi: 10.7554/eLife.06063

Olafsdottir, H. F., Carpenter, F., & Barry, C. (2016, June). Coordinated grid and place cell replay during rest. *Nat Neurosci*, *19*(6), 792–794. Retrieved from `http://dx.doi.org/10.1038/nn.429110.1038/nn.4291http://www.nature.com/neuro/journal/v19/n6/abs/nn.4291.html#supplementary-information`

O'Reilly, K. C., Alarcon, J. M., & Ferbinteanu, J. (2014). Relative contributions of CA3 and medial entorhinal cortex to memory in rats. *Frontiers in behavioral neuroscience*, *8*. doi: http://dx.doi.org/10.3389/fnbeh.2014.00292

Ormond, J., & McNaughton, B. L. (2015). Place field expansion after focal MEC inactivations is consistent with loss of Fourier components and path integrator gain reduction. *Proceedings of the National Academy of Sciences*, *112*(13), 4116–4121. Retrieved from `http://www.pnas.org/content/112/13/4116.abstract` doi: 10.1073/pnas.1421963112

O'Neill, J., Boccara, C., Stella, F., Schönenberger, P., & Csicsvari, J. (2017). Superficial layers of the medial entorhinal cortex replay independently of the hippocampus. *Science*, *355*(6321), 184–188.

Paller, K. A. (1997). Consolidating dispersed neocortical memories: the missing link in amnesia. *Memory*, *5*(1-2), 73–88.

Peyrache, A., Lacroix, M. M., Petersen, P. C., & Buzsaki, G. (2015, April). Internally organized mechanisms of the head direction sense. *Nat Neurosci*, *18*(4), 569–575. Retrieved from `http://dx.doi.org/10.1038/nn.396810.1038/nn.3968http://www.nature.com/neuro/journal/v18/n4/abs/nn.3968.html#supplementary-information`

Pfeiffer, B. E. (2015). Personal correspondence.

Pfeiffer, B. E., & Foster, D. J. (2013). Hippocampal place-cell sequences depict future paths to remembered goals. *Nature*. Retrieved from `http://www.nature.com/nature/journal/v497/n7447/full/nature12112.html`

Pfeiffer, B. E., & Foster, D. J. (2015). Autoassociative dynamics in the generation of sequences of hippocampal place cells. *Science*, *349*(6244), 180–183. Retrieved

from http://www.sciencemag.org/content/349/6244/180.abstract doi: 10.1126/science.aaa9633

Ranck Jr, J. B. (1985). Head Direction Cells in the Deep Cell Layer of Dorsal Postsubiculum in Freely Moving Rats. *Electrical Activity of the Archicortex*, 217–220.

Redish, A. D., Elga, A. N., & Touretzky, D. S. (1996). A coupled attractor model of the rodent head direction system. *Network: Computation in Neural Systems*, *7*(4), 671–685.

Rolls, E. T. (1987). Information representation, processing and storage in the brain: analysis at the single neuron level. *The neural and molecular bases of learning*, 503–540.

Samsonovich, A., & McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of neuroscience*, *17*(15), 5900–5920.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M.-B., & Moser, E. I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, *312*(5774), 758–762.

Schacter, D. L., Addis, D. R., Hassabis, D., Martin, V. C., Spreng, R. N., & Szpunar, K. K. (2012). The future of memory: remembering, imagining, and the brain. *Neuron*, *76*(4), 677–694.

Sharp, P. E., Kubie, J. L., & Muller, R. U. (1990). Firing properties of hippocampal neurons in a visually symmetrical environment: contributions of multiple sensory cues and mnemonic processes. *The Journal of Neuroscience*, *10*(9), 3093–3105.

Silva, D., Feng, T., & Foster, D. J. (2015). Trajectory events across hippocampal place cells require previous experience. *Nature neuroscience*, *18*(12), 1772–1779.

Singer, A. C., & Frank, L. M. (2009). Rewarded Outcomes Enhance Reactivation of Experience in the Hippocampus. *Neuron*, *64*(6), 910–921. Retrieved from http://www.sciencedirect.com/science/article/pii/S089662730900899X doi: http://dx.doi.org/10.1016/j.neuron.2009.11.016

Skaggs, W. E., Knierim, J. J., Kudrimoti, H. S., & McNaughton, B. L. (1995). A model of the neural basis of the rat's sense of direction. *Advances in Neural Information Processing Systems 7*, *7*, 173.

Skaggs, W. E., McNaughton, B. L., Wilson, M. A., & Barnes, C. A. (1996). Theta Phase Precession in Hippocampal Neuronal Populations and the Compression of Temporal Sequences. *Hippocampus*, *6*, 149–172.

Stensola, H., Stensola, T., Solstad, T., Frø land, K., Moser, M.-B., & Moser, E. I. (2012). The entorhinal grid map is discretized. *Nature*, *492*(7427), 72–78.

Tattersall, S., & Knottenbelt, W. (2014). PIPE–The Great Re-Plumbing.

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

Taube, J. S., Muller, R. U., & Ranck, J. B. (1990b). Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *The Journal of Neuroscience*, *10*(2), 436–447.

Terrazas, A., Krause, M., Lipa, P., Gothard, K. M., Barnes, C. A., & McNaughton, B. L. (2005). Self-motion and the hippocampal spatial metric. *The Journal of neuroscience*, *25*(35), 8085–8096.

Teyler, T. J., & DiScenna, P. (1986). The hippocampal memory indexing theory. *Behavioral neuroscience*, *100*(2), 147.

Trappenberg, T. (2010). *Fundamentals of computational neuroscience* (second ed.). OUP Oxford.

Treves, A., & Rolls, E. T. (1992). Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network. *Hippocampus*, *2*(2), 189–199.

Trimper, J. B., Trettel, S. G., Hwaun, E., & Colgin, L. L. (2017). Methodological caveats in the detection of coordinated replay between place cells and grid cells. *Frontiers in systems neuroscience*, *11*, 57.

Van Cauter, T., Poucet, B., & Save, E. (2008). Unstable CA1 place cell representation in rats with entorhinal cortex lesions. *European journal of Neuroscience*, *27*(8), 1933–1946.

Wang, Y., Romani, S., Lustig, B., Leonardo, A., & Pastalkova, E. (2015). Theta sequences are essential for internally generated hippocampal firing fields. *Nature neuroscience*, *18*(2), 282–288. Retrieved from `http://www.nature.com/neuro/journal/v18/n2/full/nn.3904.html#methods`

Wilson, M. A., & McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space. *Science*, *261*(5124), 1055–1058. Retrieved from `http://www.sciencemag.org/content/261/5124/1055.abstract` doi: 10.1126/science.8351520

Wilson, M. A., & McNaughton, B. L. (1994). Reactivation of hippocampal ensemble memories during sleep. *Science*, *265*(5172), 676–679.

Wood, E. R., Dudchenko, P. A., Robitsek, R. J., & Eichenbaum, H. (2000). Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron*, *27*(3), 623–633.

Yamamoto, J., & Tonegawa, S. (2017). Direct medial entorhinal cortex input to hippocampal ca1 is crucial for extended quiet awake replay. *Neuron*, *96*(1), 217–227.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *The journal of neuroscience*, *16*(6), 2112–2126.

Zilli, E. A. (2012). Models of grid cell spatial firing published 2005–2011. *Frontiers in neural circuits*, *6*.

Appendix A

Engineering SMS

I'm not a great programmer,

I'm just a good programmer with great habits

Kent Beck

I'm not a good programmer,

I'm just an OK programmer with good habits

the author

Mike Hill (`https://www.geepawhill.org/`), a long-time friend of the author, has spent decades coaching software teams in how to write better software at less cost. He notes that people generally focus on the software that is "made", and pay less attention to the people who are the "makers" of the software, or the process of "making" the software. The rest of this dissertation is about SMS (the "made"), but as I suggest that part of the contribution of this work is the software that is SMS, in this appendix I will talk briefly about the "maker", and at more length about the "making". This is intended to help orient scientists who may

want to use or extend SMS, but do not have an extensive background maintaining software systems.

I have spent a career of almost four decades working in software, and have learned a lot about the practicalities of writing software systems. SMS differs significantly from most of the software that I have found that is used to model cortical components, which tends to consist of single programs. The logic of these programs is often quite sophisticated, mirroring the underlying mathematics that form the theory of the model. SMS offers no new mathematics; I'm a software guy, not a math guy. Instead, SMS attempts to model the interaction of multiple cortical subsystems as the interaction of a number of software components. Several of these components are borrowed, with gratitude, from the existing software I was able to find. Most of the work in SMS is understanding the behavior of these components in their interaction with the other components.

The challenge of working on a system, rather than a program, is based in the limitations of human cognition and working memory. Beyond the scope of a single program, we lose track very quickly of the details. We can cope with the logic of the component in front of us, but remembering the details of the other components it interacts with is beyond our mental ability. We end up having to move to the other component, and re-immerse ourselves in its details. At a practical level, this slows our progress greatly, and invites some form of help.

In my opinion, over the past twenty years, the single major advance in how software systems are written is the practice of writing and running small automated unit tests as an integral part of developing the software. Test-driven development (TDD) (Beck, 2003) is the process of using a small automated test to define what the system should next be able to do. A test instantiates some portion of the system, invokes some function, and then asserts what the state of the system should be as a result of that function. Initially, the test fails, because the system can't yet do the desired new function. Then we write the code to make the test pass. Once the test passes, and all the other tests also pass, the code is *refactored* to eliminate

duplicate code. Refactoring improves the internal design of the code, without changing its other behavior, meaning that the tests continue to pass.

Repeating this process over and over allows the system's design to continually evolve, in a controlled way, such that the system gradually comes to do more without ever reaching a point where it all "falls to pieces on the floor" and has to be laboriously put back together.

In addition, the tests help us greatly in managing our cognitive limitations. A working test reminds us "oh, that's how that works", as well as assuring us by its successful execution after a change, "...and it still works that way".

I wrote three kinds of software to build SMS. The four Petri nets are individually fairly simple programs, and some of their initial testing was manual, using the PIPE Petri net editor. The interaction of the Petri nets as one executable Petri net is more complex, and was tested in the context of running SMS scenarios over and over. That manual debugging was moderately painful, and took several weeks.

The software that runs the Petri nets is written in Java. The core of the system was recently completely rewritten with extensive unit tests (300+) by Sarah Tattersall (Tattersall & Knottenbelt, 2014). Without Sarah's excellent work, my work would have taken at least a year or two longer. On that foundation, I wrote the extensions that enable Petri nets to be composed of smaller Petri nets, and that enable Petri net execution to be interfaced to external systems, adding another 400+ unit tests in the process. I think of this software as being under good control, by which I mean that the impact of a change to any line of code in the system can be accurately assessed within seconds by running the tests, and seeing what tests fail. It doesn't mean that there aren't defects and deficiencies (there are many), but that defects can be fixed and features added without much fear of undetected side effects.

The third kind of software in SMS is Matlab code, which models the cortical components and their interaction with the Petri nets that model the animal's behavior. This code is

written as Matlab classes, and it also has many tests (200+). This code, however, is only under moderate control. Perhaps the quickest way to reach that assessment is just to time the tests. The 700 tests that support the Petri net code run in 4+ seconds, about 150 tests per second. The 200 tests that support the Matlab code run in several minutes, roughly 1 test per second or slower. Although this masks significant variation (some tests are very fast, and others take many seconds), slow unit test execution times always suggest that on average, the tests are testing too much behavior. The tests are invoking too much of the system at once, which is why they run a long time. But long run times are the symptom; the real problem is that whatever assertions we make about the state of the system as a result of the test execution are likely to fall well short of verifying what has actually happened under the covers.

As a consequence of the test-driven approach, and some attributes of Matlab as a development platform, and my only moderate mastery of Matlab, the Matlab portion of SMS has certain characteristics that are useful to point out.

- *controlling randomness*: a number of functions in the Matlab classes use a random number generator. To rerun the tests reliably, each test begins by resetting Matlab's random number generator to a known seed. (A similar mechanism is used in the tests for the Java Petri net code.) See: AbstractTest.m

- *adapting standalone programs*: as discussed in their initial descriptions, two of the core components of SMS, the HeadDirectionSystem and the GridChartNetwork, were borrowed from standalone programs. These programs were rewritten as Matlab classes. As I was not particularly comfortable with the underlying mathematics, portions of the code were broken out as small functions, and tested manually or with unit tests. The habitually terse variable names that are common in Matlab programs were renamed to something that gave me a better intuition of their functions. In combination with the use of smaller methods whose names explain their purpose, and the naming of tests to

explain what each test demonstrates, this enables the code to provide much of its own documentation, eliminating the need for large numbers of comments. Next, unit tests were written to verify the initial behavior of the classes.

Gradually, methods and variables were modified or added to make these classes interact with other classes. In general, however, these later modifications were driven by new parameters, whose default values permitted the class to continue to behave as it did initially. Only by giving non-default values to the parameters would new behavior be exhibited.

As an example, the initial version of the HeadDirectionSystem, as directly adapted, could associate an arbitrary "feature" with a head direction. At a later time, if the feature were presented again, the HeadDirectionSystem activation would move to the activation that corresponded to that when the feature was originally presented. An analogous function was present in the source for the GridChartNetwork. This "feature" mechanism is the way Place IDs are associated with grid activation, and the way the LecSsytem associates Place IDs with the canonical head direction view of the salient cue. Because canonical view processing is done in the LecSystem, however, and multiple head directions are not associated with the same Place ID, this capability is disabled in the HeadDirectionSystem in its later versions. The original capability still exists, however, and can still be demonstrated (S4.m and S4Test.m).

Over time, this pattern of development has resulted in a large number of parameters throughout SMS, and only a small subset of the possible combinations have been unit tested. This is a significant deficiency. Another effect is that there are tests of capabilities that are either not used in the final integrated version of the system from which results were generated, or that are in direct conflict with later functions. The origins of these puzzles are probably best seen in the git source control commit logs. Ideally, there would be fewer of these conflicts; given somewhat more time, I would remove some of them.

– *Matlab as an object-oriented development environment*: The Matlab classes are not as simple and clear as they should be, and there is duplication and complexity that should be removed. (An annoying example is that of timekeeping; different parts of the Matlab code track time steps differently, resulting in confusing discrepancies.) This is due in part to my lack of time, and in part to the limitations of Matlab as a development environment, at least as I understand and use it.

  Java is a typed language, and a large number of automated refactorings are available in modern Java development environments, in part because the existence of types removes ambiguity in assessing whether a refactoring can be done automatically. This makes it straightforward to remove duplication quickly. Matlab is not a typed language. Its support for object-oriented concepts is workable, but basic, and it has no automated refactoring capability apart from renaming variables in a method scope. All of this makes it a much slower, more manual, and error-prone process to refactor Matlab code. Finally, as mentioned earlier, the existing test suite takes many minutes to run, which is a significant disincentive to do some refactoring.

– *consequences of large tests*: as mentioned above, the Matlab tests tend on average to be slow, because many of them test too much. Especially for tests that integrate Matlab code with the Petri nets, this often results in tests that could be called "canaries". They run for a relatively long time, resulting in some arbitrary end state, e.g., a given head direction, MEC output, LEC output, and a Place ID. The test asserts that these are the values that are expected, without explaining why. The only value of such tests is that if, all of a sudden, we find one or more of these canaries dead on the floor, we can infer that a recent change has had unexpected side effects. This gives us a hint that it might be time to back out of the coal mine, and take a closer look at recent changes, but doesn't really point to what the problem might be. Examples of such tests are the SxxTest.m tests. These are tests of scripts Sxx.m, which were developed first as a way

of visualizing the system's behavior at different points in its evolution. For example, the behavior of the HeadDirectionSystem as initially adapted can be observed by:

```
s = S4(true);
s.runAll();
```

The script can be run in non-visual mode, as in the corresponding test, S4Test.m [1]:

```
s = S4(false);
s.runAll();
```

The test is a canary – it makes several assertions about the end state of the system without explaining why they should be true. Nevertheless, canary tests have proven useful in the course of complicated refactorings, to catch unintended side effects.

A less obvious deficiency of large tests is that in an effort to verify that there are not unknown side effects, multiple assertions are often made about the state of the system. This means that we understand more about the system as a result of the additional assertions. But the cost is repetitiveness. Just as a goal of refactoring the code is to eliminate duplication, ideally we should refactor the tests to eliminate duplication. If we don't, when we later change the behavior of the system, we will find that more than one test breaks. Sometimes many tests break, because they all implicitly or explicitly test the same behaviors. This can add significantly to the time required to incorporate the new change. If I were a better developer or had a lot more time, this is one of the areas where SMS could most be improved. As the quote at the beginning of the appendix suggests, this is an area where my habits are only good, not great.

[1] It is an order of magnitude faster to run the Sxx scripts and their tests in non-visual mode than in visual mode, due to the tremendous overhead in rendering the nine sub-plots that show different aspects of the system's functioning. A short video showing one of the Sxx scripts executing in visual mode, greatly speeded up, is available here: `https://youtu.be/bYX4-dY6GbQ`. It depicts SMS running without sparse Place IDs, with large grids (10 x 9), and an early version of the LecSystem with outputs from three cues. The narrative explains the information depicted by the various sub-plots.

– *tests are results*: a common tenet of test-driven development, is that if it [a feature] doesn't have a test, it doesn't exist. I suggest that the inverse is also useful: if a feature has a test, it does exist. By this criterion, SMS has several tested features, that have not been integrated into the overall system, but do work at some basic level. These might be considered partial results; at a minimum, these are early opportunities for improvement of the system.

  – Settling, or fixing position, is a mechanism to manage the proliferation of distinct Place IDs that represent place fields that are physically very close to one another, but which appear distinct in the PlaceSystem. Instead of automatically creating a new Place ID, the PlaceSystem might attempt to retrieve a Place ID, with a high threshold for matching. If a Place ID is retrieved, we could settle back to the GridChartNetwork and LecSystem values associated with that Place ID, and treat that as a type of re-calibration of the system. This could happen as often as every step. SMS has support for this type of settling, as demonstrated in this test:

    `HippocampalFormationTest.testSettlesToOriginalPlaceWhenNear`

  – As discussed in Chapter 4, *orienting* is a behavior that rats display when placed in a new environment, where they turn around, in an apparent attempt to recognize features to determine if they have been in the environment previously. Orienting makes use of the settle logic, and it has been implemented in SMS, but not integrated in the full system, as demonstrated in these tests:

    `HippocampalFormationTest.`
    `testMoveReinitializeHdsManuallyOrientToPreviousHeadDirection`
    `AnimalTest.testBothHdsAndGridsRecallPositionDirectionAfterOrienting`
    `MotorCortexTest.testMovesAwayWhenWhiskersTouchWhileOrienting`

Through the use of orienting, the system shows a basic capability, that of recognizing that the animal has been in this environment previously, and retrieving the corresponding head direction and grid positions, as demonstrated in this test:

`MotorCortexTest.testOrientTurnsAnimalAndRetrievesPreviousPlaceId`

– *testability as a first-level design consideration*: some aspects of the SMS design are driven explicitly by the need to make the system easier to test. Accepting this is part of what makes TDD work. Two examples should help make this clear.

Part of the point of the Java extensions to the PIPE Petri net code is to enable large Petri nets to be composed of multiple smaller Petri nets. This is also reflected in the integration of the Petri nets with the Matlab classes. The initial integration was done at the lowest level, between the *run* and *turn* Petri nets, and their corresponding Run and Turn Matlab classes. This enabled basic testing of the way the Petri nets drive the position of the Animal in the Environment, and the update of the state of the cortical components at each time step.

At a later point it became important to test the integration of the *run* and *turn* Petri nets within the general Petri net, *move*, and then the integration of all three in the *navigate* Petri net that drives all system behavior. But there are some differences in behavior between a Petri net that is run in isolation, and the same Petri net that is included in a larger hierarchy of Petri nets. In isolation, each Turn or Run requires setting up a separate PetriNetRunner with its own Matlab listeners for Petri net events. Particular cleanup provisions are required to prevent memory leaks. When included in a Petri net hierarchy, exactly one PetriNetRunner is required, with one set of listeners. Different cleanup provisions are required to prevent other memory leaks.

To preserve the tests that had been written at the lower levels, these behavior differences were encapsulated in a BehaviorStatus Matlab class, with two implementations, Standalone and Include. So, for example there is MoveBehaviorStatusStandalone and

MoveBehaviorStatusInclude, to enable the *move* Petri net to be tested first in isolation, and later to be tested as part of *navigate*. On the other hand, for *navigate*, there is only NavigateBehaviorStatusStandalone, as *navigate* is the top-level behavior in the system, and is not included in any larger hierarchy. This formal distinction in the design would also simplify the possible later introduction of other Petri nets to drive other behaviors, allowing any new behavior to first be tested in isolation, before being integrated.

A second example of testability as a first-level design consideration is found in the implementation of the NavigationStatus finite state machine. Initially, navigation was only tested as physical motion, driven by a single randomNavigation() method with some conditional logic. When simulation was first introduced, relatively few changes were required in the MotorCortex class, which reflects the underlying hypothesis of this work, that simulated motion re-uses many of the same motor routines used for physical motion. But some provisions had to be made for the transition from physical to simulated motion, and vice versa, so a separate simulatedRandomNavigation() method was introduced, and the state of the Navigate.Simulated place in the executable Petri net drove which of the two methods was next invoked. This worked, more or less, but proved exceedingly tedious and time-consuming to test. Not even canary tests were useful; instead, the whole system was run for an indefinite number of steps, until it hung because there were no transitions enabled to fire in the executable Petri net. This would indicate either a logic error in the Matlab code or in one of the Petri nets. (The asynchronous nature of Petri net transition firing means that events that are in some logical sequence will sooner or later happen out of sequence. For example, the next move should not start before the current move is complete, but eventually this will happen unless explicitly prevented, usually by the addition of an inhibitory arc. Even in the system as finally integrated, there is at least one such race condition outstanding,

but it was only observed once in many tens of thousands of time steps and hasn't been debugged.)

Within the Matlab code, the amount of conditional logic in the two navigation methods became non-trivial, reflecting different possible sequences. To make the code easier to understand and test, the NavigationStatus finite state machine was introduced. As discussed earlier, this enables the logic for each status to determine exactly which status should next get control. For example, when executing a simulated move under control of NavigationStatusSimulatedRandom, the next possible states are only two; either there are no more steps available in the scenario, in which case the next state prepares for exiting the system (NavigationStatusFinal), or we must settle our updated simulated position back to the current physical position (NavigationStatusSettle). As part of making NavigationStatus easier to test, a small MoveHelper interface was introduced, which just creates and runs the next move from current parameters set elsewhere in MotorCortex. A second implementation of MoveHelper, TestingMoveHelper, simply does nothing when the next move is requested except to request the next NavigationStatus. This enables fast unit tests to be written (NavigationStatusTest.m) to test the logic of moving from one status to another, without needing to create and run Petri nets.

It took a week to create the various NavigationStatus classes and test them in isolation. Turning on the function in the full system consisted of replacing the branch to either randomNavigation() or simulatedRandomNavigation() with one line, requesting the nextStatus() from the current NavigationStatus. The number of defects in the overall system dropped significantly, and the time to debug each defect dropped from hours to minutes. In the future, introducing new Status classes and appropriate tests to handle new conditions, e.g., *orienting*, will be straightforward, rather than causing a combinatorial explosion in an ever-expanding maze of conditional logic.

Appendix B

Additional findings

This section reviews additional evidence in support of Chapter 2. Together, these findings define additional behaviors that SMS could evolve to generate, detailed in Appendix D.

B.1 Evidence supporting the spatial attractor model

As implemented by Samsonovich and McNaughton (1997), the spatial attractor model was called the *multichart map-based path integrator* (MPI), and it accounts for a number of empirical findings regarding the interaction of path integration based on self-motion cues, and self-orientation from sensory cues. Gothard et al. (1996) studied rats in an environment that appeared to shrink between trials. The rats ran back and forth on a linear track with rewards at both ends. One end of the track was fixed, but the other end was a box that was moved among one of five positions along the track for a given trial. The place fields that formed in this experiment changed across trials in ways that are explained by the attractor model. Place fields close to the box appeared to be anchored to the box, and moved with the box whenever it was moved to a different location on the track. Place fields close to the fixed

end point of the track appeared to be anchored to the end point, and never changed their position. The behavior of place fields at intermediate positions on the track is instructive. When the box was moved to a position very near an intermediate place field, on a journey from the box to the fixed end point ("outbound journey") the firing rate of that place cell would drop. The cell would cease to fire altogether when the box was moved past its place field. Similarly, for a journey from the end point back to the box ("inbound journey"), a place field at a fixed distance from the end point would fail to fire if the box was at a position closer to the end point than the field. When the box was moved further away from the end point, the place field firing rate would increase.

Both of these results are predicted as a consequence of the conflict between self-motion cues and visual cues, introduced by the experimental distortion of the mapping from the animal's chart to the spatial environment. One input into the definition of a place field is the self-motion cue of "move x steps along the track". Another input into the definition is the visual cue of the rat's destination on the given journey. Under the attractor model, these two cues cause two activity bumps to emerge, one nearer the box, generated by the self-motion cues, and one nearer the end point, generated by the visual cues. At the beginning of an outbound journey, the self-motion bump is larger, and controls the animal's self-perceived position. As the animal approaches the fixed end of the track, the visual bump becomes larger, and comes to control the animal's self-perceived position through a jump in the activity bump. This jump will bypass any place fields between the two bumps, leading such place fields to stay silent (Figure B.1). Closer to either end point, the discrepancy between self-motion and visual cues may not be so great as to cause a discontinuous jump, but rather an acceleration of the activity bump to reconcile the discrepancy. The same mechanism accounts for the reverse emergence of place fields on inbound journeys to farther box positions.

Another prediction of the model is that two different places on the chart remain distinct, regardless of whether the experimental distortion of the environment appears to move them

to the same location. As predicted, Gothard et al. (1996) found that instead of collapsing into one place field, i.e., multiple cells firing at the same time and in the same phase, the two place fields remain uncorrelated.

When the physical environment is expanded, the model makes a different prediction, but the underlying mechanism is similar. O'Keefe and Burgess (1996) recorded place fields from rats exploring open-topped boxes of square or rectangular shape located at the same spot in a laboratory room. Place fields of compact shape in a small square box, were sometimes observed to be stretched in a rectangular box, with the stretching occurring along the axis that was now longer. In some cases two separate peaks were observed, again aligned with the stretched axis. As in the shrunken environment experiment (Gothard et al., 1996), the results are predicted by the MPI model as a consequence of two different activity bumps, one generated by self-motion cues, and the other by visual cues. In this case, the activity bump first crosses the place field as calculated by the self-motion cues, and later, as more distance is covered than expected, the activity bump crosses a second place field as calculated by visual cues. If the degree of physical environment stretching is lessened, then the place field may just stretch rather than double (Figure B.1).

To make the association between vestibular cues and visual cues unreliable, Knierim et al. (1995) put rats through a disorientation procedure before placing them in a cylindrical enclosure with a single cue card. The rats sometimes showed a pattern of angular drift in head direction cells and place cells over a period of minutes, evidence that could be interpreted as another instance of the gradual reconciliation of discrepancies between the path integration system and the sensory system.

Sharp et al. (1990) placed rats into a cylinder from a consistent entry location with a cue card at a consistent angle on the cylinder wall. Once place fields formed and stabilized, they introduced a second cue card 180° across from the original cue card. For the majority of place cells, the presence of the second cue card did not affect existing place fields. They also

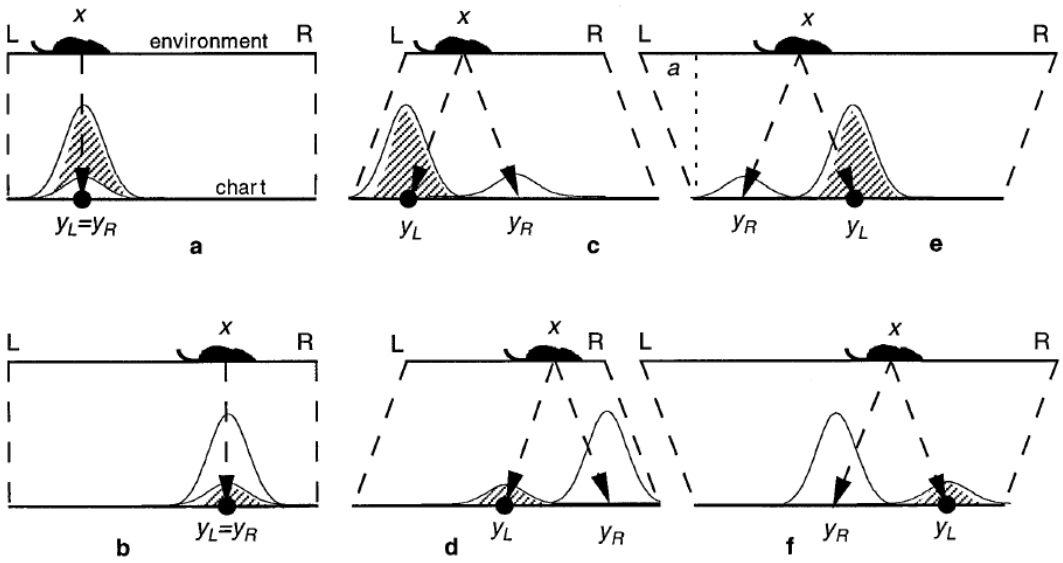


Figure 6. Splitting of the stimulated region on the chart in linearly shrunken and stretched environments. *a, b*, Original environment; *c, d*, shrunken environment; *e, f*, stretched environment. *a, c, e*, The rat is in the first half of the journey; *b, d, f*, the rat is in the second half. The activity packet location is marked by the *black dot*. The amount of stretching a is defined as shown in *e*. On each figure, the *top bar* represents the environment, and the *bottom bar* represents the chart. Given two reference points L and R (walls, landmarks, reward sites, etc.), the current rat location x is associated with two locations on the chart, y_L and y_R, according to the original map anchored at L or U_R, respectively: $y_L = x + a$ and $y_R = x - a$. These locations and surroundings are stimulated by the V array. The resultant distribution of stimulation is given by the sum of the two gaussian components, U_L (*hatched*) and U_R (*blank*), each of which is stronger near its reference point and linearly decays with distance from it (Eq. 17). For some locations x the resultant distribution has one maximum and therefore one stable activity packet location for a given x; for other locations there are two maxima, in each of which the activity packet may be trapped, and therefore, the activity packet location (and place-cell firing) may depend on the past trajectory, and through it, on the direction of motion. This may be the actual reason for acquired directionality of doubled place fields in a stretched environment, reported by O'Keefe and Burgess (1996). Switching of "host" maxima by the activity packet results in a hysteresis loop (Fig. 7). As the rat moves, the maximum of U at which the activity packet is trapped may disappear; in this case the activity packet quickly moves or jumps to another maximum and becomes bound to it. Moreover, with some probability (Eq. 15) the activity packet may jump to another maximum, when the current maximum becomes too weak and the other one is strong enough. In some cases (e.g., consider continuation of motion in **f**) the host maximum may disappear because it exits the part of the chart associated with the original environment.

Figure B.1: Place field distortions as predicted by the MPI model. Reprinted with permission from The Journal of neuroscience, 17 (15), 5900-5920. Samsonovich, A., and McNaughton, B. L. (1997). "Path integration and cognitive mapping in a continuous attractor neural network model". Copyright 1997 Society for Neuroscience.

placed the animal in the cylinder at a point 180° opposite from the standard entry location, with and without a second cue card. They found that the place field rotated 180°, relative to the room coordinates, but was otherwise unchanged. The model generates this behavior, and interprets it as the re-establishment of a known environment based on recognized self-motion and sensory cues, ignoring any less salient changes to the cue structure. Upon entry into a familiar environment, the model will generate an activity bump in the chart corresponding to the environment, at the location where the stimulation from the local view is maximized, ignoring the second cue card, and rotating with the entry location.

The model also replicates the results of Wilson and McNaughton (1993), where rats were allowed to move from a previously explored space into a novel space. Errors in the novel space associated with place field position dropped over time to levels comparable to the previously explored space. In replicating these results through simulation, the importance of self-motion cues in addition to sensory cues was reinforced, as it was the combination of both that led to a single stable activity bump. In the absence of self-motion cues, the simulation no longer generated a single activity bump; rather, a wide range of locations had activation concurrently, due to the inability of the sensory cues to establish unambiguous position.

B.2 Evidence in support of MEC / CA3 attractor dynamics

Grid cells were initially found in layer two of the MEC, but were later found in the other principal cell layers of MEC (three, five, and six), as well as in the pre- and para-subiculum (Moser et al., 2014). Sargolini et al. (2006) found other types of cells with spatially-relevant properties in the MEC. In layers three, five, and six, head direction cells were found. In the same layers, but especially in layers three and five, cells were found that were conjunctive

for both position (grid) and direction (head orientation). They fired in a grid-like pattern, but the firing rate was much elevated when the rat was facing in a preferred head direction. This made them candidates to integrate both position and direction as the animal moves. In addition, all three cell types showed firing rates that were positively correlated with the animal's velocity. The velocity, position and direction information represented by these cell types are requirements of the spatial attractor model. Their presence in the MEC suggests that the path integration function could be implemented within MEC, but there appeared to be another problem.

Sargolini et al. (2006) noted significant inter-layer connectivity within MEC, including recurrent connections from layer five to layers two and three. However, there appear to be no excitatory connections between the most common cells in MEC layer two, the *stellate cells*, which are thought to be the grid cells. This is contrary to the expectation of attractor models for the emergence of grid-like firing patterns. Couey et al. (2013) found that the stellate cells in layer two appear not to have excitatory interconnections, but do participate in a recurrent inhibitory network. Further, they showed that a simple model of that inhibition could generate the attractor dynamics to support grid cell firing patterns, given a separate source of excitatory input to drive the creation of an excitatory bump. This input may be provided by output from the hippocampus; when that output is suppressed, the grid pattern is lost (Moser et al., 2014).

Colgin et al. (2010) tested hypotheses about whether it is CA3 or MEC that is driving the spatial attractor dynamics. The authors recorded from CA1 and CA3 in rats who were trained in circular and square enclosures. For one group, the enclosures were placed alternately in the same physical location. For the second group, the enclosures were near one another and connected by a passageway. The expectation was that the first group would form one spatial representation for the enclosure, modified by sensory cues from the different shape of the enclosures, while the second group would form two distinct spatial representations for

the two enclosures, each incorporating its corresponding sensory cues. During a test trial, both groups were exposed to a series of "morphed" intermediate enclosures in the same physical location. The morphed enclosures were octagons with sides of varying lengths, which made the enclosure more "square-like" or more "circle-like".

This design enabled the test of two hypotheses about how sensory cues trigger attractor dynamics, and therefore whether those dynamics take place in CA3 or in the path-integrator system (suggested by the authors to be MEC, but this was not critical to their argument). The first hypothesis was that CA3 functions as an autoassociative network for places as modified by sensory cues, with attractor dynamics such that a significant change in either spatial location or sensory cues should lead to global remapping. The second hypothesis is that the path integration system is where the attractor dynamics are manifested. Under this hypothesis, global remapping occurs when spatial location changes, but not when sensory cues change. The training in two different locations would be expected to trigger formation of two different maps, tied to two different path integration system coordinates. Then, during the test trial, the association of the square enclosure with the first coordinates and the circular enclosure with the second coordinates might be expected to override the error in the path integration system caused by experiencing both enclosures at the same physical location. If so, as the enclosure morphed its shape, there should be an abrupt transition at some mid-point, implying that the animal perceived that it had "moved" from one enclosure to the other.

The results of the experiment support the second hypothesis. The first group of rats showed rate remapping, an approximately linear change in firing rates through the morphing sequence, without an abrupt transition. The second group of rats showed an abrupt transition approximately halfway through the morphing sequence, consistent with global remapping.

The authors point out that the hippocampal formation and parahippocampal region as a whole exhibit attractor dynamics around changes in spatial locations. The absence of the

attractor dynamics for changes only in sensory cues drives the conclusion that the spatial attractor dynamics resides in the path integration system. This is consistent with the attractor model that has been proposed for the medial entorhinal cortex, and is supported by the findings of Fyhn et al. (2007) that the grid cell population shows major changes from location differences that trigger global remapping, but little impact from changes in sensory cues that only trigger rate remapping.

J. K. Leutgeb et al. (2007) studied the response of the dentate gyrus and CA3 to environment changes that generate rate remapping. Their findings support the conclusion that the spatial attractor dynamics reside in the path integration system, not CA3, and suggest modifications to the model of CA3 as a simple autoassociative network. The authors recorded from DG and CA3 of rats who had been trained to run in square and circular enclosures. During the test trials, the enclosures were morphed to intermediate shapes between circles and squares, similar to the treatment in Colgin et al. (2010). Even minor changes in the enclosure shape generated significant changes to the activation of cells in DG, consistent with the hypothesis that one function of DG is to separate patterns into orthogonal representations. But this did not result in assigning new cells in CA3 to represent the new inputs from DG. Instead, consistent with previous findings of rate remapping (S. Leutgeb et al., 2005), the CA3 place cell population stayed constant through the incremental changes, while firing rates changed. These findings suggest that global remapping is not triggered just from the pattern separation process in DG, which appears to be active even when environmental changes are small.

B.3 Spatial attractor model objections

I have discussed some of the functions of the hippocampal formation and parahippocampal region primarily within the context of a spatial attractor model, and some of the supporting evidence for that model. Objections and alternate interpretations are now briefly considered.

Lesion studies have tested the hypothesis of the importance of the entorhinal cortex in spatial navigation. Van Cauter, Poucet, and Save (2008) performed lesions of the entorhinal cortex, confirming extensive damage to MEC and some damage to LEC, with minimal damage to other structures. Upon placement in a known constant environment, rats showed evidence of global remapping and generally less place field stability. Rotation of objects produced rotation in 49% of place fields, as opposed to 100% of place fields for controls. The authors provide several interpretations of these deficits, including that they are consistent with a role for MEC in stabilizing and/or re-establishing place fields in hippocampus. O'Reilly, Alarcon, and Ferbinteanu (2014) tested the impact of lesions in CA3 or MEC on spatial navigation tasks, and on the transmission of information to CA1, which receives inputs from both areas. The spatial navigation task involved navigating from the north or south arm of a plus maze to a goal in the east or west arm. Lesions of the complete hippocampus had previously established the need for hippocampal function to complete this task (Ferbinteanu, Shirvalkar, & Shapiro, 2011). In this experiment lesions of MEC caused less impairment than lesions of CA3, and some performance was regained over a period of days for the MEC lesion group, but not for the CA3 lesion group. Interpreting these results in terms of the spatial attractor model, I suggest that the CA3 lesion group would have established a new chart each time upon entry into the environment, due to an inability to retrieve past memories of the environment with which to anchor the chart. This might account for the observed constant performance deficit. The MEC lesion group would have had little or no ability to establish a chart to map the environment. Sensory cue inputs through LEC should have been largely unaffected, however. Perhaps this would have been sufficient to allow the establishment of a subset of stable place fields, at least within test sessions, as in Van Cauter et al. (2008). This might have been sufficient to guide performance. The improvement over several days of trials might reflect an ability to re-establish these place fields, or some other adjustment mechanism. Recording from CA1 in future experiments would enable the assessment of place field stability.

Brandon, Koenig, Leutgeb, and Leutgeb (2014) temporarily inactivated the theta oscillation in both medial entorhinal cortex and hippocampus by injecting muscimol in the medial septum. Such injections also disrupt grid cell firing patterns. During the period of inactivation, rats were exposed to a novel environment. Recording from CA1, stable place fields were seen to be established in the novel environment. Spatial selectivity, place cell size and mean firing rate in the novel environment were not different from a baseline in a familiar environment prior to theta inactivation. After the animals recovered from the septal inactivation, the novel place fields persisted. This evidence poses a problem for the hypothesis that grid cells in MEC drive the activation of a new set of place fields in hippocampus upon entry into a novel environment. Another study, however, found that formation of place fields was significantly disrupted by septal inactivation. Wang, Romani, Lustig, Leonardo, and Pastalkova (2015) recorded from CA1 in rats performing a spatial alternation task. The design included running on a wheel in the center of a two-arm maze before choosing an alternately rewarding arm. The design attempted to contrast the behavior of place fields in the presence of changing sensory cues (the arms) with behavior of place fields in the absence of changing sensory cues (the wheel). The authors found that place fields could be formed in the presence of changing sensory cues. These place fields were subject to global remapping, however, when the effects of the injections ceased, which is consistent with spatial attractor predictions that the state of the grid cell network establishes the place fields for an environment, while the impact of sensory cues is generally limited to modifying the rate of firing for an existing place cell population. In a follow-on test, the authors found that exploration of a novel, relatively large platform with few local sensory cues resulted in spatially diffuse firing patterns that were unstable. A few place fields formed in proximity to walls and corners, consistent with the previous finding regarding place fields in the presence of sensory cues.

Ormond and McNaughton (2015) addressed these discrepancies, as part of a study of place field size following different MEC inactivations. In the study, rats were given injections of muscimol directly into portions of the MEC, at low and high concentrations, along the dorsal

to ventral axis. The study tested the hypothesis that inactivation of dorsal MEC, which has a relatively small spatial scale, would lead to expansion of place fields in hippocampus, while inactivation of ventral MEC, which has a relatively large spatial scale, would lead to contraction of place fields. The authors found that place field size increased with the strength of the concentration, and at all injection locations. The impact was greater at dorsal locations. Ventral locations did not show a decrease in place field size, but the increase in place field size was smaller. Although these findings only partly support the study's hypothesis, they do show a gradation of impacts on existing place fields with differing injection concentrations and different dorso-ventral locations, results which are not consistent with the results from Brandon et al. (2014). Attempting to reconcile these differences in findings, the authors suggest that given sufficient sensory cue information from lateral entorhinal cortex, as in Wang et al. (2015), perhaps the results from Brandon et al. (2014) could be explained as an artifact of retrieval by attractor dynamics of an incorrect chart in CA3. The observed stability of the place fields following return to normal function, might then be due to binding the incorrect chart to the updated set of local sensory cues. In future studies, recording from CA3 as well as CA1 might help reconcile these differing accounts.

B.4 Theta and precession

During the theta oscillation, as an animal moves through a place field, O'Keefe and Recce (1993) found that place cells in both CA1 and CA3 fire at a slightly higher frequency than the theta oscillation. Upon entry to the place field, place cells fire late in the theta cycle, and fire earlier in each successive phase cycle. Termed *phase precession* or just *precession*, the reported range of this shift in firing was from 355° upon entry to the place field to 100° upon exit. One place field was not observed to pass through more than 360° of precession.

Hafting, Fyhn, Bonnevie, Moser, and Moser (2008) also found precession during theta in grid cells in layer two of the medial entorhinal cortex, as well as in layer five (but only sparsely in layer three). The study also found that MEC precession is independent of hippocampal precession. Precession persisted in MEC layer two after chemical inactivation of the hippocampus, suggesting that MEC may be the source of precession in the hippocampal formation[1].

In considering the functional significance of precession, O'Keefe and Recce (1993) note that precession brings the timing of firing between the place cells representing successive place fields into a time window that might serve to optimize synaptic plasticity (long-term potentiation). As an animal moves from one location to the next, a sequence of place cells is activated. If by virtue of firing in close temporal proximity, the connections between those place cells undergo Hebbian strengthening, the sequence of place cells becomes an example of what Hebb (1949) termed a *phase sequence*. Recording from CA1 and dentate gyrus in rats, Skaggs, McNaughton, Wilson, and Barnes (1996) performed a detailed study of precession, including expanding on this suggestion that one function of precession may be to strengthen connections between neighboring place fields. Hebbian strengthening requires that the post-synaptic neuron fires within a short window of time after the pre-synaptic neuron fires; for hippocampal pyramidal cells this interval is considerably shorter than the theta cycle. While an animal is passing through place fields A, B, C, and D, the authors suggested that spikes firing at different phases of the theta cycle for different cells will coincide closely in time, enabling sub-sequences to be strengthened (e.g., AB | ABC | ABC | BCD | BCD | CD). These strengthened connections would establish phase sequences in the hippocampus to represent paths through the environment. Indirect evidence for this hypothesis was found by Mehta, Barnes, and McNaughton (1997). Under the hypothesis of synaptic strengthening of traveled routes, the increased synaptic strength between A and B should cause B to fire earlier and

[1] As noted above, stable grid cell firing appears to be dependent on output from the hippocampus, so grid cell precession will eventually be disrupted.

more often, giving the place field of B the appearance of enlarging in the direction opposite the animal's movement. The study found evidence of place field enlargement, as predicted.

The establishment and strengthening of sequences of place fields is widely hypothesized as a fundamental mechanism underlying episodic memory, for example, as reviewed in Buzsáki and Moser (2013). In conjunction with the hypothesis of CA3 as an autoassociative network, it seems to make possible the storage and retrieval of temporally connected events.

Precession has an impact on computational modeling. As an approximation, the firing rates of place cells and grid cells are often assumed to be smooth distributions, but precession establishes a relationship between timing and spatial location. To account for precession in grid cell firing, Navratilova, Giocomo, Fellous, Hasselmo, and McNaughton (2012) updated the spatial attractor model discussed earlier, generating precession as the effect of cellular dynamics in the firing of the inhibitory stellate cells of MEC layer two. Although the addition of these dynamics to the spatial attractor is able to generate the additional level of detail of grid cell precession, I believe it is possible to achieve useful results in the simpler spatial attractor model without the additional complexity of the cellular dynamics. This approach is discussed briefly in Section 7.2.

B.5 Evidence on trajectories

Repetition, or *replay*, of trajectories at different time scales happens in multiple contexts.

In early evidence of trajectory replay in slow-wave sleep, Wilson and McNaughton (1994) recorded from CA1 and found that pairs of cells whose spike trains were correlated during a spatial behavioral task also showed correlation during sharp wave ripples in subsequent periods of slow-wave sleep. These correlations were absent in slow-wave sleep prior to the behavioral task. The authors suggested these correlations are consistent with models of

autoassociative memory, where portions of an input pattern can evoke a stored activity pattern. Random noise in such memories will sometimes generate the stored pattern; this might look like a SWR. The dynamics of the autoassociative network will return the system to its ground state, at which point the process could repeat. The authors cite evidence that SWRs originate in CA3, then propagate to CA1 and the output layers of the EC. As noted above, CA3 is the hippocampal structure that could implement an autoassociative network. This was suggested as a model for the repeated generation of SWRs during slow-wave sleep, and as one source of the replay of hippocampal memories to gradually update synaptic weights in the neocortex, as proposed in McClelland et al. (1995).

Lee and Wilson (2002) developed analysis techniques to extend these findings by identifying sequences of four or more place cells in CA1 that fired in the same sequence during slow-wave sleep as they had earlier during round trips on a linear track. These sequences were temporally compressed by a factor of approximately 20X.

In a study of rapid eye movement (REM) sleep in rats, Louie and Wilson (2001) found a correlation between population firing patterns in CA1 during awake task behavior and REM sleep. Over several days of trials, correlations were found in REM sleep both before and after the task behavior, suggesting that reactivation was occurring of the previous day's activity. A temporal scaling factor was introduced to assess whether the REM episodes were temporally compressed or expanded in comparison to the original task activity. The mean scaling factor was 1.4 $\pm$0.6, suggesting that REM activity proceeds at about the same speed or slower than the corresponding task activity. Theta oscillations in local field potentials were observed for both the task activity and the REM episodes. The theta values for the REM episodes were adjusted by the temporal scaling factor, and significant correlations were found between the task activity and REM episodes.

When rats ran back and forth along a linear track, receiving rewards at both ends, D. J. Foster and Wilson (2006) showed that the rats generate trajectories of place fields in reverse

temporal order during quiet awake periods. While consuming the reward, or other quiet behavior, lasting up to a minute, the rats generated repeated sequences of the place fields they had just traversed, starting with the place field they currently occupied, and moving backwards toward the other end of the track. These trajectories correlated highly with SWR events. The authors tested for the alternate interpretation, that the rats were generating a forward trajectory toward their next target, the other end of the track. In analyzing the trajectories, the authors looked for place fields that had significant levels of activation only in one direction, the direction toward the current location. The place fields were not activated in the opposite direction toward the starting point of the just-completed run, which makes implausible the interpretation that the trajectory was a forward trajectory toward the other end of the track.

These findings are consistent with a mechanism to assign temporal credit for a series of events leading to a reward. Perhaps the animal is working its way backwards towards the beginning of a rewarding strategy, updating each of the place fields with positive reward information. This is also consistent with the finding that the number of reverse replay events is higher on novel tracks than on familiar tracks; perhaps the credit assignment process is repeated until the sequence is learned, at which point the sequence requires less ongoing maintenance. Gupta (2011) proposes an alternative mechanism for generating reverse replay, based on other reports that reverse replay can be seen at points remote in time, which makes a mechanism dependent on activity decay implausible. Gupta (2011) incorporates into a model the connectivity along the septo-temporal axis of the hippocampus, across which theta oscillations propagate, and notes that inter-cellular connections extend over significant fractions of this axis. This enabled the construction of a model, with supporting simulations, wherein neurons representing places earlier in a sequence fire after neurons representing places later in the sequence, due to their locations in different layers at different spatial granularities. This has the effect of reversing the field order, enabling the strengthening of reverse connections. Stochastic factors then make it possible to see a combination of forward

and reverse sequences during SWR replay. Regardless of the detailed mechanism, reverse replay may have the adaptive value of enabling sequences representing connected locations to be traversed in either direction.

Diba and Buzsáki (2007) observed trajectories during quiet periods in both forward and reverse directions, in rats running back and forth for rewards at both ends of a linear track. Separating the place cell populations into two groups, favoring one direction or the other on the track, the authors found that a large majority of forward trajectories occurred prior to the beginning of the run, and a large majority of the reverse trajectories occurred after the end of a run.

Karlsson and Frank (2009) found that rats performing a task in one environment would replay events both from that environment and from a second physically different environment during SWR events. In their design, one environment was more novel than the other environment. During SWR events that replayed trajectories, as many of 44% of the events taking place physically in the familiar environment were replay events of the novel environment. This suggests that awake replay need not be solely driven by current sensory input, but may also be activation of representations of previous experience, especially when the previous experience is of a relatively novel environment.

The authors noted that they found replay of trajectories within SWRs, in experimental circumstances similar to those of Johnson and Redish (2007), where SWRs had only been observed during consummatory behavior. The authors suggested that the previous result might be due to a much higher threshold for detection of activity in the SWR band (seven standard deviations above the mean, as opposed to three standard deviations, used by the authors), or that the previous study was correct in distinguishing between trajectories in SWRs, and trajectories during periods of theta oscillation. I am not aware of later studies to resolve this question; if attempting to incorporate these findings in future work I would

accept the conclusion of Johnson and Redish (2007) that trajectories can be generated during periods of theta oscillation, often behaviorally associated with VTE.

Davidson et al. (2009) studied trajectories generated during periods of immobility while running on a long track (10 m), to determine whether and how replay works on larger spatial scales. The trajectories were recorded from CA1 in untrained animals, and generally coincided with SWR events. The observed trajectories represented distances of varying lengths, up to the total length of the track. The current location was the starting point 40% of the time. The end point of the trajectory was the current location only 5% of the time. This is consistent with trajectories depicting forward paths from the current location, as opposed to replaying memories of past trajectories leading to the current location. Approximately 20% of events depicted remote trajectories (starting point $>$ 50cm from current location), to positions either ahead or behind the animal. The trajectories were reconstructed from the population activities spanning multiple SWR events. The duration of the replay was found to be linearly related to the distance traversed in the trajectory.

B.6 Pfeiffer and Foster (2103) detail

Pfeiffer and Foster (2013) recorded from CA1 of rats in an open square arena, two meters on a side, containing 36 wells. Each well could be filled with a reward (liquid chocolate) from below, silently. The trial consisted of two phases. In the first phase, the rat started at a well, designated the Home location, and foraged for a reward randomly placed in some other well, designated the Random location. Finding the Random location and consuming the reward completed the first phase. In the second phase, which began automatically without a signal, the Home location was filled with a reward. Finding and consuming the reward at the Home location completed the second phase and the trial. The next trial began immediately, without signal to the animal, by filling the well at another random location. A session

consisted of 30 trials or 30 minutes, whichever was less. Rats were tested for one session per day. The location in which the rat was placed for the initial trial for each day was the Home location for all trials of that day's session. A different Home location was used for each day. For the first 19 trials of each day's session, the Random location did not repeat. These features introduced novel combinations of Home and Random locations for the animals to solve.

The authors found that routes taken by the rats to Home wells were significantly shorter in distance and duration than the routes to the Random wells. As these routes are symmetrical for each pair of Random and Home locations, this suggests that the rats could not detect the Random locations, but did remember where the Home well was.

During periods of immobility, brief increases in population spiking were identified as candidate events. These events were decoded in the same fashion as the spike trains for physical movement. For 25.3% to 43.9% of candidate events, decoding showed "temporally compressed, two-dimensional trajectories across the environment", termed *trajectory events*. There were between 144 to 373 trajectory events per session (Figure 6.2).

The mean duration of the trajectory events was 103.6 ms, and the path lengths were from 40.0 cm to 199.1 cm. The trajectory events coincided closely with SWR events identified through analysis of local field potentials.

The trajectory events were divided into those that occurred while the rat was at the Home well ("home events"), and those that were initiated at the Random well ("away events"). Trajectory events occurred at the same rate in both locations. The home location was represented more frequently than any other location in home events, termed *initiation bias*. This is unsurprising, as the home location also represents the rat's physical location at the time of the event, and would be expected to be present whether the rat was planning a future motion or remembering a recent motion. Significantly, however, the home location

was represented more frequently than any other location in away events, and was the end point of the trajectory at levels much greater than any other location on both days (P-values (Grubbs test for outliers): Day 1 6.9×10^{-4} (Lilliefors test, P-value 0.29); Day 2 6.0×10^{-4} (Lilliefors P-value 0.42)). The results held when the analysis was restricted to the first 19 trials on each day, when the combination of Home and Random locations was novel. Also, the result was not a reflection of occupancy time; other locations in the arena with greater occupancy times were not strongly represented in the away-events. These results also differ from the bias that rats show towards a "home base" location (Eilam & Golani, 1989). In a novel environment, rats will establish a home base location from which they explore the rest of the environment, and to which they frequently return. In the experiment, the arena is not novel, but is well-known. Under typical home base behavior, the rat chooses an arbitrary location as home base; in the experiment, the rat has learned that the initial location in which it is placed for a day's session is "home". Finally, the design ascribes a specific meaning to home, in that home is only rewarding once a reward has been found at an away location.

The authors examined the representation of Random locations, to determine if there was an increase in representation of the Random wells where rewards were experienced. The design of the task made the Random well useful for prediction only in a negative sense; for at least the first 19 trials, receiving a reward at a Random well meant that that well would not be rewarding again, at least in the near future. Analogous to the analysis of the Home location in home events, analysis of the away events showed an initiation bias toward the current Random location. In the analysis of home events, however, the Random locations were little represented. Summarizing this pair of findings, the animals' over-representation of the Home goal, and lack of over-representation of the Random non-goals, accurately reflects the reward structure of the experiment.

The over-representation of the Home goal is suggestive of some planning mechanism. To quantify this effect, the angular displacement was calculated between the trajectory and the

animal's immediate future path and immediate previous path. The angular displacement between the trajectory and both the future path and the previous path was clustered around zero, but the distribution was much more tightly clustered for the future path (Figure B.2). The difference existed for both home and away events, but was stronger for away events, perhaps reflecting the strength of the Home goal. For home events, there is some anti-correlation between the trajectory and the previous path, which might constitute recognition that the previous Random well was never correct as a next destination. An additional analysis showed that the current head direction of the animal during away events was not related either to the direction toward the Home location or toward the immediate future path. This rules out an interpretation of the trajectory as the path currently directly ahead of the rat. Finally, dividing the away events between those that preceded paths to Home and those that preceded paths to non-Home locations elsewhere in the arena, in both cases the trajectory was significantly more predictive of the future path than of the previous path. This suggests that the trajectory has broad planning significance.

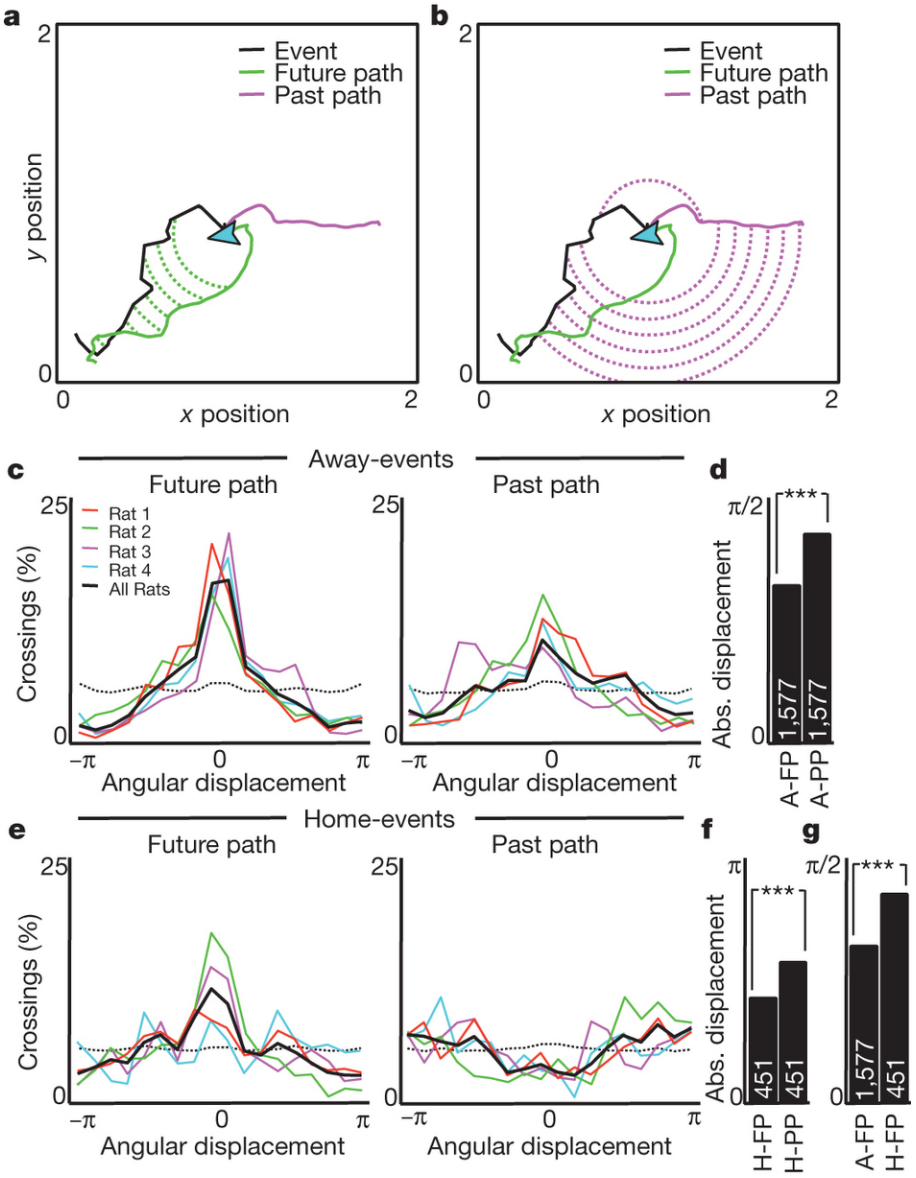


Figure B.2: Correspondence to past or future path (Pfeiffer & Foster, 2013): **a**, **b**, Representative event from R1,D1, demonstrating trajectory event vector (black), immediate future (green) and past (magenta) path (up to 10 s or 50 cm, whichever is greater), and angular displacement along the minor arc between event and future (**a**) or past (**b**) path at each crossing. **c**, Per cent of crossings across all events as a function of angular displacement for all away-events compared to future (left) or past (right) path. Dashed line indicates chance based upon 2,000 shuffled events. **d**, Mean absolute angular displacement for away-events compared to future (A-FP) or past (A-PP) path. Abs., absolute. **e**, **f**, As **c**, **d**, for home-events. **g**, Mean absolute angular displacement for future path for all away-events (A-FP) or home-events (H-FP). P-values (Wilcoxon rank-sum test): 8.60×10^{-31} (**d**); 3.54×10^{-17} (**f**); 7.25×10^{-16} (**g**). Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Nature ("Hippocampal place-cell sequences depict future paths to remembered goals", Pfeiffer, B. E., and Foster, D. J.), (2013).

Appendix C

Testable predictions

The hypothesis of simulated motion makes several neuroscientific predictions. Testing these predictions is outside the scope of this dissertation. The significance of each prediction is discussed in italics; in some cases, if subsequent experimental results do not support the predictions, it would cast serious doubt on the viability of the simulated motion hypothesis.

1. Recording from MEC and LEC layer two, and CA3, simultaneously during quiet waking SWRs should show evidence of coherent communication between the areas, possibly at the SWR frequencies, or perhaps more likely, at low gamma frequencies. *The simulated motion theory suggests that the place field activations in CA3 and CA1 are due to activations of grid cell populations on the active chart in MEC, and activation of populations in LEC representing other dimensions, both of which necessitate communication between the entorhinal cortex and hippocampus during quiet awake SWRs. O'Neill et al. (2017) directly contradicts this prediction; they found little evidence of coherent communication between MEC layers two and three and CA1 (see Chapter 4). This constitutes strong evidence against the simulated motion hypothesis. On the other hand, Yamamoto and Tonegawa (2017) found evidence that communication be-*

tween MEC and CA1 was required for replay on an extended track during quiet waking periods. This appears to support the simulated motion hypothesis.

2. Recording from MEC and LEC layer 2, and CA3, simultaneously during quiet waking periods, should show trajectories during High Synchrony Events in MEC, whose positions are correlated with those expressed in later SWRs in CA3. *This is an alternate prediction to address the failure to find coherent communication as predicted under (1). Under this prediction, perhaps activation from MEC causes sub-threshold activation in CA3 place cells, raising the probability of their firing in subsequent time periods.*

3. In motor cortex, the populations that represent the motor routine for movement should be activated during the trajectory event in the hippocampus. *Under the simulated motion theory, communication passes from motor cortex to entorhinal cortex to hippocampus. Coherent communication between all these areas is a necessary component of the theory, and as above, failure to find evidence of such communication would cast serious doubt on the simulated motion theory.*

4. The trajectory communicated between EC and CA3 during active SWRs and during vicarious trial and error during exploration is communicated at the same low gamma frequency (25-50 Hz). *It is parsimonious, but not essential, to suggest that simulated motion takes place at one rate within the motor system, and is accommodated within the hippocampal and retrohippocampal formations according to the current behavioral mode (theta or LIA).*

5. For quiet awake SWRs during spatial alternation tasks, it might be possible to predict the direction of the trajectory after the choice point, from the firing rates of the trajectory on the common path prior to the choice point. *Trajectories generated from physical movement encode retrospective and prospective information about the task (Frank et al., 2000; Wood et al., 2000); if such information is also encoded in quiet awake SWRs, it means that simulated motion can be more predictive of reward than just a representa-*

tion of the target location itself. Note that this is not the mechanism used in the SMS model, but is a plausible alternative and is consistent with the overall simulated motion theory.

6. Placement of a rest box in the center of a larger open arena without salient cues visible from the rest box, and with which the animal has had no previous experience, should lead to a set of trajectories that covers the arena in a roughly uniform distribution, and whose place fields are re-expressed on subsequent physical exploration. *Expanding on the design of Dragoi and Tonegawa (2013), this does not distinguish between the endogenous allocation of place fields that the authors hypothesize, from the simulated motion hypothesis, but the anticipated results would show the continued viability of the simulated motion theory as an alternative account.*

Appendix D

Model Behaviors

This table summarizes behaviors from findings in Chapter 2 and Appendix B, and the degree to which SMS currently support them, or the work required to support them in the future. Existing components are referenced where applicable, and are described in Chapter 4. In many cases, the evidence was collected from studies attempting to account for behavior by reference to models that have been partially re-implemented in SMS, e.g., a spatial attractor model, complementary learning systems, memory indexing theory. Implicit in the entries below is the assumption that these models and theories will continue to be the basis for SMS, until/unless they are demonstrated to be unworkable.

File and component references are from the github repository: `https://github.com/sjdayday/simulated-motion/tree/master/src`. The status is one of these:

- current: SMS currently demonstrates this behavior
- test: SMS demonstrates this behavior in isolation, in a unit test, but it has not been integrated into the overall behavior of the system

– future: SMS could potentially demonstrate this behavior, with the work defined under *Conditions*

Behavior	*Status*	*Conditions*
1) The animal's head direction should be represented, and be updated in response to angular velocity changes.	current	Part of the original head direction model adopted for use in SMS. See HeadDirectionSystem
2) The animal's head direction should be associated with features presented at a given head direction. Upon future presentation of the feature, the animal's representation of head direction should return to the orientation associated with the feature.	test	Part of the original head direction model adopted for use in SMS. Works in isolation, but superseded by canonical view logic of LecSystem. See HeadDirectionSystem and LecSystem
3) Memory Indexing theory: unique Place IDs should be stored for known locations, and should then be used as an index to retrieve the state of other systems at the time they were created.	current	Implemented for HeadDirectionSystem in isolation, and for GridChartNetwork and LecSystem
4) The head direction system should be recalibrated as needed by reference to visual cues, to control drift.	test	Termed *orienting*, and supported in HippocampalFormation
5) The animal should recognize physical locations as being close to previously visited locations, and should represent them with the existing place ID, creating a place field of nontrivial size.	test	Termed *settling*, and supported in HippocampalFormation

6) The animal's position in two-dimensional space should be represented, and be updated in response to angular and linear velocity changes.	current	Part of the original grid cell model adopted for use in SMS. See GridChartNetwork
7) The animal's position in two-dimensional space should be associated with features presented at a given location. Upon future presentation of the feature, the animal's representation of position should return to the position associated with the feature.	current	Part of the original grid cell model adopted for use in SMS. See GridChartNetwork
8) Upon re-entry to a single known environment, the animal should be able to orient, and retrieve its position from its angle to salient and constant visual cues.	test	LecSystem retrieves Place ID through visual cues alone; retrieved Place ID is used by GridChartNetworks to retrieve associated grid position. See LecSystem, GridChartNetwork, and HippocampalFormation
9) The animal should be able to represent its position as the output of multiple charts at multiple orientations and spatial gains (Maurer et al., 2005).	current	HippocampalFormation creates multiple GridChartNetworks
10) The animal should be able to retrieve its position across any of multiple known environments.	future	Moderate effort, given (5)

11) On a known linear track that shrinks or expands, place fields near a landmark should stay anchored to that landmark (Gothard et al., 1996).	future	Moderate effort
12) On a known linear track that shrinks or expands, conflicts between self-motion and visual cues tied to landmarks should lead to two conflicting activity bumps (Gothard et al., 1996).	future	Moderate effort
13) On a known linear track that shrinks, place cells representing place fields past the point of shrinkage should cease firing (Gothard et al., 1996).	future	Minimal effort, given (11, 12)
14) In a known environment that expands, place fields should expand along the axis of expansion (Colgin et al., 2010).	future	Minimal effort, given (11, 12)
15) In a known circular environment with a single salient cue that is moved by an angle between trials, should result in place fields that rotate by the same angle (Sharp et al., 1990).	future	Minimal effort
16) The animal should be able to represent the salience of a place or object in LEC, e.g., whether it has been rewarding (Deshmukh, 2014).	future	Model subcortical region (amygala) as associating Place IDs with reward. Major effort

17) Rate remapping or an equivalent mechanism should be supported to characterize changes to some but not all dimensions at the same location (S. Leutgeb et al., 2005; Deshmukh, 2014).	future	Moderate effort
18) On a linear track, rate remapping differences should increase over time for the same location in opposite directions (Navratilova, Hoang, et al., 2012).	future	Moderate effort
19) Global remapping when morphed enclosure shape more closely resembles a known second environment distal from first, morphed environment (Colgin et al., 2010).	future	Moderate effort
20) Global remapping displayed across MEC and CA3 at the same time upon moving to a new environment (Fyhn et al., 2007).	future	Moderate effort
21) Changes to some but not all dimensions at the same location should leave the index generated by the hippocampus unchanged or slowly changed (Kumaran et al., 2016).	future	Implement a model of CA1 to keep the place index slowly changing. Major effort
22) Population nodes in CA3 can be activated at sub-threshold level.	future	Move from binary to continuous representation of population node activation in CA3. Major effort.
23) Place sequences can be stored in autoassociative networks and retrieved in association with the state of other systems.	future	No work has been done on this. Major effort, as the interaction with other processes in SMS is likely to be complex

24) Place sequences fire at different rates in correlation with different future task demands (e.g., upcoming T-maze decision) (Singer & Frank, 2009).	future	Moderate effort, given (17, 23)
25) The animal should create trajectories representing the options (vicarious trial and error) at choice points (Johnson & Redish, 2007).	future	Moderate effort
26) Trajectories should represent the probability of a reward (e.g., paths down the arms of a T-maze) (Johnson & Redish, 2007).	future	Moderate effort, given (16)
27) Exploration in a previously inaccessible portion of the arena should retrace previous simulated trajectories (Dragoi & Tonegawa, 2011).	future	Minimal effort
28) Exploration in a larger environment after time in a rest box in the center of the environment should retrace a uniform distribution of previous simulated trajectories (Dragoi & Tonegawa, 2013).	future	Minimal effort
29) Preplay trajectories in a task with visible but unreachable reward should predict future physical trajectories and place IDs (Ólafsdóttir et al., 2015).	future	Moderate effort
30) Phase sequences are generated through a replay mechanism during periods without spatial displacement.	future	Moderate effort

31) Phase sequences are strengthened through replay in reverse temporal order (D. J. Foster & Wilson, 2006).	future	Moderate effort
32) Replay trajectories may begin or end at the current location, or may be remote from the current location (Pfeiffer & Foster, 2013).	future	Could start with random retrieval of a place in the arena as a starting point for a simulated motion using existing mechanisms. Minimal effort
33) Simulated trajectories may begin at locations remote from the current position (Karlsson & Frank, 2009).	future	Moderate effort
34) Complementary Learning Systems: cortical structure is updated by hippocampal replay to predict reward in current task.	future	A possible very partial approach is outlined in the dissertation proposal: Doubleday-dissertation-proposal-v4.pdf. Very preliminary work has been done, e.g, see SimulationCorticalProcess.m and CorticalNeuralNetwork.m and associated tests. Major effort

Glossary

action potential The abrupt depolarization of a neuron, generating a signal that propagates through the *axon* to downstream neurons. Also called a *spike*. 219

allocentric Allocentric references orient the animal to features of the environment, e.g., "head towards the tower". Contrast with *egocentric*. 18, 220

allocortex The older portion of the brain, as opposed to the *neocortex*. Includes the *hippocampus*. Also called *archicortex*. 27, 219, 221, 223

allothetic cues Sensory cues from the visual, auditory or olfactory systems. Contrast with *idiothetic cues*. 18, 221

assembly A group of neurons whose inter-connections are sufficiently strong such that activation of one neuron results in the activation of other neurons in the assembly. The mechanism thought to enable the formation of assemblies is *long-term potentiation*. 19, 61

attractor network A dynamic system whose state can be defined as a vector. Some states of the system function as attractors, toward which other nearby states evolve over time; see *point attractor*. 21, 218, 224

autoassociative network A network where the outputs are mapped back to the inputs, which maps from a set of inputs to an existing pattern, returning an output of the same

type; contrast with *heteroassociative network*. Functions as an *attractor network*. 22, 25, 221, 224–226

axon The portion of a neuron that sends signals to other neurons. There is typically one axon per neuron, which may project for long distances, before branching and forming *synapses* to connect with the *dendrites* of other neurons.. 25, 217, 219, 225, 226

CA1 A portion of the *cornu ammonis* area of the hippocampus, receiving major inputs from the *entorhinal cortex* and *CA3*, and sending outputs back to the *entorhinal cortex*. 27, 218, 221, 224, 226, 227

CA3 A portion of the *cornu ammonis* area of the hippocampus, receiving major inputs from the *entorhinal cortex* and *dentate gyrus*, and sending outputs both to *CA1* and back to itself through the *Schaffer collaterals*. 27, 218–221, 223–227

chart An abstract plane, mapping to a given environment, where place cells are arranged symbolically such that the pattern of neural activation is a localized activity packet centered on the animal's head. Part of an attractor model of path integration in two dimensions (Samsonovich & McNaughton, 1997). 36

coherence The phase-locked firing of groups of neurons oscillating at the same frequency, such that outputs from one group arrive at the other group at a time of maximum receptivity. 61

Complementary Learning Systems A theory proposed by McClelland et al. (1995), to explain the interaction and function of the fast-learning hippocampus, and the slow-learning neocortex. Under the theory, the hippocampus learns upon one-time exposure, and then replays what it has learned many times, to enable the neocortex to update its network weights, developing a representation of the *structure* of the events. Abbreviated CLS. 20, 223, 225, 226

consolidation The process of establishing connections between memory traces in different cortical areas so that some or all of a memory may be recalled across cortical areas without involvement of the hippocampus. 33

cortical layers The *neocortex* has a six-layered organization. Layer 1 is comprised of *axons* from other areas. The remaining layers have cells of different types, of which *pyramidal cells* are the most prominent. Layers 2 and 3 are sometimes called the superficial layers. Layers 4, 5, and 6 are sometimes called the deep layers. Layer 4 generally receives input from other areas, but is missing in *entorhinal cortex*. Entorhinal cortex is a transitional structure from the six layers of the neocortex to the three layers of the *allocortex*. 27, 224, 226, 227

dendrite The portion of a neuron that receives signals from other neurons, via many *synapses*, typically located on short projections called *dendritic spines*. Typically a neuron has many branching dendrites. In the typical depiction of a neuron, the cell body is in the center, and the *apical dendrites* rise above it like a tree, while the *axon* projects below like a root. The apical dendrites are sometimes referred to as the *dendritic tree*. There are sometimes dendrites projecting "below" the cell body, in the same area as the axon; these are *basal dendrites*. The portion of a dendrite closest to the cell body is *proximal*; the portion that is farther away is *distal*. Normally, signals synapsing in the proximal portion of a dendrite will make a greater contribution to an *action potential* than signals synapsing in the distal portion.. 25, 29, 30, 218, 223, 225, 226

dentate gyrus An area of the *hippocampus* that receives inputs from the *entorhinal cortex* through the *perforant path*. The *granule cells* are the principal cells of the DG, and are thought to orthogonalize inputs to *CA3*. 27, 29, 218–221, 223–225, 227

detonator synapse A type of connection in a *Hebb-Marr network* whose firing causes the firing of the *principal cell* with high probability. When *dentate gyrus* and *CA3* are

considered as a Hebb-Marr network, these are the collection of synapses from *granule cell*s in *dentate gyrus*, whose axons tend to run parallel to the dendrites of the *pyramidal cell*s in *CA3*, resulting in multiple synapses for a given pyramidal cell, such that when the granule cell fires, the pyramidal cell is likely to also fire. 23, 30, 221, 223

efference copy Signals to the motor system are also sent to other systems within the brain, enabling them to make predictions as to upcoming motion against which to compare actual inputs in future time periods. 39, 54

egocentric Egocentric references orient the animal with respect to itself, e.g., "turn to your left". Contrast with *allocentric*. 18, 217

entorhinal cortex Transitional region from the *hippocampus* to the *neocortex*. Most inputs to and outputs from the hippocampus pass through the entorhinal cortex. Consists of the *medial entorhinal cortex* and the *lateral entorhinal cortex*. 27, 218, 219, 222–224, 227

excitatory neuron A neuron whose firing increases the probability that downstream neurons will also fire. The information that is input to a network of neurons is conveyed by the firing pattern of excitatory neurons, modulated by *inhibitory neurons*. In referring to the activity of "cells" or "neurons", excitatory neurons are implied. 220, 222, 225

global remapping The expression of a different set of place fields upon exit from one environment and entry into a second environment. The sets of place fields will be uncorrelated with each other. See *rate remapping*. 44, 225

granule cell The principal *excitatory neuron* found in the *dentate gyrus*, named for its small size. The axons of the granule cells are known as the *mossy fibers*, and synapse near the cell bodies of the *CA3 pyramidal cells*. 29, 219, 220, 223

grid cell cells in the medial entorhinal cortex whose firing pattern resembles a pattern of tessellated equilateral triangles overlaid on the rat's environment. Each cell has

multiple firing fields, each surrounded by an inhibitory field. The distances between firing fields for a given cell is constant across the environment, resulting in a distinctive hexagonal pattern. 38, 223, 226

head direction cell cells that fire at an elevated rate when the animal's head is facing at a specific angle from some egocentric anchor point. Head direction will rotate as the anchor point rotates. Found in dorsal presubiculum, medial entorhinal cortex, and other areas. 34

Hebb's rule A theory proposed by Hebb (1949), that the simultaneous firing of the neurons on either side of a synapse would cause the efficiency of transmission to increase at that *synapse*. 19

Hebb-Marr network Neural regions that plausibly implement *heteroassociative* and *autoassociative networks*. Components consists of *principal cells*, *detonator synapses*, *modifiable pathways*, and *inhibitory interneuron*. *Dentate gyrus* and *CA3* together are thought to constitute a Hebb-Marr network.. 219, 223, 225

heteroassociative network A network that maps from a set of inputs to an existing pattern, returning an output that may be of a different type; contrast with *autoassociative network*. 22, 218, 221

hippocampal formation A set of bi-lateral regions in the medial temporal lobe, consisting of the *dentate gyrus*, *CA1*, CA2, *CA3* and the *subiculum* (Figure 2.3). 226

hippocampus One of the main structures of the *allocortex*, responsible for storing memories of places where an animal has been and their significance, termed *episodic memory*. Consists of the *dentate gyrus* and the *cornu ammonis*, whose major subareas are *CA3* and *CA1* (Figure 2.3). 217, 219, 220, 224

idiothetic cues Internal self-motion cues from the vestibular, proprioceptive and motor systems. Contrast with *allothetic cues*. 18, 34, 217

inhibitory neuron A neuron whose firing decreases the probability that downstream neurons will also fire. Inhibitory neurons (also often called *interneurons*, due to their role in connecting networks of neurons together) generally *synapse* on many *excitatory neurons*, and help control the dynamics of a network of neurons, e.g., by suppressing the firing of all but the most highly active cells. Another function is to impose a periodic pattern on the firing of cells (see *theta oscillation*). 23, 220, 221

large-amplitude irregular activity Large amplitude changes in the *local field potentials* at a slower frequency than *theta*. LIA is a characteristic pattern of activity in the hippocampus during quiet activity where spatial displacement is not involved, including sitting quietly, grooming, eating, drinking, or during slow-wave sleep. See *sharp wave ripple*. 18, 226, 227

lateral entorhinal cortex The lateral portion of the *entorhinal cortex*, whose principal function is thought to be the representation of places in the animal's environment along both spatial and non-spatial dimensions. Abbreviated LEC. 27, 220

local field potential Electrical potential measured in the extracellular space around neurons. 18, 222, 226

long-term potentiation A sustained increase in the efficiency of synapses between presynaptic and post-synaptic neurons when both fire in close temporal proximity. 20, 217

map A set of locations with spatial relationships, and possibly, additional information about what has been experienced at a given location. Navigation by use of a map is more complex than by use of a *route*, but more flexible, because paths can be calculated between any two points on the map. 17, 225

medial entorhinal cortex The central portion of the *entorhinal cortex*, whose principal function is thought to be the representation of the animal's position in two-dimensional space, through the activation of networks of *grid cell*s. Abbreviated MEC. 27, 220

memory indexing theory Suggestion that the hippocampus serves to bind together memories in multiple cortical areas by providing an index to each area through which the memories can be retrieved as a single episode. 32

modifiable synapse A type of connection in a *Hebb-Marr network* whose strength is weak in comparison with the *detonator synapse*, but whose weight is modifiable. These synapses are formed and strengthened when they fire simultaneously with the detonator synapses for a given *principal cell*, as expected by Hebb's rule. When *dentate gyrus* and *CA3* are considered as a Hebb-Marr network, these are the collection of synapses in the *dendrites* of the *pyramidal cell*, from the axons that project from layer two of the entorhinal cortex to CA3.. 23, 31, 221, 223

monosynaptic path A branch of the *perforant path* that projects from layer two of the entorhinal cortex to CA3. This is the *modifiable path* input to the CA3 *recurrent autoassociative network*. Contrast with *trisynaptic circuit*. 29

mossy fibers The axons of the *granule cells*, projecting onto the *proximal dendrites* of *CA3*. 29, 220, 227

neocortex The "new cortex", as opposed to the *allocortex*. Neocortex processes sensory input, identifying temporal patterns. Thought to build representations of the structure of events (see *Complementary Learning Systems*). 27, 217, 219, 220, 226

orienting When a rat is introduced to an environment it looks around for salient cues, to decide whether the environment is known or novel. 103, 181

orthogonalization The process of making input vectors as decorrelated as possible; orthogonal vectors have a correlation of zero. This is an important mechanism for maximizing the storage capacity of an *autoassociative network*. 30

path integration Maintenance of an estimate of one's position in an environment, and calculation of direct routes to other points in the environment, based on the position estimate. 33

pattern completion A property of an *autoassociative network*, where presentation of an incomplete input will result in the output of the complete pattern. 22

perforant path The path from the *superficial layers* of the *entorhinal cortex* to the *dentate gyrus* and *CA3* (projecting from EC layer two), and *CA1* (projecting from EC layer three) (Figure 2.6). 19, 29, 219, 223, 227

phase precession As an animal moves through a place field, the corresponding place cells fire earlier and earlier in relation to the *theta oscillation*. This precession proceeds through 360° and resets upon exit from the place field. 195

phase sequence Representations of successive sensory-motor events.. 25, 196

place cell A type of *pyramidal cell* located in *CA3* and *CA1* of the *hippocampus*, that fires when the animal is in a particular location (see *place field*) in the current environment, and is largely silent elsewhere.. 17, 224

place field The physical location within the current environment where a population of *place cells* reliably fires. 17, 224

point attractor A single state of an *attractor network* toward which nearby states will evolve over time. 21, 217

preplay Refers to *trajectories* that appear to predict the animal's future path. Used to distinguish from trajectories that are correlated with past motion, termed *replay*. 225

principal cell The type of cell in a *Hebb-Marr network* that generates the output of the network. When *dentate gyrus* and *CA3* are considered as a Hebb-Marr network, this is the *pyramidal cell* in CA3. 23, 219, 221, 223, 225

pyramidal cell The principal type of *excitatory neuron*, named for the shape of its cell body. 27, 29, 219, 220, 223–226

rate remapping The change in firing rates of existing place cells when cues change in a known environment. The cues may include sensory cues, the nature of the task, directional orientation, etc. See *global remapping*. 44, 220

recurrent autoassociative network A *Hebb-Marr network* where the outputs from the *principal cells* are also inputs to the same cells, i.e., when branches of the *axons* of the principal cell synapse on the *dendrites* of the same cell. *CA3 pyramidal cells* are recurrent. Functions as an *autoassociative network*. 223

replay Refers to *trajectories* observed during various behavioral states that are observed in hippocampal or closely associated regions, e.g., entorhinal cortex. The trajectories may be at various temporal scales, may express places that include the current location or not, may be in forward or reverse order from the current location, and may be correlated with past or future trajectories. Typically associated with *sharp wave ripples*. Thought to have multiple functions, chief among them the repetition of previous experience for purposes of update of the neocortex, under *Complementary Learning Systems* theory. Replay that correlates with future trajectories may also be termed *preplay*. 33, 197, 224

route Connected locations from a starting point to a goal, navigated from memory or by reference to a landmark or a compass orientation. Routes imply limited flexibility in navigation; an error in following a route may mean failure to reach the goal. See *map*. 17, 222

Schaffer collateral The axon from a *pyramidal cell* in *CA3*, synapsing both in the dendritic trees of CA3, and of *CA1*. The synapses within CA3 are "recurrent excitatory", and provide the connections that enable CA3 to act as an *autoassociative network*. 29, 218, 227

sharp wave ripple A combination of a large amplitude change in *local field potentials* in the hippocampus ("sharp wave"), with a series of fast oscillations ("ripple") in an approximate frequency range of 150-250 Hz. Occurs during periods of *large-amplitude irregular activity*, during activity that does not involve spatial displacement.. 19, 222, 225

stellate cell cells in *layer 2* of the medial entorhinal cortex that are thought to act as *grid cells*. Morphology consists of a cell body with dendrites radiating in all directions. 190

structure "any systematic relationship that exists within or between the events that, if discovered, could then serve as a basis for efficient representation of novel events or for appropriate responses to novel inputs" (McClelland et al., 1995, 436). Under *Complementary Learning Systems* theory, learning structure is the function of the *neocortex*. 20, 218

subiculum A region in the *hippocampal formation*, that receives outputs from *CA1*, and passes information on toward the *neocortex*. Abbreviated SUB. 27, 221

synapse Intersection point between the *axon* of one neuron (sender) and a *dendrite* of another neuron (receiver). The synapse is a small gap across which a chemical / electrical signal passes with greater or lesser efficiency. The more efficient the transmission, the greater the contribution that the sending neuron makes to the probability that the receiving neuron will fire.. 19, 218, 219, 221, 222

theta oscillation Changes in the *local field potentials* in an approximate frequency range of 6-10 Hz. Theta oscillation is a characteristic pattern of activity in the hippocampal

and retrohippocampal formations, and is associated with behaviors such as movement between locations and exploration. Contrast to *large-amplitude irregular activity*. 18, 222, 224

trajectory A sequence of adjacent place fields. 49, 224, 225

trisynaptic circuit A principal circuit by which information flows through the hippocampal formation (Figure 2.6). Beginning in the *superficial layers* of the *entorhinal cortex*, the first leg of the circuit is to the *dentate gyrus* via the *perforant path*. The second leg is from the dentate gyrus to *CA3*, via the *mossy fibers*. The third leg is from CA3 to *CA1* via the *Schaffer collaterals*. CA1 projects to various output regions, including back to deep layers of the entorhinal cortex, completing the circuit. 29, 223

vicarious trial and error A behavior of small head movements alternating in the direction of each of multiple choices. 50