

A COGNITIVE APPROACH TO MOBILE ROBOT
ENVIRONMENT MAPPING AND PATH PLANNING

Peter J. Zeno

Under the Supervision of Dr. Tarek Sobh and Dr. Sarosh Patel

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

May, 2017

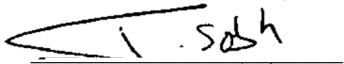
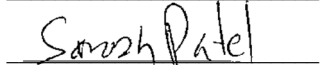
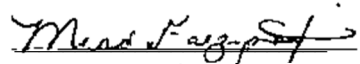
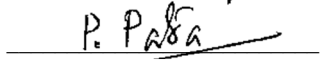
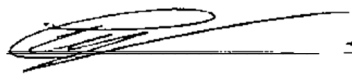
A COGNITIVE APPROACH TO MOBILE ROBOT
ENVIRONMENT MAPPING AND PATH PLANNING

Peter J. Zeno

Under the Supervision of Dr. Tarek Sobh and Dr. Sarosh Patel

Approvals

Committee Members

Name	Signature	Date
Dr. Tarek Sobh		6/20/2017
Dr. Sarosh Patel		6/20/2017
Dr. Miad Faezipour		6,21,2017
Dr. Prabir Patra		06/26/17
Dr. Nicolas Cuperlier		29/05/17


Ph.D. Program Coordinator

Dr. Khaled M. Elleithy		6/28/17
------------------------	--	---------

Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood		6-26-2017
-------------------	--	-----------

Dean, School of Engineering

Dr. Tarek M. Sobh		6/20/2017
-------------------	--	-----------

A COGNITIVE APPROACH TO MOBILE ROBOT ENVIRONMENT MAPPING AND PATH PLANNING

© Copyright by Peter Zeno 2017

A COGNITIVE APPROACH TO MOBILE ROBOT ENVIRONMENT MAPPING AND PATH PLANNING

ABSTRACT

This thesis presents a novel neurophysiological based navigation system which uses less memory and power than other neurophysiological based systems, as well as traditional navigation systems performing similar tasks. This is accomplished by emulating the rodent's specialized navigation and spatial awareness brain cells, as found in and around the hippocampus and entorhinal cortex, at a higher level of abstraction than previously used neural representations. Specifically, the focus of this research will be on replicating place cells, boundary cells, head direction cells, and grid cells using data structures and logic driven by each cell's interpreted behavior. This method is used along with a unique multimodal source model for place cell activation to create a cognitive map. Path planning is performed by using a combination of Euclidean distance path checking, goal memory, and the A* algorithm. Localization is accomplished using simple, low power sensors, such as a camera, ultrasonic sensors, motor encoders and a gyroscope. The place code data structures are initialized as the mobile robot finds goal locations and other unique locations, and are then linked as paths between goal locations, as goals are found during exploration. The place code creates a hybrid cognitive map of metric and topological data. In doing so,

much less memory is needed to represent the robot's roaming environment, as compared to traditional mapping methods, such as occupancy grids. A comparison of the memory and processing savings are presented, as well as to the functional similarities of our design to the rodent's specialized navigation cells.

ACKNOWLEDGEMENTS

My deepest gratitude goes to my family, friends and educators who have blessed me with their support along this incredible journey. For without their encouragement and enlightenment, such a journey would not have been possible.

Thank you, Dad, for always being there for me and for your unquestioning support. Your insight and stories related to animal navigation behavior have always been inspirational and thought provoking.

I am honored to have had the opportunity to work under the supervision of Dr. Tarek Sobh and Dr. Sarosh Patel. I am fortunate to have been guided by Dr. Sobh in my thesis topic. For he believed in my vision and allowed for me to follow a topic that was along a path less traveled. Additionally, I greatly appreciate the real-world advice and continued guidance I receive from Dr. Patel.

I would like to sincerely thank Dr. Miad Faezipour, Dr. Prabir Patra and Dr. Nicolas Cuperlier for accepting to be members of my dissertation committee. I greatly appreciate their time and efforts in the review of my work, taking part in my defenses, and giving incredibly valuable feedback.

Finally, I would like to express my deepest gratitude to my best friend Roberta Voss for her company and loving support that always kept me motivated.

TABLE OF CONTENTS

ABSTRACT	iv
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES.....	xi
ACRONYMS	xiii
GLOSSARY	xv
CHAPTER 1: INTRODUCTION.....	1
1.1 Research Problem and Scope.....	1
1.2 Motivation.....	3
1.3 Contributions	4
CHAPTER 2: SPATIAL AWARENESS IN RODENTS	6
2.1 Specialized Navigation and Spatial Awareness Rodent Brain Cells	6
2.1.1 Place Cells.....	7
2.1.2 Head Direction Cells.....	8
2.1.3 Boundary Cells.....	8
2.1.4 Grid Cells	8
2.2 Path Integration.....	9
2.3 Review of Rodent Spatial Awareness and Navigation Models	10
2.3.1. Arleo and Gerstner 2000	11

2.3.2. Fleischer et al. 2007	14
2.3.3. Strösslin et al. 2005	16
2.3.4. Hafner 2008.....	18
2.3.5. Barrera and Weitzenfeld 2008	19
2.3.6. Wyeth and Milford: RatSLAM, Version 3	22
2.3.7. Cuperlier et al. 2007.....	24
2.3.8. Grid Cell Centric Systems.....	28
2.4 Analysis of Reviewed Systems' Localization	31
CHAPTER 3: SENSORY INPUT.....	35
3.1 Idiothetic Sensors for Path Integration Model.....	35
3.1.1 Heading Sensor	37
3.1.2 Motor Encoders.....	39
3.2 Allothetic Sensors	39
3.2.1 Ultrasonic Range Sensors	39
3.2.1 Visual System	42
CHAPTER 4: NEW MULTIMODAL PLACE CELL MODEL	45
4.1 Multimodal Place Cell Model Basics	45
4.2 Logical Architecture of Multimodal Place Cell Model	45
CHAPTER 5: NAVIGATION SYSTEM IMPLEMENTATION.....	49
5.1 Hardware System Design.....	49

5.2 Software Design.....	51
5.3 FPGA Design.....	57
CHAPTER 6: LOCALIZATION AND PATH PLANNING.....	59
6.1 Localization	60
6.1.1 Level of Confidence Calculation	61
6.1.2 Localization Accuracy	62
6.1.3 Place and Boundary Field Initialization Accuracy	62
6.2 Route Planning.....	64
CHAPTER 7: SUMMARY & FUTURE DIRECTIONS.....	67
7.1 Summary.....	67
7.1.1 Power Analysis	67
7.1.2 Navigation Environment Scalability.....	68
7.1.3 Episodic Memory	69
7.1.4 Importance of Visual Recognition in Navigation	70
7.2 Possible Future Directions in Model Computation.....	71
7.2.1 Neural Networks	71
7.3 In Brief.....	76
REFERENCES	78

LIST OF TABLES

Table 2.1. Neurobiologically based navigation research.	33
Table 3.1. Electrical and Mechanical Specifications of the HC-SR04 Ultrasonic Range Sensor.....	40
Table 5.1: Visual Part of Exploration Mode.....	53
Table 7.1. Power Consumption of Ratbot's Processors.....	68

LIST OF FIGURES

Figure 2.1. The rodent brain.	7
Figure 2.2 A functional overview of the ANN based directional system.....	13
Figure 2.3. Neural connectivity of the medial temporal lobe, including the hippocampus of Darwin XI.	15
Figure 2.4. Simulated neural system.....	17
Figure 2.5. Neural network structure as a result of learning connectivity between place cells.	19
Figure 2.6. Computational spatial cognitive model.....	20
Figure 2.7. World graph layer module.....	21
Figure 2.8. Connectivity diagram of the RatSLAM, version 3.....	22
Figure 2.9. The RatSLAM system.	24
Figure 2.10. The system's neural network based model architecture.....	26
Figure 2.11. Assignment of dedicated place cell fields.	27
Figure 2.12. Topographical cognitive map.....	28
Figure 2.13. Architecture representation of the persistent spiking computational model	31
Figure 2.14. Grid cell firing fields.	31
Figure 3.1. A conceptual overlay of an internal Cartesian graph representation of the <i>ratbot</i> 's navigation environment.....	36
Figure 3.2. MPU6050 post drift compensated gyroscope data.	38
The allothetic heading θ from the gyroscope is calculated as follows:	38

Figure 3.3. Ultrasonic range sensors covering the front of the ratbot.....	40
Figure 3.4. Chart No. 67 from Acoustic Design Charts.....	42
Figure 3.5. The ratbot's Pixy Cam downward looking FOV.....	43
Figure 3.6. Demonstration of the Pixy camera.	44
Figure 4.1. The proposed multimodal model of the PC firing field sources.	46
Figure 4.2. The logic architecture of the multimodal PC model.	47
Figure 4.3. BC activation cases.....	47
Figure 4.4. PC activation cases.	48
Figure 5.1. Top-level block diagram of the ratbot's neurobiological based navigation system.	50
Figure 5.2. The ratbot and its hardware.	50
Figure 5.3. Central processor's main loop pseudo code. CPM is the central processor's memory.	51
Figure 5.4. Software block diagram of the <i>ratbot's</i> neurophysiological based navigation system.	54
Figure 5.5. Ratbot Simulator output.	55
Figure 5.6. The PC data structure.	55
Figure 5.7. BC implementation in the FPGA.	58
Figure 6.1. Recording of BC (BVC) location and angle of incident.	63
Figure 7.1. Single layer ANN with two inputs, two outputs and two neurons.	74

ACRONYMS

ANN	Artificial Neural Network
ATN	Anterior Thalamic Nuclei
BC	Boundary Cell
BVC	Boundary Vector Cell
CA	Carnus Amonis
CAN	Continuous Attractor Network
CPU	Central Processor Unit
DG	Dentate Gyrus
EC	Entorhinal Cortex
FF	Firing Field
FOV	Field of View
FPGA	Field Programmable Gate Array
GC	Grid Cell
GPGPU	General Purpose Processing using a Graphics Processor Unit
GPU	Graphics Processor Unit
HD	Head Direction
IR	Infrared
LEC	Lateral Entorhinal Cortex
LMN	Lateral Mammillary Nuclei
LV	Local View

mEC	Medial Entorhinal Cortex
MEMS	Microelectromechanical Systems
PaS	Parasubiculum
PC	Place Cell
PI	Path Integration
PoS	Postsubiculum
PrS	Presubiculum
SLAM	Simultaneous Localization and Mapping
TC	Turn Cell
VPC	Visual Place Cell

GLOSSARY

Word	Definition
Allocentric	Perspective with respect to objects in the environment.
Allothetic	External or global based. Example: Allothetic based cues used for navigation/localization, such as landmarks.
Affordances	All possible actions available to take at a given moment after sensing the environment.
Anterior	Near the front of an object, such as an organ.
Cognitive	Relating to cognition; concerned with the act or process of knowing, perceiving, etc. [1].
Distal	Situated away from the center of the (robot's or animal's) body.
Dorsal	The upper side of an animal or organ (e.g., hippocampus).
Foraging	Searching for food in by an animal in its environment.
Hippocampus	A region of the brain that is primarily associated with memory, as well as a key role in spatial processing and navigation.
Kinesthetic	Relates to the learning of sense of body (e.g., position and motion) through feedback to the brain from muscles, tendons and joints.
Idiothetic	Internal or self-motion based.
Incentives	A thing that motivates a biological or mechanical entity (e.g., food seeking and food intake driven by the hypothalamus).
Lateral	Relating to the sides of an object, such as an organ.

Neurophysiological	Relating to a branch of physiology and neuroscience that is focused on the study of the functioning of the nervous system.
Posterior	Towards the rear of an object, such as an organ.
Proprioceptive	Relating to stimuli that is produced and perceived by the nerves connected to tissue (e.g., muscle) used for position and movement of the body.
Ratbot	The name given to the autonomous mobile robot used in this paper.
Salient	Most noticeable or important.
Somatosensory	Relating to sensation from touch, pain, or warmth from any part of the body.
Taxon	As used as a navigation type, taxon navigation refers to following a visual cue to get to a target location.
Ventral	Of or relating to the bottom portion of an animal or organ.
Vestibular	Relating to the inner ear or an animal's sense of balance.
Wayfinding	Following a known route, from one known location to another. This involves the use of spatial awareness.

CHAPTER 1: INTRODUCTION

Autonomous mobile robotics have many diverse applications and domains (i.e., indoor, outdoor, underwater, and airborne). For instance, indoor applications include security, rescue, and service mobile robots, while outdoor applications include driverless automobiles. Underwater and airborne robot systems include ocean and space exploration robots, respectively. The success of any autonomous mobile robot is based on its ability to reliably navigate in its environment. This is especially true for animals and other living creatures whose survivability is dependent on their ability to navigate effectively in their environment. They would perish if they were unable to find and relocate food and cache locations, their home, as well as shelter spots from predators. Navigation, for both biological creatures and machines, can be defined as the ability to maintain a course when going from one location to another [2, 3].

1.1 Research Problem and Scope

The basic tasks required for accomplishing navigation are localization and mapping. In robotics, the combined task of mapping and localization is also referred to as the simultaneous localization and mapping (SLAM) problem [4-6]. Thus, it is assumed that the robot is starting in a completely unknown environment, and it needs to map the environment while localizing itself within that environment. Because autonomous mobile robots have sensors, actuators and navigation algorithms that cater to their application and working environment [7, 8], these robots are still very rigid in their navigation capabilities.

Problem areas include navigating through dynamic environments and the need for high precision localization data for mapping and path planning.

Animals, on the other hand, are masters at navigating in their environments. For central to biological based navigation is the ability to travel from one place to another without getting lost. It was suggested by Tolman in 1948 that for rats and humans to be able to accomplish various navigation tasks, they must have a cognitive map of their environment in their head [9, 10]. In 1971, O'Keefe and Dostrovsky [11] discovered a special type of neuron in the rodent's hippocampus that fired only when the rodent was in a specific location and was aptly named the place cell. It became evident that the place cell (PC) was part of the suspected cognitive map and has been heavily researched from that point on. Since the discovery of the place cell, the head direction (HD) cell, and the boundary cell (BC) were discovered in the rodent's hippocampus, and the grid cell (GC) in the neighboring entorhinal cortex (EC). These specialized neurons are believed to play a vital role in the navigation abilities of the rodent. The hippocampus is also believed to be involved in the storage of new episodic memory [12, 13].

In addition to cognitive maps used by rodents, many species, such as spiders, crustaceans, insects, birds, and many mammals continually update an internal vector trajectory with respect to their previous location [9, 14]. This internal vector math allows for the animal to take the most direct path between any point in the environment to their home, even in the dark and through unknown areas, and despite having left via a circuitous route. This is accomplished through dead reckoning, which is also known as path integration (PI), as originally proposed by Darwin [15]. This natural form of vector based

navigation is speculated to take place in, or around the hippocampus and its surrounding area, in the rodent's brain.

We could have chosen a simpler neural based navigation system to implement in a robot, such as that found in many insects (e.g., bees and ants). These systems are fundamentally dominated by optical flow data. For many types of insects use image-matching memory, as well as PI to accomplish navigation [16-18]. Such a choice would have made our system purely dominated by its visual recognition capabilities of landmarks and other visual cues. However, much of what is known about insect navigation is surmised from external manipulation tests and behavioral observance of insects. Hypotheses are thus drawn from these observations. In vivo observation of the insect's neural circuitry related navigation is very limited at this point. Thus, we chose the more complex neurophysiological based navigation system of the rodent to emulate due to the wealth of data gathered from use of probes in various parts of the rodent's brain, while it performed navigation tasks.

1.2 Motivation

Over the past couple of decades, a great deal of research has gone into creating computational models of the rodent's spatial awareness and cognitive mapping neural circuitry. The goal of such research has been to better understand their role in navigation, and estimate the dependency between these neurons. Another motivating factor for such research is its application to autonomous mobile robot navigation. Because of the need for high computational processing demands of multi-neural network systems, most of the robot systems that are used to represent the rodent are a combination of an external cluster

of computers connected to a mobile “robot” that provides the computers with sensor data and acts on the action commands sent from the computers. The low-level detail of these specialized neurons, their connections and dependencies, as well as the validity of the models certainly have a place in research to better understand the brain. However, the question that is worth answering, and is the topic of this thesis, is whether the specialized spatial awareness and navigation neurons, and their interconnections, can be adequately approximated by a core system of greatly reduced processing demand. Particularly, a core navigation system that has the ability to create and use cognitive maps in Tolman’s sense, that is, for shortcut abilities and for latent learning [3, 9]. If so, and if this core navigation system works for many disparate navigation strategies as exhibited by the rodent, then such a system would be a great candidate to use in mobile robotic navigation systems. Additionally, this would allow for the testing of supporting mechanisms to rodent navigation by neuroscientists, biologists and alike to use a simplified core to better understand the rodent’s behavior (e.g., navigation strategies, stimulus-reward associations, associations between location information with reward and emotional information [9], testing of various levels of visual systems, etc.) rather than creating a computational neurophysiological based navigation system that is only designed and tested to work for a single navigation strategy (e.g., water maze, 8-arm maze, regular maze, open area, foraging, wayfinding, etc.).

1.3 Contributions

This paper presents a novel, low power, neurophysiological based, navigation system which mimics the basic functionality of the rodent hippocampus in terms of spatial

awareness and navigation capabilities. The issues at hand for replicating the same functionality of neurophysiological based circuits are: 1) the brain does all its processing using neural networks composed of neurons, axons, dendrites and synapses, 2) very little of how the brain performs various functions are fully understood, and 3) the implementation of artificial neural network (ANN) based algorithms on currently available conventional computational resources is very computationally intensive, and thus requires much power and time to perform. Therefore, the removal of realism in terms of ANN based algorithms, where possible, is the only way of accomplishing the low power goal. As will be shown, some processing performed by the brain, such as directionality (vestibular stimuli) and movement (proprioceptive stimuli), are accomplished easily with microelectromechanical systems (MEMs) based gyroscopes and wheel encoders, respectively. However, without the use of ANNs for place recognition, the robot's environment needs to be somewhat engineered for the use of less computationally intensive goal and landmark recognition techniques.

An additional benefit of our research is the creation of a neurophysiological based navigation framework which may help neuroscientists obtain a better understanding of how these specialized neurons interact at a more abstract level.

CHAPTER 2: SPATIAL AWARENESS IN RODENTS

As briefly discussed in the Introduction section, the rodent performs mapping and localization through the use of multi-sensory input, yet can perform homing behavior with only internal sensory input (vestibular and proprioceptive) through path integration [19, 20]. It would at first seem that these two navigation methods are unrelated. However, after reviewing the specialized neurons the rodent uses for navigation and spatial awareness, we will discuss the neurophysiological connection between these two navigation methods. Additionally, key state of the art research in rodent neurophysiological based navigation systems as installed on, or integrated with, mobile robot platforms will be presented. Further details can be found in [21]. Additionally, earlier, similar research can be found in [2, 3, 22].

2.1 Specialized Navigation and Spatial Awareness Rodent Brain Cells

The rodent brain has been studied greatly, particularly the hippocampus and its surround area for its navigation related cells [9, 23]. These cells (neurons) include: place cells, boundary cells, and head direction cells (in the subiculum), and grid cells (in the neighboring EC). The rodent is not the only mammal with these special brain cells. Mice, rats, and bats have been found to also have place cells and grid cells [24, 25]. However, this list is most likely longer. A brief description of the firing characteristics of these navigation related brain cells follow and can also be found in [21, 26]. Fig. 2.1a illustrates the location and size of the rodent hippocampus (left and right), while Fig. 2.1b illustrates the major components of the hippocampus, via a cross section horizontal slice of the

ventral portion of the hippocampus. The location(s) of the specialized navigation cells with respect to the areas shown in Fig. 2.1b, as well as their basic behavior are covered next.

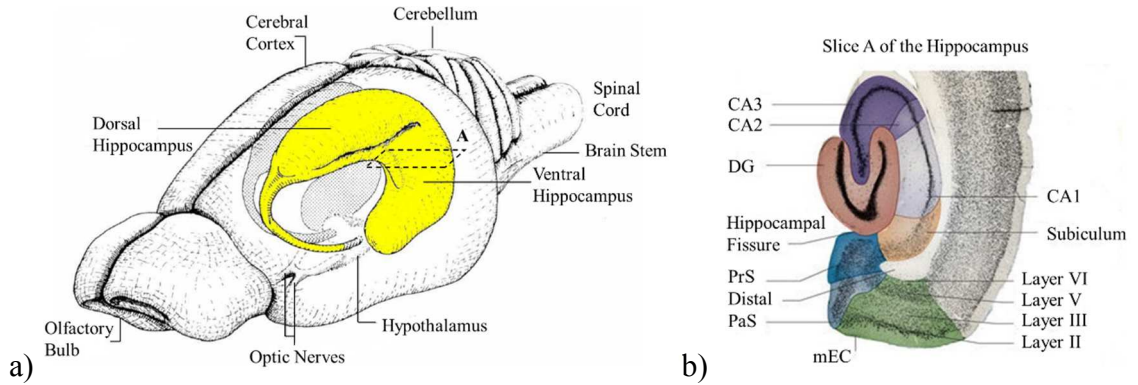


Figure 2.1. The rodent brain. (a) In yellow is the left hemisphere hippocampus. (b) Anatomy of hippocampal formation and parahippocampal region (horizontal slice A in part a). Abbreviations: Carnus amonis (CA), dentate gyrus (DG), lateral entorhinal cortex (IEC), medial entorhinal cortex (mEC), parasubiculum (PaS), and presubiculum (PrS). Picture adaptations: Fig. (a) from [27], and (b) from [25].

2.1.1 Place Cells

The place cell (PC) was the first spatial type of brain cell to be discovered (O’Keefe and Dostrovsky in 1971). A PC fires maximally when the rodent is in a particular location of its environment [23, 28]. A PC is usually limited to a single firing field (FF), unless the environment is large. Additionally, the firing of a PC is only dependent on location and not direction (in rodents), unless the place field is in a constrained location, such as a maze corridor. PCs are found mainly in CA3 and CA1 of the hippocampus, and to a lesser degree in the dentate gyrus (DG) with smaller place fields [9]. Thus, PCs play an important role in the mapping of the rodent’s environment. A PC’s FF size is dependent on its type and location in the hippocampus. As stated in [29], a place field can be defined as the area

between the points in an environment where the theta phase precession begins and terminates.

2.1.2 Head Direction Cells

The head direction (HD) cell fires at a preferred direction (+/- a few degrees) of the rodent's head in the horizontal plane, irrespective of the rodent's location, the angle between its head and body, and eye movement. Thus, HD cells provide the rodent with a sense of directional heading relative to its environment (allocentric based). HD cells are found primarily in the rodent's postsubiculum (PoS), the anterior thalamic nuclei (ATN) and the lateral mammillary nuclei (LMN) [9, 30].

2.1.3 Boundary Cells

The boundary cell (BC) is similar to the PC in that it is direction invariant and location specific in its firing. Also, as with the PC, the BC typically has a single FF, which is dedicated to a specific border, barrier, or boundary. BCs can be found in the medial entorhinal cortex (mEC), parasubiculum (PaS) and subiculum [23]. However, it is believed that there are boundary vector cells (BVCs) in the subiculum which fire according to a fixed distance and direction to a boundary [31, 32].

2.1.4 Grid Cells

The grid cell (GC) is the most unique spatial awareness brain cell to be found in the rodent and was discovered by the Mosers in 2005. GCs are predominantly found in layer II of the mEC (shown in Fig. 2.1b), which is located one synapse upstream of the PCs in

the hippocampus [33, 34]. The GC differs from the PC and BC such that it has many FFs. Each GC's FF maps over the rodent's entire roaming environment at the vertices of equilateral triangles, which creates a hexagonal lattice. The FFs of a GC create a hexagonal lattice across the environment, which is defined shortly after a rodent is introduced to a novel area [35]. It is suggested that the lattice is anchored in orientation and phase to external landmarks and geometric boundaries [34, 36, 37]. Additionally, each FF of a GC is direction independent. Although, there exists conjunctive grid cells in the middle and deeper layers of the EC, which fire only on a given absolute direction [37-39]. The FFs of GCs differ from one another in three possible ways: size, orientation and phase. The GCs FF's size increases monotonically from its dorsal to ventral location in the mEC [37, 40].

2.2 Path Integration

PI is accomplished through different methods by various creatures. However, any PI system requires the integration of some form of compass (sense of direction) and distance cues [41, 42]. Influencing factors as to the type of sensory used by a given species includes the complexity of their nervous system or brain, and their native environment. For example, the *Cataglyphis fortis* (desert ant), as well as other arthropods, receives directional information from the sun (angle of polarized light and/or direct light), and distance traveled from proprioceptive cues, and to a lesser extent, from optic flow information [14, 43, 44]. This allows for the desert ant to navigate circuitously hundreds of meters from its home, in a featureless environment, and still return in a straight vector.

It is generally believed that the PI neural circuitry of a rodent is located in and around its hippocampal formation, see Fig. 2.1b. More specifically, it is argued in [9] that the subiculum (Sub), the parasubiculum (PaS), and the superficial layers of the EC make up the PI circuit. This circuit creates a chain of neural processing stages that involve the head direction system, primarily found in the PoS, and the PaS is interconnected with the posterior cingulate [23], which possibly supplies additional directional and self-motion information. However, it has been shown that the hippocampus and EC are not essential for PI in humans [45].

2.3 Review of Rodent Spatial Awareness and Navigation Models

As found in [21], this section covers state-of-the-art research in neurobiological based navigation systems, where the systems have been implemented in a mobile robot since the early 2000s. Although the emphasis on the review of these rodent inspired navigation models is on the computation resources needed to realize ANN based models, the relationship between the various navigation based specialized brain cells can be easily extracted from these reviews. Due to the heavy computation resources required, many of the systems researched rely on external central processor units (CPUs) to perform neurophysiological simulation for the robot (e.g., Khepera mobile robot platform). Of course, silicon packages have continued to shrink during this time frame (early 2000's to present day). However, since increasing a processor's clock has no longer been an option due to heat dissipation issues, multi-core CPUs have been the solution for squeezing out any possible performance increase. It is certainly possible to use today's multi-core technology, such as multi-core CPUs and graphics processor units (GPUs) to create an

onboard solution for this type of research. Such systems need to be well planned and tradeoffs made as to response time latency incurred by the transfer of data back and forth between different multi-core technologies, processing time, as well as power limitations.

The definition of an autonomous mobile robot used in this paper, requires the complete processing system to be onboard the mobile robot system, with no external computing resources required. Thus, the autonomy classification of each robot presented is included in Table 2.1.

2.3.1. Arleo and Gerstner 2000

Arleo and Gerstner 2000. The study article by Arleo and Gerstner, 2000 [46], has had an influence, in one form or another, on many future works covered in this section, particularly [47, 48]. The references used in [46] fall into the categories of both neuroscience: O'Keefe and Nadel, 1978 [49]; Taube et al. 1990 [50]; Redish, 1997 [22]; and so forth and neurophysiological inspired circuits and models: Burgess et al. 1994 [24]; Brown and Sharp, 1995 [51]; Redish and Touretzky, 1997 [52], Zrehen and Gaussier, 1997 [53]; and so forth, which form a basis of references used by the other proceeding studies/articles. More references can be found in Arleo and Gerstner, 2000 [46] and 2000 [54]. Additionally, this paper's presentation and functional use of neurobiological specialized spatial navigation cells found in the rodent's hippocampus, for modeling in robotic navigation, are central to the theme of all papers reviewed.

(1) System Architecture

The Khepera robot system used consists of the following: an onboard camera for vision based self-localization (90° field of view in horizontal plane), eight infrared (IR) sensors for obstacle detection and light detection, a light detector for measuring ambient light, and an odometer for sensing self-motion signals. The neurobiological based navigation system models two crucial spatial navigation cells: HD cells and PCs. This is performed on an external computer.

(2) Head Direction and Place Cells for Spatial Navigation

In Fig. 2.2, the allothetic (external cue sourced stimuli) inputs consist of data from the onboard camera, which is used for the place cells in the sEC submodule, as well as data from the eight IR sensors and the ambient light sensor, which are used by the visual bearing cells in the VIS submodule (left side of Fig. 2.2). The neural networks (Sanger's [50]) to the PC from the camera input are programmed offline during an initial unsupervised, Hebbian learning phase [51]. During this initial, exploration/neural network training phase, each PC location is learned by dividing images taken into smaller 32×32 pixels, running the reduced image through 10 different visual filters of 5 set scales each. This is done for the north, west, south, and east views of the robot's arena from each snapshot/PC location. The networks of each cell are then trained with the reduced images and adjusted for maximum response for each image location. Thus, the place cells are programmed neural networks with the onboard camera image, divided into four quadrants of 32×32 pixels each, at the input, and will allow for self-localization in the online mode.

A light source is added to one wall of the robot's arena, where the IR sensors and ambient light sensor can lock onto this global direction (with the help of neural networks for fine-tune positioning to the light source). This allows for calibration of the robot's directional module (right side of Fig. 2.2), which bounds the accumulated error in directionality.

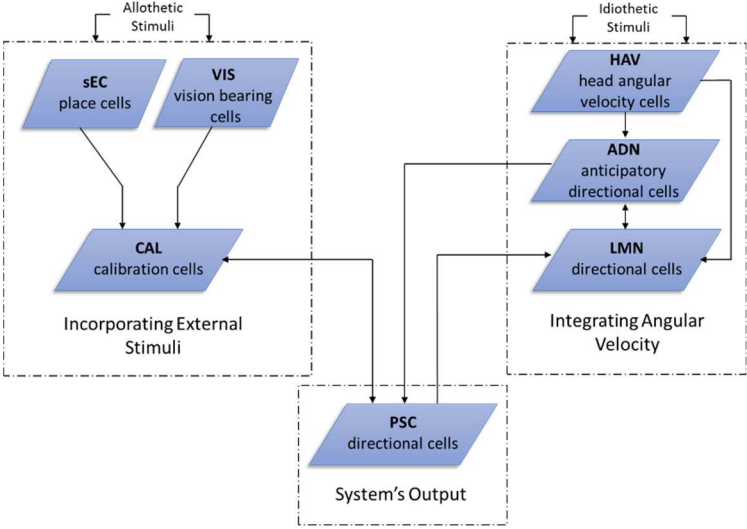


Figure 2.2 A functional overview of the ANN based directional system [46].

The robot uses three different neural populations of cells (right side of Fig. 2.2) to calculate its head direction from its current angular velocity and anticipated angular velocity and feedback from the system output and calibration cells. The result is a set of quantized, directional cells to drive the robot's motors for proper heading.

(3) Computational Demand

The computational demand of this system is a bit more extensive than briefly covered here. Further details can be found in [46, 54, 55]. However, any neural network system is going to have a relatively high need for computational resources and processing time requirements, based on the number of neural networks and the processing status of offline and online/real-time learning. The environment is somewhat engineered and needs to be static. This is true though of any system in the initial stages of wringing out system integration errors, model problems/accuracy, and so forth.

(4) Mapping and Route Planning

Visual based mapping, through the use of snapshot recognition (place cells), is used to help correct head direction error and not for obstacle avoidance or route planning. Therefore, true mapping and any form of route planning are not addressed in [46, 54].

2.3.2. Fleischer et al. 2007

(1) System Architecture

The neurophysiological modeled navigation system for Darwin XI mobile robot designed by Fleischer et al. [56] is not autonomous, by the definition used in this paper, due to the use of external computers to simulate a detailed neurophysiology based system. However, the system pushes the limit on simulating large scale features of vertebrate neuroanatomy and neurophysiology (the medial temporal lobe specifically) in real time. Using a Beowulf cluster of 12 x 1.4 GHz Pentium IV computers running a Linux operating system, sensor data is communicated on a wireless link from the mobile robot to one of

the cluster computers, while motor data is sent back to the robot. The simulation processing cycle from sensor data input to motor command output is approximately 200 ms of real time. The simulator, referred to as the brain-based device (BBD), simulates 57 neural areas, 80,000 neuronal units, and approximately 1.2 million synaptic connections.

Darwin XI is equipped with a visual system (camera), a head direction system (compass) plus wheel odometry (current head direction), a laser range finder system (facing downward to detect neuronal reward), and a whisker system which reads bumps along the plus-maze walls.

(2) Modeled Hippocampus

A schematic of the mobile robot mobile I/O sensors connected to the corresponding neurophysiological based navigation system can be found in [56]. However, Fig. 2.3 illustrates the type of connections and simulated parts of the medial temporal lobe, including the hippocampus entities. Fig. 2.3 is similar to that found in [23, 25]; however further details pertaining to the various layers of the EC are lacking in this figure.

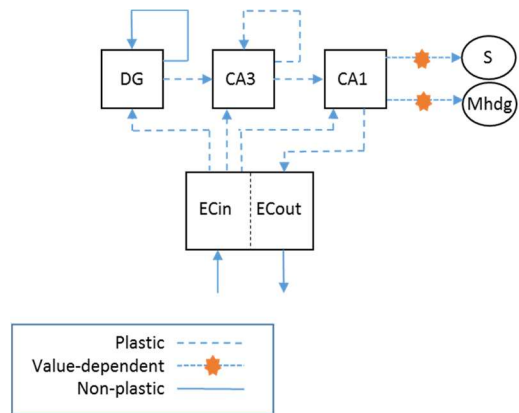


Figure 2.3. Neural connectivity of the medial temporal lobe, including the hippocampus of Darwin XI. Hippocampus: DG, dentate gyrus, and CA3 and CA1. EC: ECin and ECout. Neural interfaces to external sensors: S, value system and Mhdg, motor.

Although both the previous research using Darwin X [57, 58], which used a dry variant of the Morris water maze task [59], and that using Darwin XI, which uses the plus-maze, are performed on rodent based navigation testing platforms, the focus of these studies is on the formation of episodic memory. Using a backtrace analysis tool, several seconds of neuronal activity and synaptic changes can be analyzed to determine causality of a particular neural event. Both studies showed the strongest synaptic influence from the entorhinal neuronal units on episodic memory, particularly from the performant path ($EC_{in} \rightarrow DG$, $EC_{in} \rightarrow CA3$, and $EC_{in} \rightarrow CA1$ in Fig. 2.3), while Darwin XI specifically focused on journey-dependent and journey-independent memory, as well as path prediction. A further detailed analysis can also be found in [60].

2.3.3. Strösslin et al. 2005

(1) System Architecture

Strösslin et al. [48] use the same mobile robot platform (Khepera) as Arleo. The robot has a camera, odometers, and proximity sensors. Thus, the robot only uses body-centric, local sensor information for navigation. The Khepera is attached to an external computer, running the neural model, with a long cable that also provides power to the robot and allows for sensor data to be transmitted from the robot to the computer.

(2) Neural Model: Place Code and Cognitive Maps

In a dry water maze, similar to that used for Darwin X, a navigation map is learned by the PCs in 20 trials, which is similar to the results found with rodents in the water maze [59]. Thus, visual and idiothetic (self-motion cues) information feeds the external neural

model, which is composed of step cells (SCs) and rotation cells (RCs). These cells make up the local view (LV) and are fed by the visual input, a head direction (HD) system in the PoS, PI in the mEC, and combined place code (CPC) in the hippocampus (HPC) and subiculum. The directional action cell (AC) in the nucleus accumbens (NA) is what eventually drives the navigational learning of the CPC. See Fig. 2.4 for connectivity.

The cognitive map or spatial representation of the robot’s environment is accomplished through unsupervised Hebbian learning between the PCs and the HD cells. Additionally, route planning is accomplished by use of biologically inspired reinforcement learning mechanism in continuous state space (place cells) and ACs.

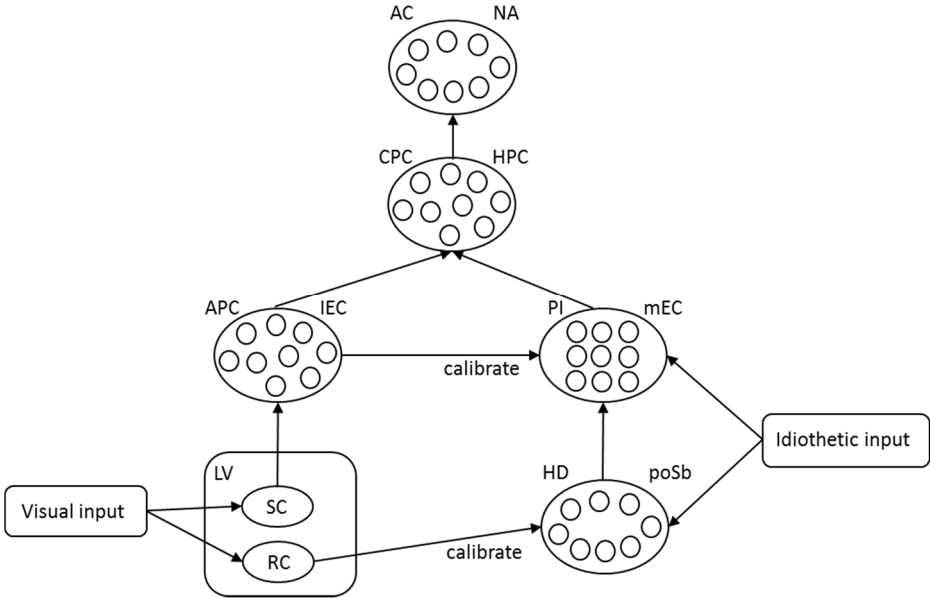


Figure 2.4. Simulated neural system. LV, RCs and SCs, processes, stores and compares visual stimuli. SCs drive the allothetic place code (APC) in the IEC and RCs calibrate the HD system in the PoS. Internal odometric input drives HD and the position integrator (PI) in mEC. APC calibrates PI and they both project to the CPC in the HPC and subiculum. CPC is used for navigation learning on the ACs in the NA. Redrawn with permission from Strösslin et al. [48].

2.3.4. Hafner 2008

(1) Place Code and Cognitive Maps

In [61], Hafner uses PCs for creating a cognitive map of a mobile robot's area. The mobile robot, outfitted with only an omnidirectional camera and a compass, produces a cognitive map during an exploration phase, where the map is represented by place fields and PCs. Each snapshot taken by the camera is converted into a 16-dimensional transformation, which is used as the sensory input to a neural network system. That is, each 360° camera snapshot is divided up into 16 angular, azimuth sections of 22.5° each, filtered, and sent to the PCs' neural networks. The weights of each neural network, initially set to random values, take on evolved values during the exploration phase. The place cells, as shown in the "output layer/map layer" in Fig. 2.5, become relationally connected to each other based on a self-organizing map (SOM) methodology [62], where each single winner of a particular snapshot becomes connected to the previous winner and the corresponding connection weight is increased. Since the PCs are not geometrically fixed, they are assigned relative angles to each other, creating a topological map. This is all done without the use of reward during learning. Additionally, there is no goal state.

(2) Simulated Route Planning

However, once the neural cognitive maps have been built, they can only be used in simulation for navigation. The topological and metric information requires too much memory to reside in the mobile robot [61]. Thus, the mobile robot relies on landmark (snapshot) recognition and use of the SOM to reach goal spots or areas.

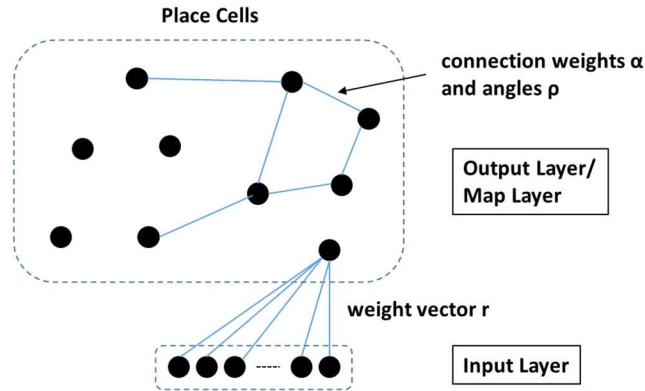


Figure 2.5. Neural network structure as a result of learning connectivity between place cells. The input layer represents input from the robot's sensors [61].

2.3.5. Barrera and Weitzenfeld 2008

(1) System Overview

Barrera and Weitzenfeld [47, 63] propose and implement an intricate, and modular neurophysiological based navigation model. As with Arleo and Gerstner [46, 54], all of the proposed functionalities are mapped back to existing neurophysiological entities. Many of these modules are implemented using Gaussian distribution functions for calculating affordances, and the Hebbian learning rule/equation for neural networks. The main goals of this research are: (1) for the mobile robot to be able to learn and unlearn path selections for goal locations based on changing rewards, (2) to create a realistic neuroscience based test bed for use in further behavior studies, and (3) to add to the existing gap in the SLAM model between mapping and map exploitation [47]. The mobile robot's test environment configurations are limited to the T-maze and the 8-arm radial maze.

The neurophysiological theory that forms the basis for this study comes from [64]. Thus, in addition to idiothetic and allothetic sensory inputs, there are also internal state/incentives (e.g., food seeking and food intake driven by the hypothalamus) and affordances (possible actions to take) information sensory inputs. Fig. 2.6 shows the functional modules of this system, while removing many of the underlying details of the neurophysiological framework. Further details, such as model description, the neurophysiological framework, and equations for each of these modules can be found in [47, 65-67].

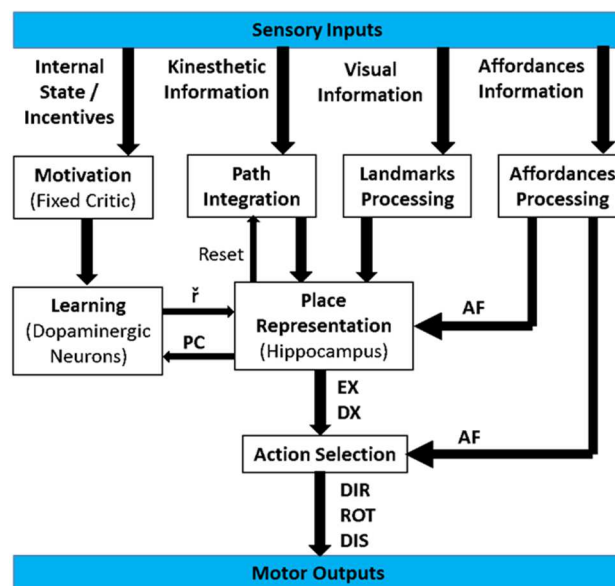


Figure 2.6. Computational spatial cognitive model of the Barrera and Weitzenfeld neurophysiological based mobile robot navigation system. Some submodules and neurophysiological framework are not shown and can be found in [47]. \tilde{r} = effective reinforcement; PC = place information pattern; EX = expectations of maximum reward on their corresponding directions (DX); DIR = next rat direction; ROT = rat rotation; and DIS = next rat moving displacement.

Idiothetic data comes in the form of kinesthetic data, which is processed by PI and then the hippocampus module, before being sent to an external motor control module, via

the Action Selection module, as shown in Fig. 2.6. This is used for executing rotations and translations of the robot.

(2) Place Cells and Cognitive Map Generation

The Place Representation module in Fig. 2.6 is where the cognitive map is made, stored, and accessed for the mobile robot to select movement options. Thus, this module represents the functionality of the hippocampus. The path integration information is combined with landmark information, through the Hebbian learning rule, to create a PC layer. The overlapping PC fields in this layer represent given locations or nodes that are found in the world graph layer (WGL), as shown in Fig. 2.7.

The WGL uses a simple algorithm to decide its next move. It analyzes active nodes connected to the Actor Unit and, based on the highest weight, the WGL chooses the step that will get it closer to its learned goal or the best move for the time when a goal has been changed or not learned yet.

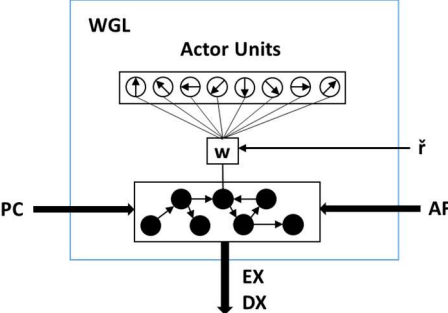


Figure 2.7. World graph layer module which implements a topological map of the mobile robot’s environment inside the Place Representation module.

(3) Computational Resources

Because of the high computational resources required for this neurophysiological based navigation system, most of the model runs on an external 1.8GHz Pentium 4 PC, which communicates wirelessly with a Sony AIBO ERS-210 4-legged robot. Thus, the system is not autonomous.

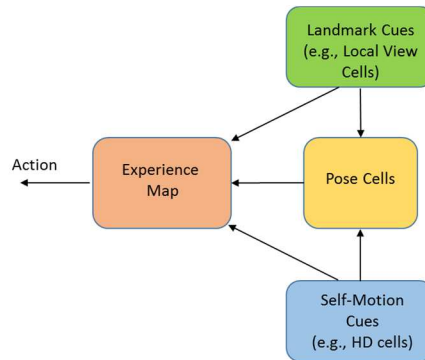


Figure 2.8. Connectivity diagram of the RatSLAM, version 3.

2.3.6. Wyeth and Milford: RatSLAM, Version 3

(1) System Overview

Wyeth and Milford focus in [39, 68] on a neurobiologically inspired, SLAM based, mapping system for a mobile robot navigation system, based on models and earlier versions of RatSLAM [69]. Their robot, a Pioneer 2-DXE base system, performs mock deliveries in a large, single floor, office building using simple sensors: motor encoders for odometry, sonar and laser range finder for collision avoidance and pathway centering, and a panoramic camera system for landmark recognition. This system, named RatSLAM, uses the concept of place cells coupled to head direction (HD) cells to derive, what they call, pose cells.

(2) Pose Cells

The continuous attractor network (CAN) [22, 70] based pose cells are used with local view cells, which are snapshots of the panoramic camera along the robot's journey. Thus, Milford and Wyeth have added a new type of cell: the pose cell. The pose cell is similar to the conjunctive grid cells, which is a combination of grid cells and head direction cells found in the rodent brain. The pose cells work like weighted probabilities that each local view cell is in the direction and location of the stored pose (averaged). Fig. 2.8 illustrates the connectivity of the RatSLAM, version 3, as described here and in [39].

(3) Cognitive Map

The mapping algorithm incorporates a loop closure and map relaxation techniques to correct PI errors, thus creating more of a topological map than a metric map. A loop closure event only occurs when a threshold of consecutive local view cells matches the camera's input, thus allowing for a change in the pose data. To save original pose data, the relaxed map is saved to an "Experience Map" (see Fig. 2.9 for an illustration of the Experience Map Space), and the local view cells with accompanying pose cell data are stored in a connection matrix. Due to the topological nature of the Experience Map, transitions between experiences are stored, thus allowing route planning to be possible.

The benefit that comes from this design is that it is a first step into implementing the functionality of some of the specialized, navigation and spatial awareness, brain cells in a mobile robot. The downside is that it has been shown that the competitive attractor

network can be easily replaced by a filter system [71], which leads to substantial computational speedup.

2.3.7. Cuperlier et al. 2007

(1) Transition Cell

Cuperlier et al. built a neurobiologically inspired mobile robot navigation system in 2007 [72] using a new cell type which they named the “transition cell.” Their cell is based on the concept of moving from one place cell to the next over a defined interval of time. Thus, two place cells are mapped to a single transition cell, creating a cell which represents both position and direction of movement or spatiotemporal transitions, thus a graph-like structure.

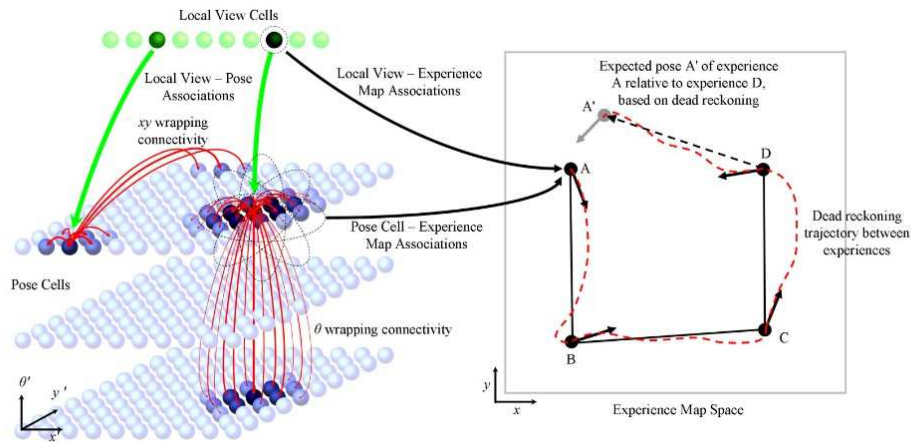


Figure 2.9. The RatSLAM system. The left side represents the CAN system which forms pose cells from local view cells using a 3D CAN algorithm. The right side represents the Experience Map, which helps disambiguate scenes that are similar in a semi-metric form. A further, detailed description can be found [39]. Permission for replication given by Dr. Michael Milford.

(2) Computation Resources

Multiple neural networks span the system's architecture, as shown in Fig. 2.10, from the landmark extraction/recognition stage to the cognitive map and motor transition stages. The many inputs of video, place cells, and so forth into a system of neural networks require many calculations to be carried out during each time step. This computational resource demand is similar to Arleo and Gerstner [46, 54] and Barrera and Weitzenfeld [47, 63, 67], covered in the previous section. To illuminate the amount of processing that is required it is stated in [72] that the system uses 3x Dual Core Pentium 4 Processors which run at 3GHz each. However, the author reports that this processing architecture has since been reduced to a single Intel® Core™ i7 processor, which has 4 cores that run at just over 3GHz. Azimuth angles are measured using an onboard compass, displacement is obtained from wheel encoders, and the visual is obtained from a panoramic camera.

The navigation process starts at the leftmost part of Fig. 2.10, where a single, potential landmark is selected and analyzed at a given time. This occurs up to N times per snapshot, where N is set to a value to help balance the algorithm's efficiency with its robustness. Therefore, as expected in any visual extraction/recognition system, a fair amount of processing time and power is spent during this stage. Additionally, during the initial exploration phase, weighted neural network coefficients are calculated for each potential landmark (32×32 pixels) and azimuth grid value, so that these small local views can be learned online. For more details on the calculations performed to arrive at the place cells from the landmark-azimuth matrix (PrPh) consult [72].

(3) Cognitive Map

Each place cell (center of Fig. 2.10) is connected to each neuron of the landmark-azimuth matrix, where each connection has its own, unique, learned weights for that landmark-azimuth-place cell combination, as well as temporary scalars for the current, potential landmark view. However, it is very likely that several place cells will be active enough at a given location. The paper states that when a whole area has been mapped, during the initial exploration phase, the place cells are divided up into their own areas to eliminate these overlaps (see Fig. 2.11), thus, creating a cognitive map.

An assumption is made about the average number of possible place cell transitions from any particular place cell for the test conducted in [72]. This is done to reduce $N \times N$ neural network based, transition matrix to $6 \times N$, where N represents the number of possible transition place cell targets, thus, greatly reducing the computational complexity from $O(N^2)$ to $O(N)$. However, this value may not work for all test cases, or in-field use.

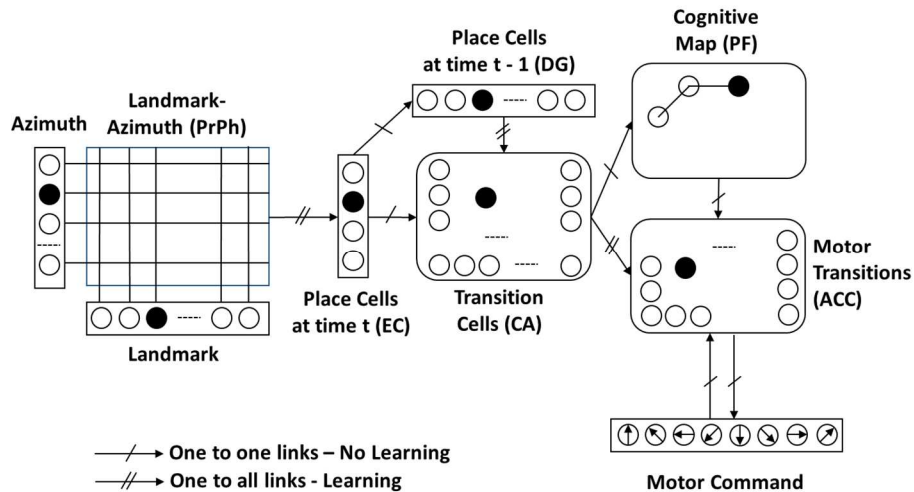


Figure 2.10. The system's neural network based model architecture. Processing flow starts at the far left with the input of each camera snapshot [72].

(4) Route Planning

The robot's cognitive map built during an initial exploration phase, as previously described, consists of nodes and edges, as shown in Fig. 2.12, and is thus a graph: $G = (N, E)$. Each node is a transition cell and an edge signifies that the robot has traveled between the two transition cells or nodes. The edges hold weight value (e.g., function of use) and the nodes hold activity values. The recorded nodes/edges of the cognitive map are used in a neural network version of the Bellman-Ford algorithm [73] to find the most direct route from a motivation point to the single source destination, while several types of motivations (drink, eat, sleep, etc.) are used to initiate the robot's travel to the proper destination source. The satisfaction level of the motivations changes with time and distance traveled, while increasing at the source.

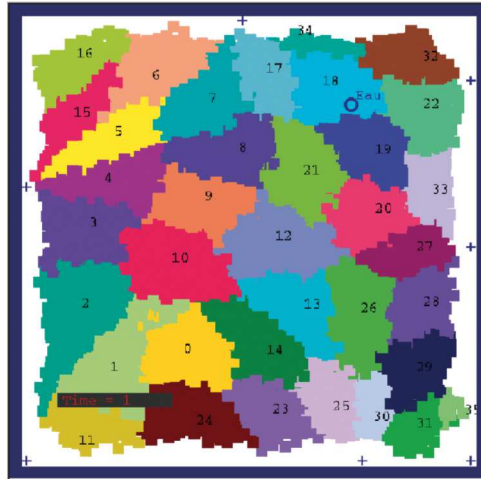


Figure 2.11. Assignment of dedicated place cell fields. Permission for replication given by Dr. Cuperlier et al. [72].

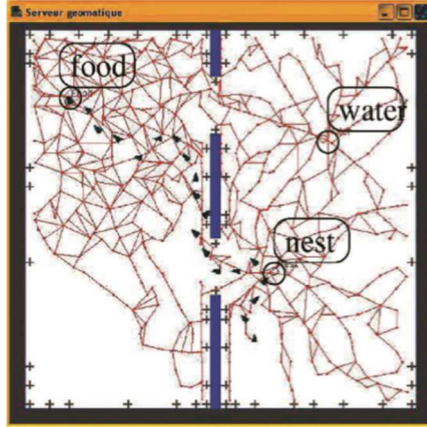


Figure 2.12. Topographical cognitive map in the form of a graph is produced in the system, as illustrated. Permission for replication given by Dr. Cuperlier et al. [72].

2.3.8. Grid Cell Centric Systems

Perhaps since the grid cell was not discovered until 2005, or due to its complex nature and uncertain contribution to navigation, there are a sparse number of robot navigation systems that are based on the grid cell. Instead related research in grid cells comes from computational/oscillational models [25, 74-78].

There are currently two prevailing computational model classes for describing the stimuli configuration required for the grid cell firing pattern. The first is the continuous attractor network (CAN) which follows along the lines of what was covered under the RatSLAM navigational model. The CAN model, which in simplest terms, is a neural network based model that describes the stabilization or convergence of a multistate system to a single state over time, by way of synaptic interaction between excitatory and inhibitory neurons [70, 79]. An example of a CAN model which describes the role of PI in the firing of the GCs is outlined in [80]. The second computational model, is the oscillating interference model [81]. The oscillating interference model is typically

simulated using spiking neural networks on non-robotic systems [74, 77, 78]. Both working models have strong pros and cons to their validity. Further details will be given on the oscillating interference model shortly, since it is a foundational concept that is used in the proposed thesis navigation model.

As covered in the previous section, Milford and Wyeth [39, 68] use pose cells in their neurobiologically based navigation model RatSLAM, which are based on the conjunctive GCs found in the deeper layers of the mEC, as further described in [82, 83]. Additionally, the wrapping connectivity of the pose cell grid creates a GC type firing pattern. However, there is much scientifically backed detail missing pertaining to the functionality of regular, non-conjunctive grid cells found at the superficial layers of the mEC, as well as the specifics of the conjunctive GCs' connectivity based on attributes of scale, orientation, and phase modeled.

Gaussier et al. [84, 85] use a mathematical model of the grid cell for their mobile robot navigation system. The GC's firing pattern is a modulo projection of the PI input. The tests performed on the mobile robot show poor patterns for the grid cell firing when relying on just path integration with growing accumulated errors as expected. Adding visual input to reset and recalibrate the path integration fixes the noisy path integration input, thus sharpening the firing pattern of the grid cells. The system described and tested on a robot in [85] illustrates a well-integrated system composed of a visual system, path integrator, place cells and grid cells. The results obtained from various tests show a promising beginning to a grid cell based system.

(1) Oscillatory Interference Model

As previously touched on, one theory on what causes GCs and PCs to fire is oscillatory interference [76-78]. This model relies on HD cells, based on their preferred direction and the current head direction of the rat, to modulate persistent spiking cells (oscillators) who's frequency is a function of the distance traveled by the rodent over a delta time-period. Each oscillator has a given offset phase and frequency scaler. Each GC is fed by the same input network of oscillators in a neural network layer configuration, then the output of these GCs feed PCs. Thus, the HD cells and oscillators act as a PI system, which feeds the GCs. The oscillatory interference computational model for the implementation presented in [74] is as follows:

$$\phi_{(i,j)}(t) = 2\pi(ft + b_j \int_0^t d_i(\tau) d\tau) \quad (2.1)$$

$$s_{(i,j)}(t) = H(\cos(\phi_{(i,j)}(t) + \psi_{(i,j)}) - s_{thr}) \quad (2.2)$$

$$g_j(t) = \prod_{s \in S_j} s(t) \quad (2.3)$$

where $\phi_{(i,j)}$ is the persistent spiking cell's phase modulated by the i^{th} head direction cell and projecting to the j^{th} grid cell, f is the frequency, b_j is the scaling factor for all persistent spiking cells projecting to the j^{th} grid cell, $s_{(i,j)}$ is the persistent spiking cell signal, ψ is the phase offset, s_{thr} is the threshold, H is the Heaviside function with $H(0) = 0$, g is the grid cell signal, and S_j is the set of persistent spiking cells projecting to the j^{th} grid cell. Fig. 2.13 illustrates the persistent spiking circuit architecture described above and in [74]. From this model, the hexagonal firing pattern of each grid cell (see Fig. 2.14a and Fig. 2.14b), is supposedly created.

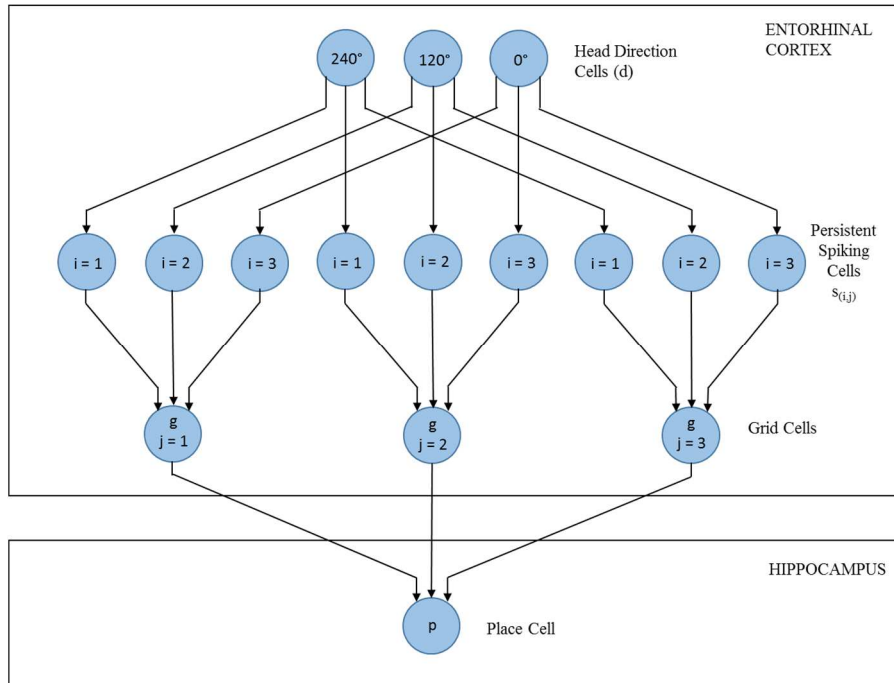


Figure 2.13. Architecture representation of the persistent spiking computational model which drives the selection of GCs and PCs as presented in [74].

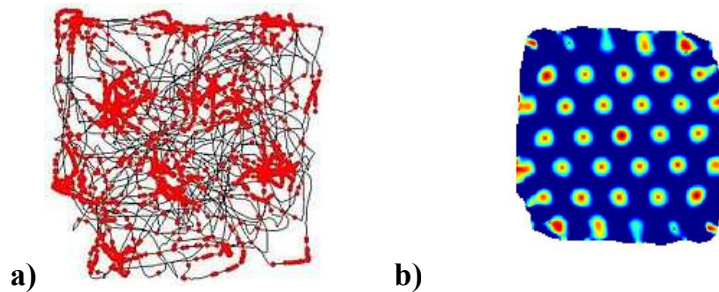


Figure 2.14. Grid cell firing fields. a) Recorded firing locations (red dots) of a single grid cell, as a rat explores (black line) a square, enclosed area. b) The autocorrelogram of the firing data for the grid cell. The hexagonal pattern of the firing locations can be seen in both parts a and b of the figure [86].

2.4 Analysis of Reviewed Systems' Localization

The visual capabilities of the reviewed material, as listed in Table 2.1, play an important role in the localization accuracy, as well as place field mapping, of these systems in their environment. Most of these neurological based navigation systems [39, 46, 48, 61, 68, 72] use a camera that has a near to full 360° field of view (FOV). This is

accomplished using a spherical lens or by the robot performing multiple in-place rotations to obtain connecting/overlapping snapshots. The Barrera and Weitzenfeld [47, 63] navigation system uses a camera with a narrow FOV, however, the uniquely color coded landmark indicators are positioned at the end of corridors of a movement restricted maze. Thus, the visual data obtained is processed by various image processing algorithms and stages, then fed into neural networks, along with heading/azimuth data so that locations in the environment can be uniquely identified, via place field assignment. The use of global, allocentric data can greatly reduce PI error and increase overall localization accuracy when used along with movement and rotation estimation ANNs, as demonstrated in [48], and checked against a compass or similar [85]. This accuracy does cost processing time and computational resources, as presented in these systems, due to their integration into ANNs. However, these types of navigations systems very much adhere to a neurophysiological based architecture.

Quantifying the accuracy of the localization and actual processing/memory requirements is difficult, if not impossible, to obtain from the literature alone. However, omnidirectional visual processing is a visual type of triangulation, which can be very accurate (even more so than civilian grade GPS systems [87]), particularly as landmarks are plentiful and/or close. Correct heading information (azimuth data) is stored with the visual processed data. This is required to prevent perceptual aliasing. An issue can still arise with this form of localization if images vary due to different sensory information at different points in time (e.g., illumination, noise, etc.). Additionally, such systems will not work in areas of very limited visual cues, whether due to lack of light or lack of landmarks.

Table 2.1. Neurobiologically based navigation research.

Authors/Articles	Platform/Sensors	Visual Capabilities	Brain Cells Emulated	Cognitive Map	Route Planning and Autonomy (*)
Arleo; Gerstner [46]	1) Khepera mobile robot. 2) 8 IR sensors – Obstacle detection. 3) Light detector – Ambient light measure. 4) Camera 90° H – Self-localization. 5) Odometer – Self-motion	- Offline, unsupervised, Hebbian learning, network (NN) training. - Four 90° horizontal snapshots taken (N, W, S, E) to create a single, location recognizable view. - Used primarily to assist with robot NN directionality.	- Place cells and - Head direction (HD) cells	Built into NNs of place cells & head direction cells. (Use of external homing light and offline NNs).	*Use of external computer, thus not autonomous.
Fleischer et al. [56]	1) BBD – Beowulf cluster 2) Robot platform: a) CCD Camera b) Compass c) Laser range finder d) Whisker system e) Odometer	- Transformation of RGB video data (320 x 240 pixels) to YUV color space on one of the cluster computers. After some processing, interfacing of color neuronal units to inferotemporal cortex, and edge units to parietal cortex.	- Place cells. - Dentate gyrus. - EC and other medial temporal lobe cells.	Limited movement in plus-maze. Directional choices at intersection is learned by place cells in the hippocampus.	-Route retrospective and prospective responses/planning are shown in backtrace analysis. *Not autonomous due to external BBD.
Strösslin et al. [48]	1) Khepera mobile robot. 2) Camera 60° H FOV. 3) Odometers. 4) Proximity sensors.	- Simulates rodent's FOV by rotating camera 4 times to obtain 240° FOV image. - Extracts directional information from visual inputs. - Path integration through visual and self-motion information.	- Place cells and - HD cells - Action-cells, located in dentate gyrus. - Many neurophysiological based elements.	Combined place code (CPC) neurons, where visual and odometric information are stored.	Biologically inspired reinforcement learning mechanism in continuous state space. *Not autonomous due to use of external PC.
Hafner [61]	1) Omnidirectional camera 2) Compass.	- 360° snapshot divided into 16 segments. Input into place cell NN, thus assists with robot's position determination.	- Place cells.	Topological map- Relational navigation connections between place cells.	Can only be performed in simulations due to the amount of metric data processing required. *Not autonomous.
Barrera & Weitzenfeld [47, 63]	1) Sony AIBO, 4-legged robot. 2) Camera 50° H 3) Limited turns in increments of +/- 45°. 4) External PC w/ 1.8 GHz Pentium 4 Processor. Runs nav. model and connects. wirelessly to AIBO robot.	- Simple color recognition representing landmarks and goal. - Distance extracted from images of engineered environment and known relations.	- Places cells & many neurophysiological based elements.	Place cells (nodes) and connections (edges). Simple T-maze and 8-arm maze.	Ability to learn and unlearn goal locations. *Not autonomous due to use of external PC.

Table 2.1 (continued). Neurobiologically based navigation research.

Authors/Articles	Platform/Sensors	Visual Capabilities	Brain Cells Emulated	Cognitive Map	Route Planning and Autonomy (*)
Wyeth & Milford [39, 68]	<ol style="list-style-type: none"> 1) Pioneer 2-DXE robot. 2) Motor encoders – Odometry 3) Sonar & laser range finder – Collision avoidance & pathway centering. 4) Panoramic camera syst. – Landmark recognition. 	- 360° snapshot. Each unique snapshot is stored as a local view cell (VC) for landmark recognition.	- Place cell & head direction cell combined as a pose cell.	A cognitive map is stored in an experience map. The map is created from the pose cells in the continuous attractor network (CAN).	Office delivery locations are stored in the mobile robot, which uses the experience map and CAN to make deliveries. *Autonomous.
Cuplier et al. [72]	<ol style="list-style-type: none"> 1) Robot with 3x Dual Core Pentium Processors (3 GHz each). 2) Panoramic camera. 3) Compass to measure azimuth angles. 4) Wheel encoders. 	- 360° snapshot taken at low resolution and image is convolved using difference of Gaussian (DoG) to detect characteristic points (Landmark recognition).	- Place cells coupled together to create transition cells.	Topological map. Created online during initial exploration phase: images and directions used to create place cells which are then used to create trans. cells.	Use of the Bellman-Ford algorithm to choose most direct route from the cognitive map (transition cells with weighted links). *Autonomous.

CHAPTER 3: SENSORY INPUT

3.1 Idiothetic Sensors for Path Integration Model

Removing the neurophysiological implementation details results in a simplified functional module that can replace the GCs and the path integrator neuron circuitry. The PI related inputs to the persistent spiking model are the HD cells and distance traveled. This information is replaced by heading information from a MEMS gyro (vestibular data), and distance traveled information in delta time ($\int_0^t di(\tau) d\tau$). The distance traveled, assuming straight segment movements by the robot, is captured by the motors' encoders (proprioceptive data). Therefore, the travel vector that emerges from the path integrator/GCs module, in moving from one point in the environment to another, is of magnitude d and at heading θ , or vector $\mathbf{d} = (d, \theta)$. The travel vectors of the mobile robot presented in this paper are acquired and transformed into Cartesian coordinates by a microcontroller. Our model maps the firing characteristics of a GC to a Cartesian coordinate system. Being that GCs are neural network based, this might not be a perfect one-to-one comparison, however, in a top-level view, there are many similarities.

In our system, the x, y coordinates for an internally stored Cartesian based map, are found by using the sine and cosine functions on the mobile robot's tracked allocentric heading θ . This is similar to the cybernetic models of PI found in [88], and also presented by Mittelstaedt as described in [9]. The travel vector to coordinate equation used is as follows:

$$x_k = d_k \sin(\theta_k) + x_{k-1} \quad (3.1)$$

$$y_k = d_k \cos(\theta_k) + y_{k-1} \quad (3.2)$$

The terms x_{k-1} and y_{k-1} represent the Cartesian coordinate of the robot's last stop/turn. For $k=0$, the values of these terms, (x_{-1}, y_{-1}) , are defined as $(0, 0)$. This which represents the initial starting location (home) of the robot. Fig. 3.1 shows the graph assignment with respect to "home" and an initial allocentric bearing of 90° ($\theta_0 = 0^\circ$ for the robot's internal calculations).

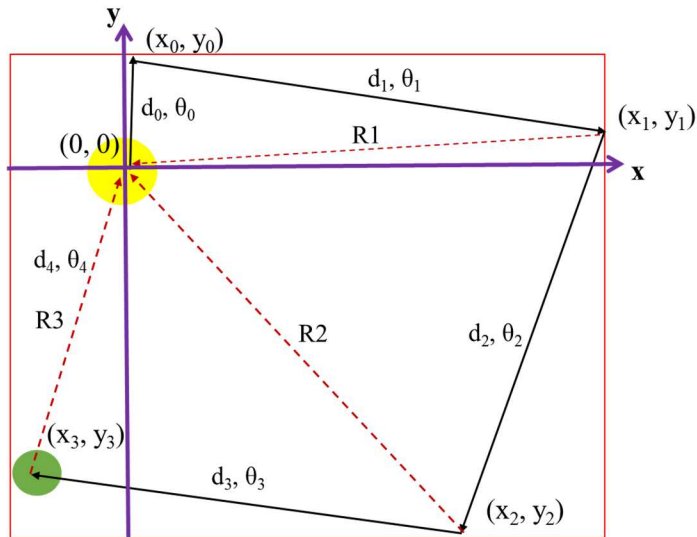


Figure 3.1. A conceptual overlay of an internal Cartesian graph representation of the *ratbot's* navigation environment. The yellow circle represents the *ratbot's* home and starting location, while the green circle is a goal location (e.g., food or water). The PI algorithm always assumes the starting position to be at the origin $(0, 0)$ of an imaginary graph. The black vectors represent the robot's path, while the red vectors (Rn) represent calculated homing vectors. θ_i is the *ratbot's* allocentric heading.

Mobile robots are prone to systematic PI errors such as unequal wheel diameters, imprecision in odometry and direction measurements, as well as non-systematic errors, such as floor slippage and uneven floors. The result is an accumulation of PI error over time [89]. As expected, these errors have a major impact on mapping and localization [90].

3.1.1 Heading Sensor

A pertinent example of a systematic accumulated PI related error is found with our mobile robot's gyroscope. Our robot is here after referred to as the *ratbot*. The *ratbot* uses the InvenSense MPU6050, MEMS - 6 axis, accelerometer and gyroscope. Particularly, the yaw rotational axis of the gyroscope is used to determine the robot's heading. With MEMS based gyroscopes, however, there is a relatively constant drift. To compensate for this drift, the gyroscope measurement data is sampled in a loop at the beginning of the robot's main program, from which an average drift rate is derived. This drift rate is subtracted from all future reads from the MEMS gyro. However, since the drift rate is not perfectly constant, this value will slowly drift as well. The graph in Fig. 3.2 shows how the measured heading still drifts when the gyroscope is stationary over a 12-minute interval. The drift is approximately 15 degrees in this time frame, which works out to be about 0.02 degrees/second. The initial, uncompensated drift rate was measured at 0.47 degrees/second. This, of course is just the static error. There are three phases of movement during the turning of the robot: (1) initial acceleration, (2) constant velocity, and (3) deceleration. Since the turn rate is a rotational velocity measurement, there will be rate averaging occurring over these three phases. When the sampling occurs during these phases, and the duration of the time frame that the sample is used, will no doubt be a source of additional heading measurement error.

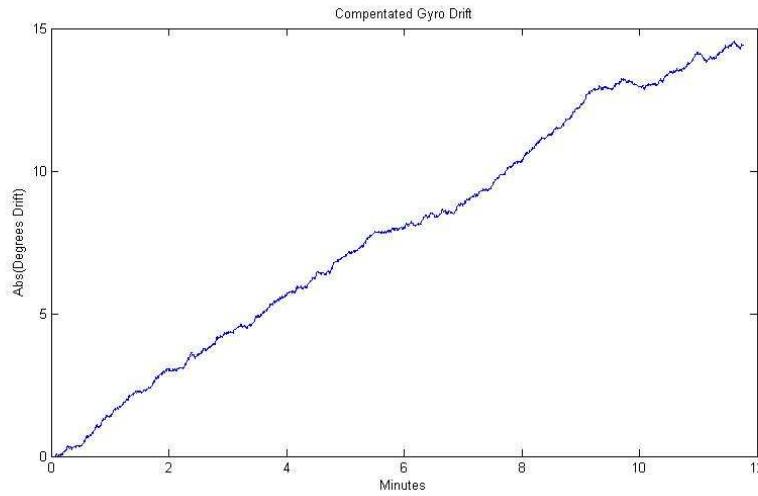


Figure 3.2. MPU6050 post drift compensated gyroscope data.

The allothetic heading θ from the gyroscope is calculated as follows:

$$\theta = \theta_{\text{prev}} + (\omega - \omega_d) * \Delta t \quad (3.3)$$

where, θ is the current heading, θ_{prev} is the previous heading, ω is the measured gyro rate of change (16 bit A/D value) at Δt microseconds after the previous gyro rate measurement, and ω_d is the drift rate of the sensor (measured average at startup).

As with rodents and other animals, PI error is reset by observing known external distal cues, which allows them to become certain again of their local or global location [34, 91, 92]. Autonomous systems have found that using sensors that capture allothetic stimuli, such as visual recognition hardware and software, greatly helps with this area [39, 48, 93]. Therefore, the use of some form of allothetic based system on the mobile robot is imperative to its autonomous capabilities. The allothetic sensors used on the *ratbot* are covered in section 3.2.

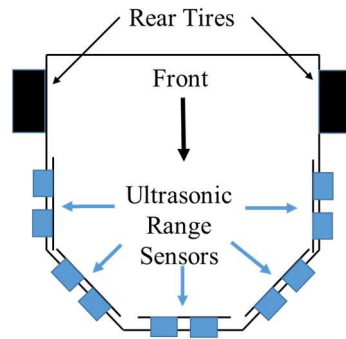
3.1.2 Motor Encoders

The ratbot uses two Devantech 12V, 30:1 gear motors with encoders. A Daventech MD25 motor controller board is connected to these motors for digital control of H-bridge motor drivers, as well as data acquisition (i.e., motor encoder values, supply voltage level, etc.) via a PIC microcontroller. The main controller of the *ratbot* sends and receives data to/from the MD25 to regulate the movement of the *ratbot* over a serial communication interface, and collect encoder values to derive distance traveled. The encoder values are summed by the MD25's PIC microcontroller over time, and can be zeroed out at any time. The encoder values collected are in degrees of wheel rotation at a resolution of 2 degrees.

3.2 Allothetic Sensors

3.2.1 Ultrasonic Range Sensors

The *ratbot* is equipped with five ultrasonic (sonar) sensors to achieve an object detection coverage of approximately 180°, as shown in Fig. 3.3. These sensors are located around the front of the *ratbot*: one forward, a pair of left/right angled “whiskers”, and a pair of left/right side facing ultrasonic sensors.



Top View of Robot Platform

Figure 3.3. Ultrasonic range sensors covering the front of the ratbot. Five sensors giving a forward 180° field of view coverage.

Table 3.1. Electrical and Mechanical Specifications of the HC-SR04 Ultrasonic Range Sensor.

Working Voltage	DC 5V
Working Current	15mA
Maximum Range	4m
Minimum Range	2cm
Beam Angle	20 degrees off axis (3 dB)
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	TTL pulse width in proportion to target range
Dimension	45x20x15mm
Ultrasonic Frequency	40 kHz

The ultrasonic sensors used on the *ratbot* are the HC-SR04 Ultrasonic Range Sensor. The working specifications for this sensor is listed in Table 3.1. The ultrasonic sensor is not as fast/responsive (speed of sound vs. light), nor as accurate as an optic range finder.

Additionally, the beam width of the ultrasonic sensor is much wider and doesn't have the same range capabilities as an optic range finder. However, the ultrasonic sensor does have the advantage of not being affected by the color and texture of the target. For object detection at relatively short distance, the target offset error due to the larger beam width is reduced. The ultrasonic sensor's beam angle is defined as the total angle, where the sound pressure level of the main beam has been reduced by 3dB (half power) on both parts of the center axis, represented here by θ . This angle is obtained using Chart No. 67 from Acoustic Design Charts, replicated in Fig. 3.4, based on the results of the following equations:

$$\lambda = c/f = 343 \text{ m/s} / 40\text{k cycles/s} = 8.6 \text{ mm} \quad (3.4)$$

$$D/\lambda = 13\text{mm}/8.6\text{mm} = 1.5 \quad (3.5)$$

where, wavelength of the sound pulse λ is equal to the speed of sound c divided by the pulse frequency f . The ratio of the diameter of the round transmitter (infinite planar baffle) to the wavelength, determines the sound beam's width or angle θ of 20° at 3dB.

The ultrasonic sensors are used for object/boundary detection and avoidance. The data collected from these range sensors, along with pose data, are used for BC FF activation/initialization, which becomes part of the navigation system's cognitive map.

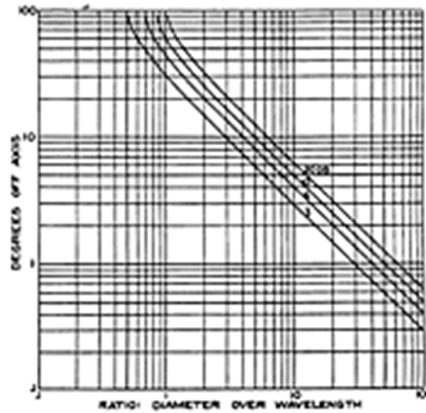


Figure 3.4. Chart No. 67 from Acoustic Design Charts [94].

The ultrasonic sensor is unable to collect the level of detail needed to replace a visual system. Particularly, the details required to identify landmarks and goals, and thus perform a recalibration/reset of the PI error. For this, the *ratbot* uses a visual system and a slightly engineered environment.

3.2.1 Visual System

The *ratbot* uses the Pixy Cam (CMUcam5) from Charmed Labs for landmark and goal location recognition. The Pixy Cam has the capability to swivel on a two degrees of freedom platform, via two mini servos. One servo rotates the camera along the horizontal axis, while another servo rotates the camera along the vertical axis. Currently, the camera is used in a stationary position, pointing directly forward and downwards at a 40° angle with the parallel plane of the *ratbot's* platform. The camera lens FOV is 75° horizontal and 47° vertical. Fig. 3.5 illustrates this configuration of the *ratbot's* camera.

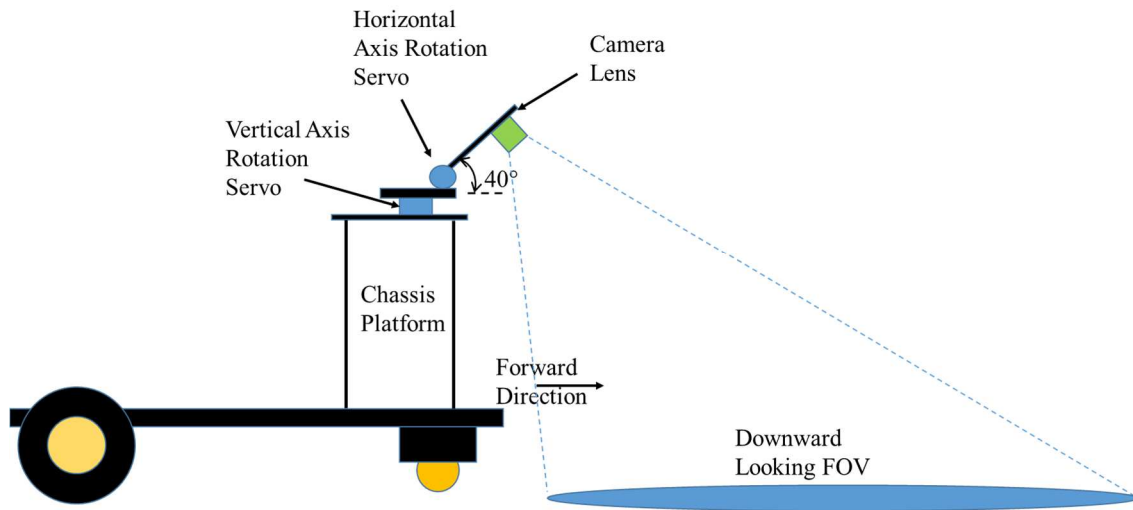


Figure 3.5. The ratbot's Pixy Cam downward looking FOV. Illustration is a view from the ratbot's right side.

Instead of using neural networks and vision data compression algorithms to record and compare gathered visual data to, as was covered in the review section, the Pixy Cam identifies objects by color, using a connected components algorithm to determine where one object begins and another ends. Additionally, using more than one color placed next to each other (color code), allows for many more objects to be uniquely identify. For the *ratbot's* environment, goal places (i.e., home, water and food), and unique landmark locations are marked by color coded paper. An example of this is shown in Fig. 3.6. These color codes are pre-programmed into the Pixy Cam's flash memory using the PixyMon application.

Therefore, a tradeoff is made between having an ANN based visual recognition system, which doesn't need an engineered environment and works with distal salient cues, versus using a simple color-code based system with narrow, local visual capabilities only, which requires very little processing power and resources, but a slightly engineered

environment. Since the aim of this paper is to test the core of the navigation system, the actual vision system used is of no consequence. However, the processor onboard the Pixy Cam does calculate relative X, Y position data with respect to the object's location in the camera's field of view. Additionally, the angle of the color-code image, with respect to the axis running between the two or more colors, is calculated and is available for use. This is displayed as ϕ in Fig. 3.6b. Therefore, the color-coded object's pose can be translated into an allocentric pose, based on the robot's current pose data.

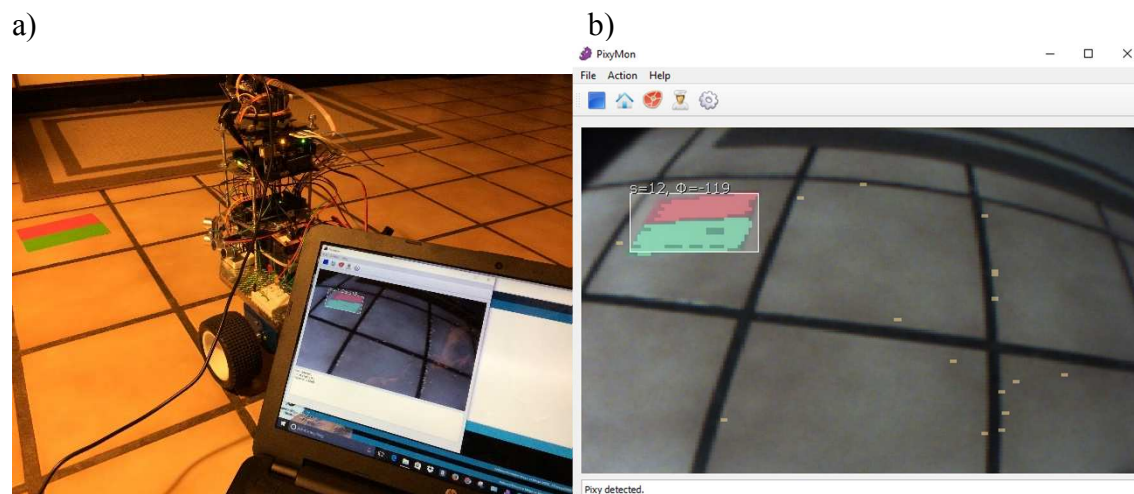


Figure 3.6. Demonstration of the Pixy camera. a) The ratbot's Pixy camera is connected to a laptop to demonstrate what the camera sees. The color code card (red and green) represents a preprogrammed goal or landmark that has been recognized. b) A screen shot of the PixyMon program, which is used to program the color codes and/or display what the Pixy camera sees.

CHAPTER 4: NEW MULTIMODAL PLACE CELL MODEL

4.1 Multimodal Place Cell Model Basics

As proposed in [23], GCs and PCs are speculated to not be a concurrent hierarchy, but complementary. Additionally, BCs have a great influence on the creation of PC fields [31, 95]. Plus, adding the fact that taxon navigation takes place by visual input only works into a newly derived model by this paper as to how PCs are activated. As is illustrated in Fig. 4.1, there are three parallel sources which feed the input to the PCs. Firstly, as illustrated and described in [23], active BVCs (simply stated as BC in this paper) in the rodent's brain for a particular environment can source a PC to fire near the intersection of two boundaries, or an internal corner in a boundary. Similarly, our model produces place fields at the ends of boundaries. Secondly, unique locations, such as landmarks and goals locations, can be learned from the visual data, thus creating place fields which can be used to help reduce PI error. Thirdly, the metric/coordinate based system used by the *ratbot* to map out its environment is similar to the function of GCs in the rodent's brain, which sources the activation of PCs for cognitive map generation, (i.e., place code), as well as allow for PI in complete darkness (no allothetic stimuli).

4.2 Logical Architecture of Multimodal Place Cell Model

Further logic details on the multimodal PC model implemented in the *ratbot* are presented in Fig. 4.2. As will be covered in the FPGA and software design sections in the next chapter, BCs are identified in a logic block that analyzes data gathered from the sides

and front ultrasonic range sensors. This BC/PC determination logic also identifies PC fields that are assigned at boundary corners (C), and at the open ends of boundaries, when they are used as a FF for a turn cell (TC), as will be described in the next chapter. Equivalently, the visual place cell (VPC) determination logic identifies PC fields that occur at goal locations (G) and landmarks (L) from visual data gathered from the *ratbot*'s camera. The exact cases for BC activation are illustrated in Fig. 4.3, while the cases for PC activation are show in Fig. 4.4.

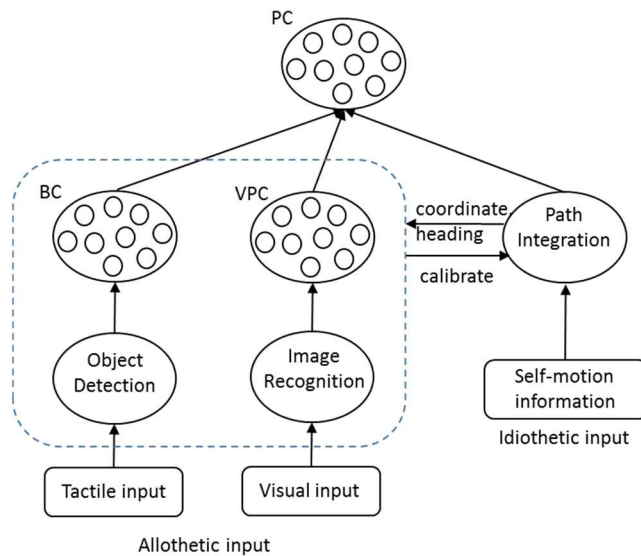


Figure 4.1. The proposed multimodal model of the PC firing field sources. The cognitive map located in the FPGA will possess BCs, VPCs and PCs. The output from the PI source to the PCs represents the interaction of PI data with GCs (pseudo coordinate data), which in turn enable the place fields.

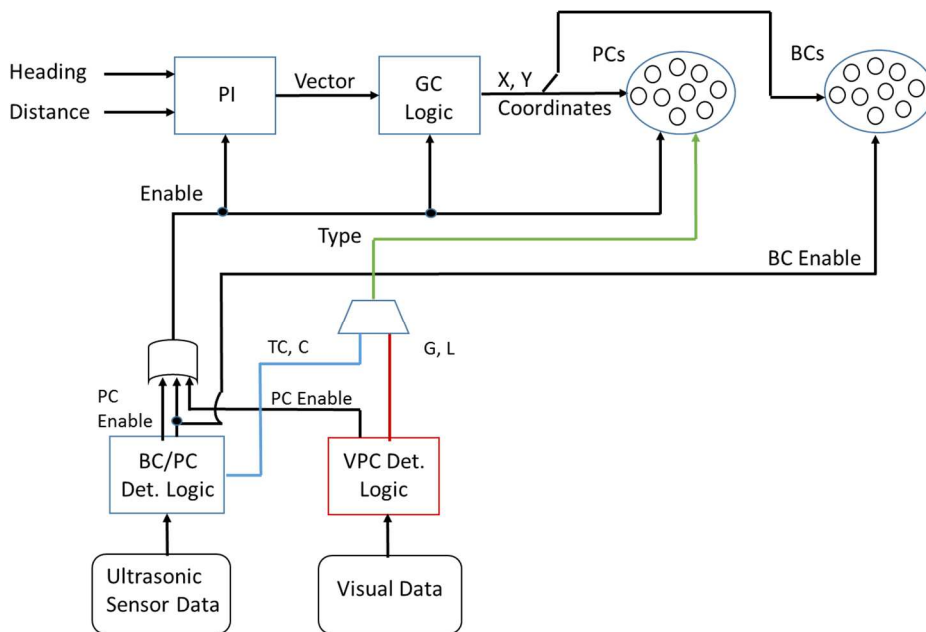


Figure 4.2. The logic architecture of the multimodal PC model. The VPC determination logic block sends goal (G) or landmark (L) type indicators to the PC data structures, while the BC/PC determination logic block sends TC or corner C type indicators. These logic blocks are evaluated sequentially and enable coordinate data to the PC data structures, as well as the PC type, on condition of PC found.

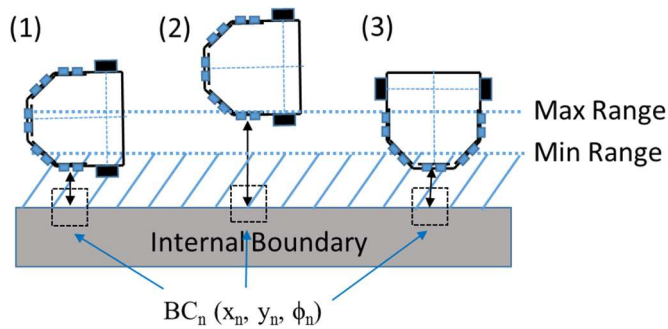


Figure 4.3. BC activation cases. A BC is activated in the following cases: (1) an internal object is detected by either side ultrasonic range sensors within the minimum distance range, (2) or between the maximum and minimum distance ranges, or (3) when the front sonic ultrasonic sensor measures the distance to an internal object within the minimum distance range.

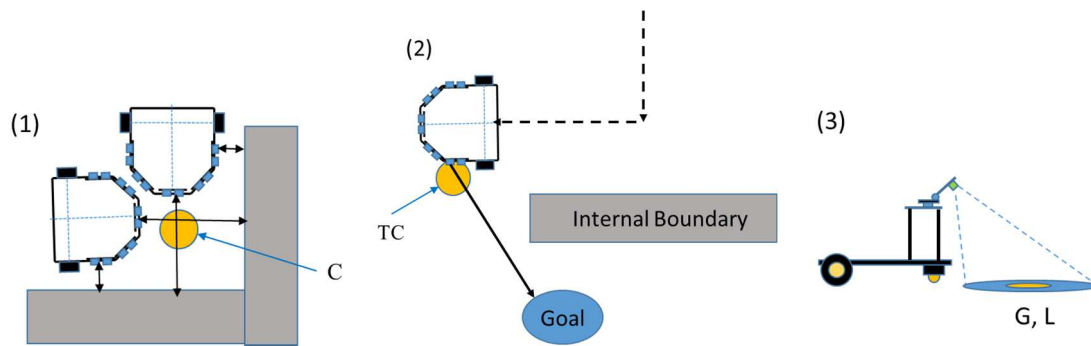


Figure 4.4. PC activation cases. (1) The C type PC is activated at an internal boundary corner, (2) the TC type PC is activated when the robot clears a corner of an internal boundary, and (3) the G and L type PCs are activated when they are initially found and identified by the robot via its camera.

CHAPTER 5: NAVIGATION SYSTEM IMPLEMENTATION

To better understand how the new multimodal place cell is integrated into our navigation system, we will first briefly describe the hardware connectivity and data flow of the total system. Then the software implementation of the model, as well as the general architecture of the navigation system is covered. Finally, we present the cognitive map and spatial awareness created using the central processor with an FPGA.

5.1 Hardware System Design

The main agent of the *ratbot* is the central processor board, an Arduino Mega 2560, which uses an Atmel® ATmega2560 microcontroller, and is integrated to the external sensors and actuators previously covered. The ATmega2560 microcontroller is limited to 256 kbytes of program memory and operates at 16 MHz. Additionally, the central processor board uses many of its 54-digital input/output pins and four serial ports to gather data from sensors, communicate with another microcontroller board, which is connected to the Pixy Cam, and communicate with the motor controller board, as show in Fig. 5.1 and Fig. 5.2. Thus, the central processor gathers data about the environment through the ultrasonic range sensors, camera, motor encoder data, and MEMs based gyroscope (via I2C bus), and makes decisions on the next action to take, based on the sensor data and its current motivation state.

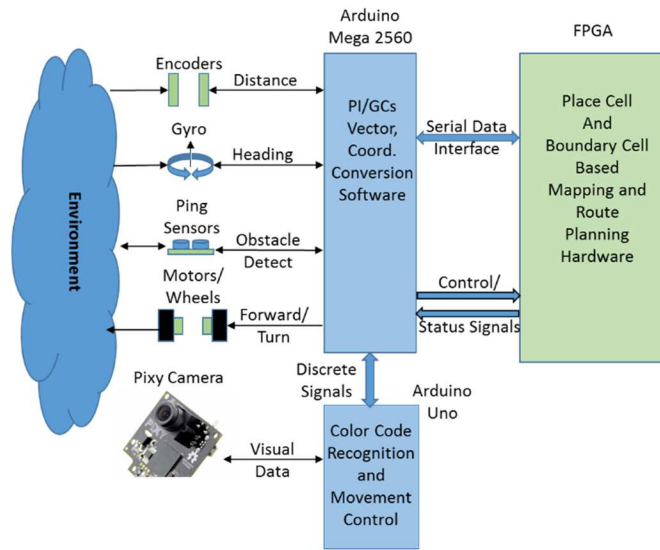


Figure 5.1. Top-level block diagram of the ratbot's neurobiological based navigation system. Shown are the ratbot's sensors, actuators and computational resources.

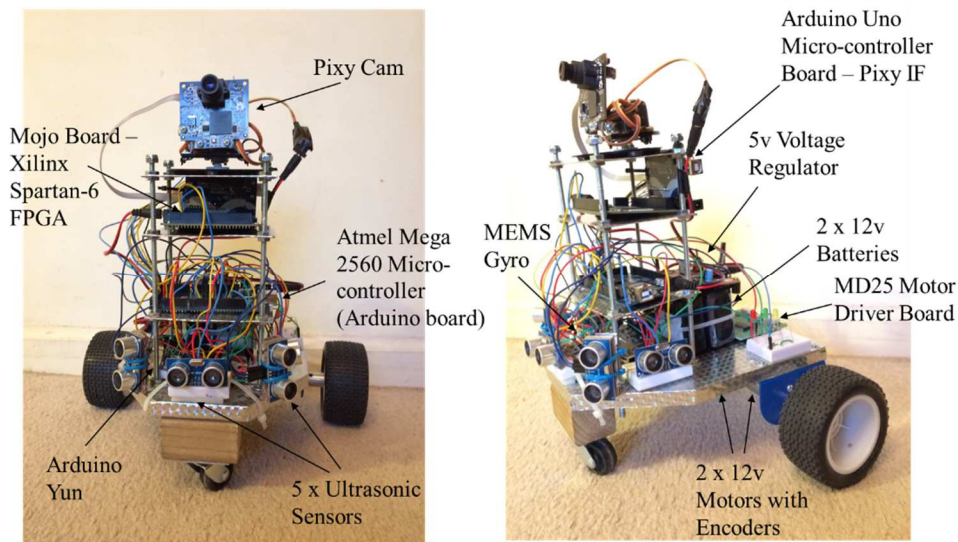


Figure 5.2. The ratbot and its hardware.

The basic decision making of the core multimodal PC model, illustrated in Fig. 4.1 and Fig. 4.2, is carried out in the central processor. Possible new BC, VPC and PC FFs are identified in the central processor's main loop program. The pseudo code for the main loop program is listed in Fig. 5.3. The data from newly identified BC FFs are sent from the

central processor to the FPGA through a serial interface, while PC FFs are stored in the central processor's memory.

-
1. While MotivationState \neq done:
 - a. Check visual data gathered from camera for objects recognized.
 - b. If (Identified Object[k] AND Searching for Object[k]) then
 - i. Go to Object[k];
 - ii. Record/Verify Object[k] (VPC/PC) /* in CPM */
 - iii. Take action based on MotivationState AND ObjectType;
 - c. Get sonar data (distances of objects) from all five sensors.
 - d. Based on MotivationState and barrier(s) distance(s)/location(s):
 - i. Record/Verify BC or PC /* in FPGA or CPM respectively */
 - ii. Take action. /* e.g., stop, turn, go forward ... */
 - e. Check MotivationState for change.
-

Figure 5.3. Central processor's main loop pseudo code. CPM is the central processor's memory.

5.2 Software Design

Further details of the central processor's main loop program, as generalized in Fig. 5.3, are covered here. As can be seen from Fig. 5.3, the action that takes place by the actuators (motors) of the *ratbot* is a function of the agent's motivation state. Thus, the motivation state integrates with the core multimodal navigation model, and is influenced by the current state of the environment. The motivation states of the *ratbot*'s navigation model include being: hungry, thirsty, tired, lost, a predator threat, and curiosity (explore mode). The explore mode is the initial motivation state of the *ratbot*. The *ratbot* randomly navigates its environment, while mapping the area using BCs, VPCs and PCs as described by the multimodal model. This phase continues until the *ratbot* has discovered the food and water goal locations, and saved the path information between the goals and home. Table 5.1 lists the logical steps/cases that the taxon navigation block of the *ratbot*'s navigation software system, shown in Fig. 5.4, uses in the explore mode. The specific steps of the explore phase are as follows:

- a) Random exploration until a goal location is found (water or food).
- b) Dedicate a place cell to the goal location (store coordinates).
- c) Return home directly (single vector home) or via a scan/backtrack algorithm if path is blocked by barrier (adding/dedicate boundary vector cells, recording direction of barrier and coordinates, and turn cell/place cell for transition point around barrier, recording coordinates and ID).
- d) Save place cell path in linked list or similar (example G1->PC1->G0). Save length of path as weight for this path. Goal memory (e.g., nucleus accumbens).
- e) Go back to step (a), unless all goal locations have been found.

The navigation system block diagrams in the review section (i.e., Fig. 2.2, 2.3, 2.4 and 2.6) are relatively similar in respect to the flow of sensor data to motor/action output, and the involvement of the rodent spatial awareness and navigation neurons. Our software block diagram is similar to the computational spatial cognitive model of the Barrera and Weitzenfeld neurophysiological based mobile robot navigation system shown in Fig. 2.6. Fig. 5.4 illustrates the block diagram of the *ratbot*'s navigation software system.

Table 5.1: Visual Part of Exploration Mode

Visually Recognized	Explore Sub-Mode	Goal Searched For	Current Action To Take	Next Action To Take	Goal To Return To	New Goal Searched For	Status Sent to InternalStateGen (Motivation State Satisfied)
Water	Free Run	Any	Go to goal and Record PC data	Return Home	Water	No Change - Any	False
Water	Return to Goal	Any	Go to goal	Free Run	None	Food	False
Water	Free Run	Food	Ignore	Free Run	None	No Change - Food	False
Water	Free Run	Water	Go to goal and Record PC data	Return Home	Water	No Change - Water	False
Water	Return to Goal	Water	Go to goal	Finished	None	None	True
Food	Free Run	Any	Go to goal and Record PC data	Return Home	Food	No Change - Any	False
Food	Return to Goal	Any	Go to goal	Free Run	None	Water	False
Food	Free Run	Water	Ignore	Free Run	None	No Change - Water	False
Food	Free Run	Food	Go to goal and Record PC data	Return Home	Food	No Change - Food	False
Food	Return to Goal	Food	Go to goal	Finished	None	None	True
Landmark	Free Run	N/A	Go to landmark	Free Run	None	No Change	False
Landmark	Other	N/A	Ignore (for now).	Other	None	No Change	False
Home	Free Run	N/A	Go to home if LoC is Low to reset PI	Free Run	None	No Change	False
Home	Return Home	Any, Water, Food	Got to home, Reset PI	Function of Goal to Return to	No Change	No Change	False

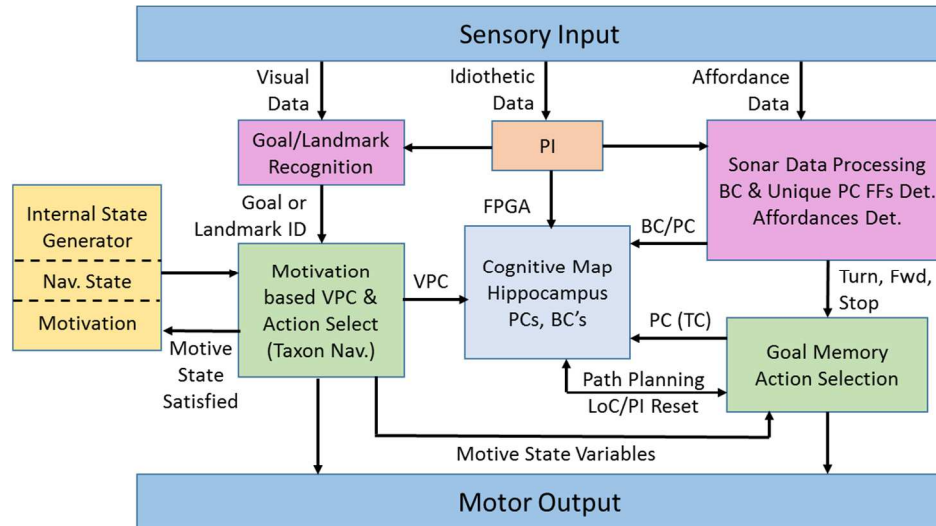


Figure 5.4. Software block diagram of the *ratbot's* neurophysiological based navigation system.

(1) Place Cells in the Central Processor

A PC is activated and assigned the coordinates of a goal area as they are found (e.g., home, food, water, etc.). Additionally, PCs are assigned to turning points (e.g., boundary edges/ends) to help with remembering a path between goals. This is based on the fact that a greater number of smaller PC fields are found near boundaries and objects [23, 61]. These types of PCs are designated as “turn cells” (TCs). The TCs are illustrated in the Ratbot Simulator output shown in Fig. 5.5. Thus, the TCs are used in route planning when the *ratbot* is following remembered paths, or performing a look-ahead feature with the map data. Additionally, as previously covered in chapter 4, there are additional PC types of C, L and G. The data structure of a single PC module, as described here and in chapter 4, is shown in Fig. 5.6.

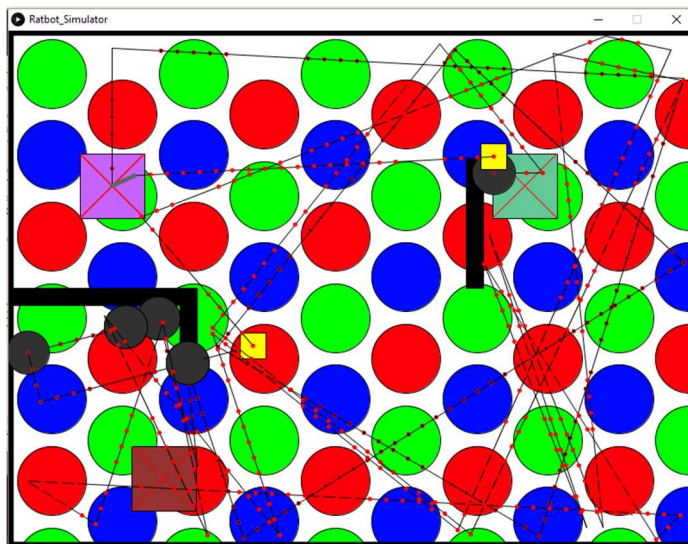


Figure 5.5. Ratbot Simulator output. PCs are used for goal locations (large squares) and as turn cells (yellow squares). BCs (grey circles) are used for path planning and PI error detection. Three GC's firing fields, with different spatial phases, are shown with red, green and blue circles.

PC Data Structure
- ID
- x, y coords
- active/inactive
- type (IC, C, G, L)
Pointers to other PC structures and their Euclidean distances.

Figure 5.6. The PC data structure. Many of these are stored in the central processor's memory.

Besides the type of PC and its coordinates in the *ratbot's* coordinate system, a set number of pointers to other PC structures are included in the PC data structure. As paths are discovered between one goal location to another, the PC FFs of the goals, as well as intermediate PC FFs, such as TC FFs are linked together to remember these paths. Additionally, the Euclidean distance between each pair of place fields is also stored in this data structure. Thus, goal memory represents the use of this navigation data for path

remembrance and execution. Additionally, new paths can be found by using the A* path planning algorithm on a global graph that links these paths. The size of a single PC data structure is 21 bytes.

(2) Goal Memory

As defined in [9], goal memory in a rodent plays a role in route planning to goal locations, and is based on the position of the animal and its current needs/motivations. Our implementation of goal memory is a linked list of PC structures defined above. During the exploration phase, the path from a goal location found by the *ratbot* (e.g., food or water) to the home location is recorded in a linked list. The steps that occur to find the path from the goal to home during exploration is as follows:

- a) Random exploration until a goal location is found (water or food).
- b) Dedicate a place cell to the goal location (store coordinates).
- c) Return home directly (single vector home) or via a scan/backtrack algorithm if path is blocked by a barrier (adding/dedicate boundary vector cells, recording direction of barrier and coordinates, and TC for transition point around barrier, recording coordinates and ID).
- d) Save PC path in linked list (e.g., G1->TC1->G0). Save length of path as edge value for this path. Goal memory (e.g., nucleus accumbens).
- e) Go back to step (a), unless all goal locations have been found.

(3) Main Loop Time

The time duration of each iteration of the central processor's main loop shown in Fig. 5.3, assuming no goal is detected which needs to be approached, is 325ms. The majority of this processing time is spent in collecting and analyzing the ultrasonic data of the five sensors, as well as sending serial communication data to the motor controller board. There is a 20ms delay between activating each ultrasonic sensor. Additionally, each sensor is activated three consecutive times to determine the median distance. However, the time delay between these consecutive executions is minimal. Thus, $5 \text{ sensors} \times 20\text{ms/sensor} = 100\text{ms}$ in delay alone. Therefore, the frequency of the main perception/action loop = $1/325\text{ms} = 3\text{Hz}$.

5.3 FPGA Design

Cognitive mapping is accomplished using PC modules in the central processor and BC modules instantiated in the onboard FPGA. Since BC emulated brain cells can't be instantiated on the fly (same with a brain), the FPGA already has a certain number of BC modules. All BCs are initially set as inactive (active bit set to logic 0), until assigned to a location.

(1) Boundary Cells in the FPGA

BCs are activated and assigned to boundaries or barriers that lay between two goal locations. Typically, the FFs of BCs are assigned to any boundary of any kind in a rodent. However, due to the limited number of BCs, only boundaries found in the environment's interior location are being recorded. The BC module includes a directional field, which

records a discretized value of the angle of incident of the sensor. Fig. 5.7 illustrates the FPGA logic for the BCs, as well as the major data fields included in each BC module.

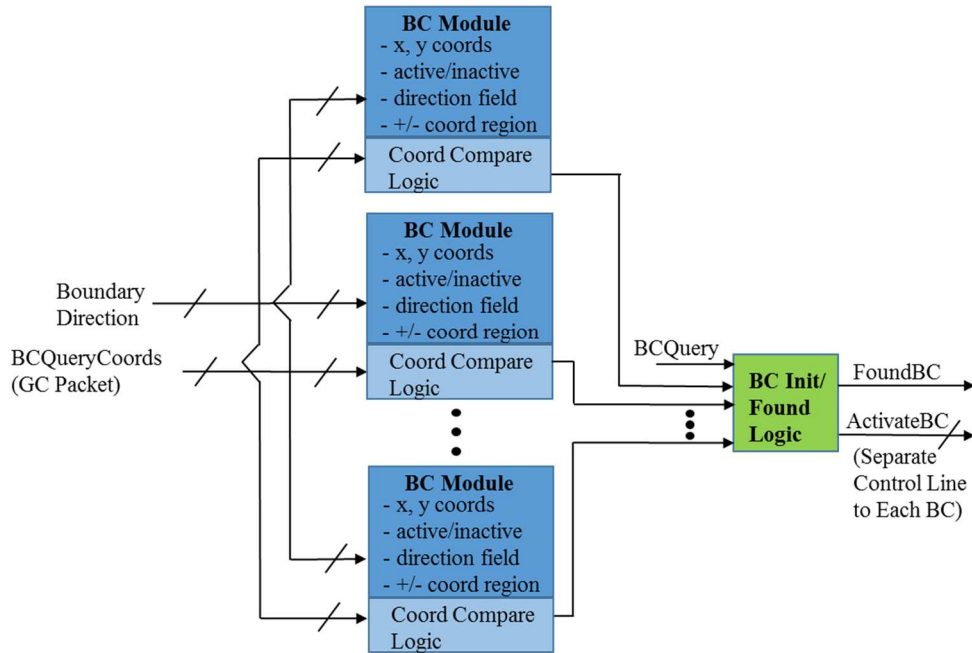


Figure 5.7. BC implementation in the FPGA.

When the central processor identifies a boundary, it sends the coordinates and angle to the FPGA. The FPGA checks this data against BC modules already activated in parallel. If the coordinates of the new BC FF are within range of an already activated module, then the module's range will either grow, or the data will be ignored (if close enough to the center range). Coordinates out of range of currently activated BC modules will cause the activation of a new module, and the data will be saved. BC FFs are given a rectangular area that is centered on the initial coordinates and perpendicular to the angle of incident. An example of BC modules instantiated is shown in the simulator's output in Fig. 5.5 as grey circles.

CHAPTER 6: LOCALIZATION AND PATH PLANNING

The ideal product of a neurophysiological based navigation system implemented in a mobile robot is to create a system that can navigate in varying types of environments. Rodents, as well as other animals, relate various allothetic and idiothetic cues with memory to derive a cognitive map of its environment, as well use this sensory input for spatial awareness. When we picture robots, we assume they are very precise in their actions/movements. This is true for various types of robots, such as those used in industrial applications (e.g., car manufacturing), or unmanned vehicles. However, for mobile robots used for disaster control and recovery, such as going into hazardous environments unfit for humans, these robots require more autonomy, and can get away with less precision in certain areas of their navigation capabilities. Autonomous mobile robots need to map their environment dynamically, while localizing themselves within the map without use of global positioning systems (GPS), as well as maneuver through tough or blocked terrain. The difficulty of the map making task is its mutual dependency on the robot's localization capabilities. Many traditional navigation systems use a statistical approach to localization and mapping, or SLAM specifically. For example, various forms of the Kalman filter (e.g., indirect, extended, and augmented) are used to reduce error between odometry and other sensor information [89, 96, 97].

Path planning requires the ability to search through the cognitive map efficiently and obtain movement details to accomplish the task of performing a successful trip. The

method and accuracy of the *ratbot*'s localization and path planning capabilities are covered next.

6.1 Localization

The *ratbot* initially localizes itself to its environment based on its starting location and heading at its home position, as is illustrated in Fig. 3.1. The initial anchoring of spatial orientation of the *ratbot*'s coordinate system is similar to the way a rodent anchors its grid network (GC FFs) with respect to external landmarks shortly after being introduced to a new environment [36, 37]. The use of both allothetic and idiothetic data is essential in adding spatial information to memorized visual information [98]. The occupancy grid is an example of a metric/grid based traditional (non-biological based) localization and mapping technique which shares similarities to our neurophysiological based system. Occupancy grids rely on highly accurate pose data with respect to a single global coordinate system. The pose system is typically a combination of both odometry and external/environment based sensors. The robot's area is divided up into equally sized squares. Each internally represented square, in the robot's memory, is given a probability of being occupied. This value is based on sensor readings, such as sonar sensors, which is typically represented by a two-dimensional Gaussian equation [8, 99-101]. Due to the amount of detail collected for the map of the occupancy grid, and the increasing number of squares for large areas, the memory requirement becomes unbounded, as well as the time to map the area. The *ratbot*, however, only creates place code for a select group of salient entities (i.e., goals, turns, and landmarks) and BC mapped areas for internal boundaries. Comparisons between

the occupancy grid localization and mapping method and our neurobiologically based system occur throughout the remainder of this section due to their many similarities.

Since idiothetic data is cumulative, so is the error. Thus, after time, the accrued PI error becomes too great for a robot to rely on its internal position estimate. Additionally, allothetic information can be misleading due to different locations having similar views or representations (perceptual aliasing) [39, 55, 61]. The *ratbot* uses a level of confidence function to keep the PI error bounded, as well as unique characteristics of the color-coded landscape markers to deal with these two issues.

6.1.1 Level of Confidence Calculation

The *ratbot* minimizes PI error by performing timely resets based on a calculated level of confidence (LoC). The LoC can be as simple as an allotted amount of time before a robot needs to return home to recalibrate [54], to being a function of how well known a place cell is recognized [85]. The *ratbot's* LoC is calculated using a combination of these two methods.

Due to the constant drift over time of the *ratbot's* gyroscope, as discussed in the sensors section, the LoC requires a time element to its calculation. Additionally, there will be assumed a level of systematic and non-systematic error occurring with each turn the *ratbot* makes. A threshold is set such that if the LoC decreases to a certain point, such that BC, PC, and VPC FFs will no longer be assigned to the cognitive map. Since the true accuracy of the PI system is nondeterministic in the presence of non-systematic errors, the *ratbot* needs to perform verification of already learned place fields it comes across during

any of its navigation modes, if the LoC is above the threshold. Additionally, the *ratbot* periodically returns home to recalibrate with its home base and initial heading.

6.1.2 Localization Accuracy

As was covered back in section 2, the robot's visual system can play a dominant role in its localization capabilities. For the *ratbot*, its current visual system is limited to recognizing pre-programmed color codes only. The visual system as is, allows for goal locations and landmarks to be found within the camera's downward facing FOV, while the *ratbot* is navigating. Additionally, the visual system allows for the *ratbot* to make action modification to reach goals, when their actual location falls within the camera's FOV. That is, the actual location of a remembered goal is now skewed from its previously determined coordinates by the accumulated PI errors up to that point. The right side of Fig. 2.9 illustrates this effect. Thus, the localization accuracy of the *ratbot*'s navigation system, with its present visual capabilities, is reduced to the accuracy of the system's PI values at any given time.

6.1.3 Place and Boundary Field Initialization Accuracy

The global or allocentric location of barriers detected via ultrasonic sensors, and goal locations detected via the *ratbot*'s camera, are calculated by translating their relative position to, and direction from, the *ratbot*'s inertial frame, as illustrated by the blue lines on the *ratbot* in Fig. 6.1a & b. Only the front and side ultrasonic sensors are used in the creation or verification of BCs. Fig. 6.1a illustrates how the *ratbot* reacts when the "whisker" sensor detects an object which becomes too close in range (pre-defined

threshold) with respect to the *ratbot's* path. The *ratbot* will stop and rotate until a measurement from the side sensor hits a minimum, indicate a near perpendicular position to the object, see Fig. 6.1b. The data from the side sensor is then stored in the FPGA for the BC module. Boundary detection occurs in the sonar data processing/affordances portion of the navigation software, as illustrated in Fig. 5.4.

Due to the divergent nature of sonar sensors, the accuracy of a detected barrier or object's actual location, assuming perfect PI, is rather poor. For long objects, such as walls, this is not an issue. Additionally, where walls or barriers end can be discerned and maneuvered around in real time. Only with smaller objects does accuracy become a real issue. Therefore, due to the wide beam width of the sonar sensor, BC FFs are given larger areas of representation than PC FFs.

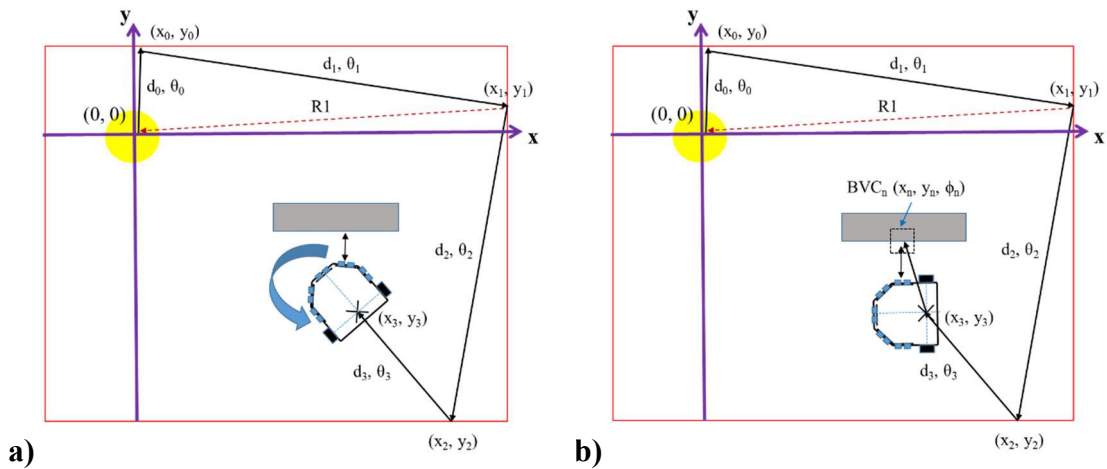


Figure 6.1. Recording of BC (BVC) location and angle of incident. a) The right “whisker” sensor detects a barrier (grey rectangle) to be too close (threshold range), so the *ratbot* stops and rotates. b) The distance measured from the side ultrasonic sensor is translated to a global (allocentric) reference frame from the *ratbot's* inertial frame.

The Pixy Cam can position the *ratbot* within a cm or two of the center of a goal or landmark. This is due to the precise calculations of the Pixy Cam's x, y coordinates of a

color-coded object in its FOV. Thus, any error in the recorded coordinates of a goal or landmark location is approximately a function of the PI error only. Thus, a PC's FF is smaller and more accurate the BC's FF. The size of the goal and landmark color-coded cards are 10cm x 14cm.

6.2 Route Planning

There are two methods of route planning. The first is for the *ratbot* to follow a saved path, such as those that are found during the *ratbot's* exploration phase. When the *ratbot* wishes to go from one goal location to another, it can do so quickly by “remembering” the path it has used in the past. This method is similar to a rodent's goal memory, as discussed in [9]. During the exploration phase, if a goal location, such as water or food, is found (detected by the Pixy camera), then the *ratbot* will return home to record the return path. However, if the *ratbot* is blocked during its trip home, due to a barrier, the *ratbot* will go into a scanning/backtrack mode to find a way around the barrier. The key turning points are saved as PCs and the full path is saved to memory in the microcontroller. This is illustrated in the Ratbot Simulator's output, Fig. 5.5. The small yellow squares represent turning points between goal location PCs and at the corner of the barriers. Therefore, the paths found this way may not be optimum, but are a solution for the *ratbot* to follow until it detects or learns a short cut. The total length of the path is saved with the path to represent its weight, which is used for determining shortcuts.

The second method of route planning is possible when the *ratbot* has found all the goal locations. At such point, there should be a fair number of activated BCs. The *ratbot* is then capable of using a look-ahead mode, which is similar to how humans can visualize

traversing a path while remaining stationary. This form of navigation, wayfinding, allows for changes to occur in the environment and gives the *ratbot* the capability to reroute on the fly. The BC FFs stored in the FPGA come into use in this route planning method. The *ratbot* performs the following sequence of events to find a new path:

- 1) Create line equations for the target path and for each BC that has multiple locations assigned to it.
- 2) Check to see if the target path intersects any of the recorded boundaries.
- 3) If no intersection (blockage) is detected, then proceed straight towards target. Go into exploration mode if an unrecorded barrier is found in the *ratbot*'s path.
- 4) If an intersection is found, add the TCs connect/associated with this BC and perform the A* algorithm on this graph.
- 5) If the A* algorithm fails to find a previously found goal, then exploration is required.

The A* path planning algorithm variant of the Dijkstra algorithm was chosen because it best aligns with the concept of the rodent or animal/insect following the Euclidean distance between two points. For the A* heuristic function, which performs a least-cost path algorithm on the nodes/PCs, is as follows:

$$f(x) = g(x) + h(x) \tag{6.1}$$

where, $g(x)$ is the sum of the distances between the initial position and the current node (PC) being examined, and $h(x)$ is the Euclidean distance (straight line calculated from stored coordinates) from the current node to the target node (PC).

CHAPTER 7: SUMMARY & FUTURE DIRECTIONS

7.1 Summary

As stated in the introduction, our goals include providing a low power solution to indoor mobile robot navigation systems, which requires less precise localization data than previous traditional or biomimetic models/systems to accomplish navigation. How well we met these challenges are analyzed next.

7.1.1 Power Analysis

One of the goals of this navigation system was to require relatively low power, such that the robot could easily carry the batteries required, as well as the sensors and processors. The *ratbot* currently carries two 12V, 2000mAh battery packs. However, only one of the batteries is currently used to power the entire system. The second battery is dedicated to an Arduino Yun microprocessor board, which is used during debugging only. The Yun microcontroller board becomes a WiFi access point, which a laptop can connect to and receive debug data from. The average running voltage and currents for processors onboard the *ratbot* are tabulated in Table 7.1. An estimated power of 2.5 watts is being used by the processors (in addition to some sensors).

Most of the power consumed by the system is by the two 12V motors, which can draw up to 530mA each. That is a potential 12W for the two motors alone. However, normal cruising speed for the *ratbot* is only about half of maximum and is used on a level/hard surface. Thus, the power used by the motors is probably around 6W, given the max power

of the battery and the fact the *ratbot* can usually last 2 hours or so, until the battery dips to approximately 6V.

Table 7.1. Power Consumption of Ratbot's Processors

Processor Type	Note	DC Volts (V)	DC Amperes (mA)	Power (mW)
Central Processor-Atmel ATmega2560	Plus 5 Ultrasonic Sensors & MEMS Gyro	5	128	640
Pixy Cam Dual Processor + Atmel ATmega328	Full Pixy Camera System	5	300	1500
Xilinx Spartan 6 XC6SLX9 FPGA		3.3	98.5	330
			Total Power:	2470

The low processing power required is attributed to low operating frequencies of the processors. For processor power consumption is proportional to the operating voltage (squared) times the clock frequency. The Atmel microcontrollers run at 16 MHz, the Pixy Cam's onboard NXP LPC4330 dual core processor operates at 204 MHz, and the Xilinx Spartan 6 XC6SLX9 FPGA is run with a 50 MHz clock.

The dimensions of the *ratbot* are approximately 21cm long by 17cm wide. The height of the chassis on the front of the *ratbot* is 25.5cm.

7.1.2 Navigation Environment Scalability

Given the limited memory of the microcontroller used as the main processor of the developed physiological based navigation system, scalability is an issue. The PC data structures require 21 bytes and are minimally used. The central processor (microcontroller)

used has a total of 256k bytes for both program memory and data. Currently, the program takes up 15,668 bytes out of the 253,952 bytes available (8k bytes of memory are dedicated to the boot loader). Therefore, there is plenty of memory available for PCs. However, to make the system reach across large areas, such as a large office building, the area should be segmented to several unique block areas. This helps with resetting/re-initializing the PI system across large spaces. Additionally, the BC data in the FPGA would have to be stored to memory, as should the PC data in the main controller. Thus, to adapt the current prototype system to one that can deal with larger areas, the following changes should be made:

- a) Add external memory and a processor core to the FPGA.
- b) Upgrade the central processor to be able to store and retrieve larger amounts of data.
- c) Change the visual system to be omnidirectional, thus needing to include visual processing algorithms and comparison/learning system (e.g., ANN based, comparison, etc.).

7.1.3 Episodic Memory

As mentioned at the beginning of this paper, the hippocampus is believed to play a major role in the storage of episodic memory, particularly during a biological creature's sleeping stage. From the point of view of our model, although sleep is not involved, the remembrance of paths taken and locations (PCs) found do model episodic memory [61].

7.1.4 Importance of Visual Recognition in Navigation

Although many animals, such as the rodent, can navigating previously learned paths while relying on only internal stimuli (no visual aid), such navigation breaks down with time as the PI error accumulates. Additionally, the initial learning of the animal's environment requires external stimuli. Thus, from working with the *ratbot* and researching many mobile robot systems, it is quite apparent that there is a strong correlation between the visual recognition capabilities and the overall navigation capabilities of the neurobiological based mobile robot. Navigation dominant on visual cues is referred to taxon navigation, and applies to animals, humans, insects, etc., as well as traditional and neurobiological based mobile robot navigation systems. This comes as no surprise as it has been shown that the specialized navigation and spatial awareness cells of a rodent are dependent to some degree on visual cues [9, 23, 102-104]. Additionally, biological systems, such as those found in rodents, can navigate on non-visuals cues as well. These can be auditory, olfactory, and/or somatosensory cues.

Of course, the caveat with using visual data, is the ability to process this data fast enough to be used in real-time. Additionally, information extraction requires deep learning neural networks, or similar, for image recognition. The neurophysiological based systems reviewed and cited in this paper use simple neural networks and compressed data techniques for simple environment recognition. However, for smarter navigation systems that can get more useful information, such as space or other detailed environmental information, and analyze it, then navigation becomes more informed. But the processing and power requirement is become too much to allow for the system to be onboard the

mobile robot. Some robots use cloud computing for this purpose. The next section analyzes artificial neural networks and possible processing technologies for onboard systems.

7.2 Possible Future Directions in Model Computation

Stanford University and Sandia National Laboratories have been working on creating a non-volatile organic electrochemical artificial synapse for neuromorphic computing [105]. This low-voltage, artificial synapse mimics the way neurons are connected in the brain. Thus, the neural inspired system could theoretically learn and keep its memory through the artificial synapse connectivity. Perhaps a neuromorphic computing machine, which more closely mimics the functionality of the brain than current processing systems, will be realized in the future. A system with elements that more closely resembles the dynamic learning structure of the human brain, and is similar with respect to processing capabilities, power requirements and size of the human brain.

7.2.1 Neural Networks

For completeness, a discussion on the computational demand required of the various neural networks used, such as the continuous attractor network of the RatSLAM [39, 68, 69, 106], the Hebbian learning rule and how it relates to the type of artificial neural networks (ANNs) used in the neurophysiological based navigation system literature surveyed for this paper [21], as well as deep learning, which wasn't used but has interesting possibilities given current computational technologies. Additionally, the computational limitations due to scalability of these types of navigation systems are covered.

(1) Continuous Attractor Network

To keep on track with closely modeling a neurophysiological system, both allothetic and idiothetic stimuli are fed into ANNs of the reviewed literature. The one difference is with the RatSLAM system, which uses a variant of an ANN system called the (3-D) continuous attractor network (CAN) system (see Fig. 2.9). Although the CAN is a type of ANN, it is less computationally demanding to update because the activity values of the CAN units are varied between 0 and 1, while keeping the weighted connections fixed. However, the statistical nature of the RatSLAM cell calculations, as covered shortly, will tax the processing system. Changes in the CAN cell's activity level ΔP is given in [39] by:

$$\Delta P = P * \varepsilon - \varphi, \quad (7.1)$$

or,

$$\Delta P_{x', y', \theta'} = \sum_i \sum_j \sum_k P_{i,j,k} \varepsilon_{a,b,c} - \varphi \quad (7.2)$$

where P represents the activity matrix of the network, ε is the connection matrix, $*$ is the convolution operator, and the constant φ is used to create global inhibition and general inhibition in the connection matrix. At the CAN cell level, as described in equation (7.2), $P_{x', y', \theta'}$ is the change in activity level for each cell, and $\varepsilon_{a,b,c}$ is the 3-D Gaussian distribution of weighted connections equation that creates local excitation and inhibition at the cell level, where a , b , and c are wrap around functions of x' , y' , and θ' respectively. Greater detail can be found in [82].

Another difference between the RatSLAM system compared to the rest of the systems presented in the literature review section, is that the other systems use ANNs throughout

their navigational system (thus increasing the computational complexity, but staying with the neurophysiological model theme), while RatSLAM only uses the CAN for mobile robot pose determination. The visual snapshot matching appears to be of a non-ANN based algorithm. Hence, the scaling down of neurophysiological realism due to on-board computational constraints.

(2) Hebbian Learning Rule

Hebbian based ANNs used in the research literature covered in this paper can be described by the general equation of:

$$y_i = \sum_j w_{ij}x_j \quad (7.3)$$

and
$$\Delta w_{ij} = \alpha x_j y_i \quad (7.4)$$

where, y_i is the output from neuron i , x_j is the j^{th} input, and w_{ij} is the weight from x_j to y_i . The scalar α is known as the learning rate and it may change with time. The Hebbian learning rule (Δw_{ij}) is named after D. Hebb [107] and his theory that the connection or synapse between two neurons strengthen as a result of a repeated pre- and postsynaptic neuron firing relationship. Incorporating a bias or threshold term w_0 , and some transfer function σ results in the Hebbian rule, as shown in [108-110], in the form of:

$$y_i = \sigma (\sum_j w_{ij}x_j - w_0) \quad (7.5)$$

The transfer function σ is typically a discrete step function:

$$\text{sgn}(t) = \begin{cases} 0 & \text{if } t < 0, \\ 1 & \text{if } t \geq 0, \end{cases} \quad (7.6)$$

or a smooth “sigmoid”, e.g.

$$\sigma(t) = (1 + e^{-t})^{-1}, \quad (7.7)$$

The sigmoid, as well as the tanh and rectified linear unit (ReLU) functions are typical non-linear neurons used. The ReLU is currently a very popular activation neuron in deep learning.

The Hebbian general equation is inherently unstable, where all the synapses can either reach their maximum allowed value or transition to zero [111-113]. Thus, a simple alternative equation to (4), such as that used in [46], [48] and [93], is as follows:

$$\Delta w_{ij} = \alpha x_j y_i (1 - w_{ij}) \quad (7.8)$$

The neural networks used in the literature surveyed typically use no more than a single hidden layer and are feedforward neural networks, see Fig. 7.1. These ANNs are adequate for simple, discrete input/output combinations, such as heading, turn angle, etc.

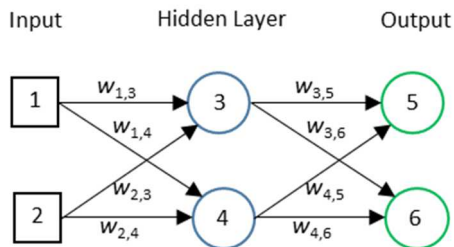


Figure 7.1. Single layer ANN with two inputs, two outputs and two neurons.

(3) Deep Learning

Deep learning is a growing variant of the previously described simple ANNs. This is due to its ability to find intricate structure in large data sets. Deep learning networks

accomplishes this through added multiple non-linear processing layers. These processing, or hidden, ANN layers extract various object feature layers. As previously stated, deep learning has offered advances in many domains, such as: image recognition, speech recognition, reconstructing brain circuits, natural language understanding, relational data, etc. Specifically, for navigation, it is the visual object recognition ability of deep learning and deep convolutional networks (e.g., traffic sign recognition, detection of pedestrians, etc.), which allow for autonomous mobile robots and self-driving cars [114] to be realized.

(4) Computational Time and Resources Limiting Scalability

When determining the computational demand of a neural network, there are three important parameters to consider: *size*, *depth* and *weight* of the network. The *size* is the number of neurons, the *depth* is the length of a longest path from an input point to an output neuron, while the sum of the absolute values of the weights represent the *weight* of the network.

The training of the ANNs that are used for complex pattern recognition, such as those found in interfacing allothetic stimuli to the navigation system, can only be accomplished off-line. The processing power and time required would have too large of an impact on mobile robot resources and usability. This is due to the many forward propagation and back propagation cycles required to set the weights of the ANN to the most optimum values possible (given set number of cycle constraints) for each training sample in the training phase. This is particularly true for deep neural networks, which have many hidden layers. Thus, the time complexity will be a function of network *size* and particularly *depth*. An

example of a simple two input, two output, single layer ANN is given in Fig. 7.1. Further examples can be found in the navigation models review section.

Ways in which to add neurobiological based entities, such as allothetic stimuli, other percepts and/or controlling influences (e.g., nucleus accumbens, grid cells, etc) from various parts of the brain, while maintaining a usable mobile robot footprint, are as follows:

- 1) Use of mobile GPGPU of more complex ANNs,
- 2) Removing ANNs from simpler parts of the system that can be easily replaced by a good, cheap sensor (e.g., head direction ANN in [46] with a MEMs gyroscope).
- 3) Creating an application specific integrated circuit (ASIC) that models ANNs.
- 4) Use of FPGAs

Option 3 would be the most expensive, but also the most efficient in power, size and processing capabilities. Option 1 is a more flexible option, but still requires a significantly more power than an ASIC and special programming expertise. An example of what is available is the NVIDIA[®] Jetson[™] TX1 Module GPU with 256 light weight parallel processor (CUDA[®]) cores. They can be programmed using CUDA or cuDNN. Option 2 takes the system away from the realism of a neurobiological system, but some tradeoffs need to be made to model portions that are most important to the research. Using FPGAs are also a possibility, especially if a processor core is included.

7.3 In Brief

It is the hope of many researchers that work being performed in neurobiological based navigation and spatial awareness systems will offer added technological advances to the autonomous navigation capabilities of mobile robots, as well as to better understanding at least a small portion of the brain.

REFERENCES

- [1] Dictionary.com "cognitive," in Dictionary.com Unabridged. Source location: Random House, Inc. <http://www.dictionary.com/browse/cognitive>. Available: <http://www.dictionary.com/>. Accessed: May 18, 2017.
- [2] M. O. Franz and H. A. Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*, vol. 30, pp. 133-153, 2000.
- [3] O. Trullier, S. I. Wiener, A. Berthoz, and J.-A. Meyer, "Biologically based artificial navigation systems: Review and prospects," *Progress in neurobiology*, vol. 51, pp. 483-544, 1997.
- [4] I. J. O. Wallgrün, "Chapter 2: Robot Mapping," in *Hierarchical Voronoi Graphs*, ed: Springer, 2010, pp. 11-43.
- [5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 108-117, 2006.
- [6] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *Robotics & Automation Magazine, IEEE*, vol. 13, pp. 99-110, 2006.
- [7] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms," in *Research and Development, 2006. Scored 2006. 4th Student Conference on*, 2006, pp. 183-188.

- [8] D. Gonzalez-Arjona, A. Sanchez, F. López-Colino, A. de Castro, and J. Garrido, "Simplified Occupancy Grid Indoor Mapping Optimized for Low-Cost Robots," *ISPRS International Journal of Geo-Information*, vol. 2, p. 959, 2013.
- [9] A. D. Redish, *Beyond the cognitive map: from place cells to episodic memory*: MIT Press Cambridge, MA, 1999.
- [10] E. C. Tolman, "Cognitive maps in rats and men," *Psychological review*, vol. 55, p. 189, 1948.
- [11] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat," *Brain research*, vol. 34, pp. 171-175, 1971.
- [12] N. Burgess, E. A. Maguire, and J. O'Keefe, "The human hippocampus and spatial and episodic memory," *Neuron*, vol. 35, pp. 625-641, 2002.
- [13] M. Fyhn, S. Molden, M. P. Witter, E. I. Moser, and M.-B. Moser, "Spatial representation in the entorhinal cortex," *Science*, vol. 305, pp. 1258-1264, 2004.
- [14] M. Müller and R. Wehner, "Path integration in desert ants, *Cataglyphis fortis*," *Proceedings of the National Academy of Sciences*, vol. 85, pp. 5287-5290, 1988.
- [15] C. Darwin, "Origin of certain instincts," *Nature*, vol. 7, pp. 417-418, 1873.
- [16] M. Collett, L. Chittka, and T. S. Collett, "Spatial memory in insect navigation," *Current Biology*, vol. 23, pp. R789-R800, 2013.
- [17] M. Collett and T. S. Collett, "How do insects use path integration for their navigation?," *Biological cybernetics*, vol. 83, pp. 245-259, 2000.
- [18] P. Graham and M. Mangan, "Insect navigation: do ants live in the now?," *Journal of Experimental Biology*, vol. 218, pp. 819-823, 2015.

- [19] A. S. Etienne, R. Maurer, and V. Séguinot, "Path integration in mammals and its interaction with visual landmarks," *Journal of Experimental Biology*, vol. 199, pp. 201-209, 1996.
- [20] A. S. Etienne and K. J. Jeffery, "Path integration in mammals," *Hippocampus*, vol. 14, pp. 180-192, 2004.
- [21] P. J. Zeno, S. Patel, and T. M. Sobh, "Review of Neurobiologically Based Mobile Robot Navigation System Research Performed Since 2000," *Journal of Robotics*, vol. 2016, p. 17, 2016.
- [22] A. D. Redish, "Beyond the cognitive map: Contributions to a computational neuroscience theory of rodent navigation," Carnegie Mellon University, 1997.
- [23] D. Bush, C. Barry, and N. Burgess, "What do grid cells contribute to place cell firing?," *Trends in neurosciences*, vol. 37, pp. 136-145, 2014.
- [24] N. Burgess, M. Recce, and J. O'Keefe, "A model of hippocampal function," *Neural networks*, vol. 7, pp. 1065-1081, 1994.
- [25] E. I. Moser, Y. Roudi, M. P. Witter, C. Kentros, T. Bonhoeffer, and M.-B. Moser, "Grid cells and cortical representation," *Nature Reviews Neuroscience*, vol. 15, pp. 466-481, 2014.
- [26] P. J. Zeno, "Emulating the Functionality of Rodents' Neurobiological Navigation and Spatial Cognition Cells in a Mobile Robot," *International Journal of Computing*, vol. 14, pp. 77-85, 2015.
- [27] K. C. Little, "A Rat Model of Sytemic Chemotherapy for Breast Cancer to Evaluate and Treat Chemobrain," DTIC Document2007.

- [28] J. O'Keefe and D. Conway, "Hippocampal place units in the freely moving rat: why they fire where they fire," *Experimental Brain Research*, vol. 31, pp. 573-590, 1978.
- [29] K. B. Kjelstrup, T. Solstad, V. H. Brun, T. Hafting, S. Leutgeb, M. P. Witter, *et al.*, "Finite scale of spatial representation in the hippocampus," *Science*, vol. 321, pp. 140-143, 2008.
- [30] J. S. Taube, "The head direction signal: origins and sensory-motor integration," *Annu. Rev. Neurosci.*, vol. 30, pp. 181-207, 2007.
- [31] C. Lever, S. Burton, A. Jeewajee, J. O'Keefe, and N. Burgess, "Boundary vector cells in the subiculum of the hippocampal formation," *The journal of neuroscience*, vol. 29, pp. 9771-9777, 2009.
- [32] D. Derdikman, "Are the boundary-related cells in the subiculum boundary-vector cells?," *The Journal of Neuroscience*, vol. 29, pp. 13429-13431, 2009.
- [33] M. Fyhn, T. Hafting, A. Treves, M.-B. Moser, and E. I. Moser, "Hippocampal remapping and grid realignment in entorhinal cortex," *Nature*, vol. 446, pp. 190-194, 2007.
- [34] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 436, pp. 801-806, 2005.
- [35] C. Barry and N. Burgess, "Neural mechanisms of self-location," *Current Biology*, vol. 24, pp. R330-R339, 2014.
- [36] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the 'cognitive map'," *Nature Reviews Neuroscience*, vol. 7, pp. 663-678, 2006.

- [37] E. I. Moser and M. B. Moser, "A metric for space," *Hippocampus*, vol. 18, pp. 1142-1156, 2008.
- [38] F. Sargolini, M. Fyhn, T. Hafting, B. L. McNaughton, M. P. Witter, M.-B. Moser, *et al.*, "Conjunctive representation of position, direction, and velocity in entorhinal cortex," *Science*, vol. 312, pp. 758-762, 2006.
- [39] G. Wyeth and M. Milford, "Spatial cognition for robots," *Robotics & Automation Magazine, IEEE*, vol. 16, pp. 24-32, 2009.
- [40] E. I. Moser, E. Kropff, and M.-B. Moser, "Place cells, grid cells, and the brain's spatial representation system," *Neuroscience*, vol. 31, p. 69, 2008.
- [41] D. Goldschmidt, S. Dasgupta, F. Wörgötter, and P. Manoonpong, "A neural path integration mechanism for adaptive vector navigation in autonomous agents," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, 2015, pp. 1-8.
- [42] H. Wolf, "Odometry and insect navigation," *Journal of Experimental Biology*, vol. 214, pp. 1629-1641, 2011.
- [43] D. Heusser and R. Wehner, "The visual centring response in desert ants, *Cataglyphis fortis*," *Journal of experimental biology*, vol. 205, pp. 585-590, 2002.
- [44] B. Ronacher and R. Wehner, "Desert ants *Cataglyphis fortis* use self-induced optic flow to measure distances travelled," *Journal of Comparative Physiology A*, vol. 177, pp. 21-27, 1995.
- [45] Y. Shrager, C. B. Kirwan, and L. R. Squire, "Neural basis of the cognitive map: path integration does not require hippocampus or entorhinal cortex," *Proceedings of the National Academy of Sciences*, vol. 105, pp. 12034-12038, 2008.

- [46] A. Arleo and W. Gerstner, "Modeling rodent head-direction cells and place cells for spatial learning in bio-mimetic robotics," *From Animals to Animats*, vol. 6, pp. 236-245, 2000.
- [47] A. Barrera and A. Weitzenfeld, "Biologically-inspired robot spatial cognition based on rat neurophysiological studies," *Autonomous Robots*, vol. 25, pp. 147-169, 2008.
- [48] T. Strösslin, D. Sheynikhovich, R. Chavarriaga, and W. Gerstner, "Robust self-localisation and navigation based on hippocampal place cells," *Neural networks*, vol. 18, pp. 1125-1140, 2005.
- [49] J. O'keefe and L. Nadel, *The hippocampus as a cognitive map* vol. 3: Clarendon Press Oxford, 1978.
- [50] J. S. Taube, R. U. Muller, and J. B. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis," *The Journal of Neuroscience*, vol. 10, pp. 420-435, 1990.
- [51] M. A. Brown and P. E. Sharp, "Simulation of spatial learning in the Morris water maze by a neural network model of the hippocampal formation and nucleus accumbens," *Hippocampus*, vol. 5, pp. 171-188, 1995.
- [52] A. D. Redish and D. S. Touretzky, "Cognitive maps beyond the hippocampus," *Hippocampus*, vol. 7, pp. 15-35, 1997.
- [53] S. Zrehen and P. Gaussier, "Building grounded symbols for localization using motivations," in *Proceedings of the Fourth European Conference on Artificial Life, ECAL97, Brighton, UK*, 1997, pp. 299-308.

- [54] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," *Biological Cybernetics*, vol. 83, pp. 287-299, 2000.
- [55] A. Arleo, F. Smeraldi, and W. Gerstner, "Cognitive navigation based on nonuniform Gabor space sampling, unsupervised growing networks, and reinforcement learning," *Neural Networks, IEEE Transactions on*, vol. 15, pp. 639-652, 2004.
- [56] J. G. Fleischer, J. A. Gally, G. M. Edelman, and J. L. Krichmar, "Retrospective and prospective responses arising in a modeled hippocampus during maze navigation by a brain-based device," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 3556-3561, 2007.
- [57] J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 2111-2116, 2005.
- [58] J. L. Krichmar, A. K. Seth, D. A. Nitz, J. G. Fleischer, and G. M. Edelman, "Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions," *Neuroinformatics*, vol. 3, pp. 197-221, 2005.
- [59] R. Morris, "Developments of a water-maze procedure for studying spatial learning in the rat," *Journal of neuroscience methods*, vol. 11, pp. 47-60, 1984.
- [60] H. De Garis, C. Shuo, B. Goertzel, and L. Ruiting, "A world survey of artificial brain projects, Part I: Large-scale brain simulations," *Neurocomputing*, vol. 74, pp. 3-29, 2010.

- [61] V. V. Hafner, "Robots as Tools for Modelling Navigation Skills—A Neural Cognitive Map Approach," in *Robotics and cognitive approaches to spatial mapping*, ed: Springer, 2008, pp. 315-324.
- [62] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, pp. 59-69, 1982.
- [63] A. Barrera and A. Weitzenfeld, "Rat-inspired model of robot target learning and place recognition," in *Control & Automation, 2007. MED'07. Mediterranean Conference on*, 2007, pp. 1-6.
- [64] A. Guazzelli, M. Bota, F. J. Corbacho, and M. A. Arbib, "Affordances, motivations, and the world graph theory," *Adaptive Behavior*, vol. 6, pp. 435-471, 1998.
- [65] A. B. Ramírez and A. W. Ridel, "Bio-inspired Model of Robot Adaptive Learning and Mapping," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 4750-4755.
- [66] A. B. Ramírez and A. W. Ridel, "Biologically Inspired Neural Controller for Robot Learning and Mapping," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, 2006, pp. 3664-3671.
- [67] A. Barrera and A. Weitzenfeld, "Computational modeling of spatial cognition in rats and robotic experimentation: Goal-oriented navigation and place recognition in multiple directions," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, 2008, pp. 789-794.

- [68] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *The International Journal of Robotics Research*, vol. 29, pp. 1131-1153, 2010.
- [69] M. J. Milford, G. F. Wyeth, and D. Prasser, "RatSLAM: a hippocampal model for simultaneous localization and mapping," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2004, pp. 403-408 Vol.1.
- [70] C. Boucheny, N. Brunel, and A. Arleo, "A continuous attractor network model without recurrent excitation: maintenance and integration in the head direction cell system," *Journal of computational neuroscience*, vol. 18, pp. 205-227, 2005.
- [71] N. Sunderhauf and P. Protzel, "Beyond RatSLAM: Improvements to a biologically inspired SLAM system," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, 2010, pp. 1-8.
- [72] N. Cuperlier, M. Quoy, and P. Gaussier, "Neurobiologically inspired mobile robot navigation and planning," *Frontiers in neurorobotics*, vol. 1, 2007.
- [73] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms second edition," *The Knuth-Morris-Pratt Algorithm*, year, 2001.
- [74] U. M. Erdem and M. E. Hasselmo, "A biologically inspired hierarchical goal directed navigation model," *Journal of Physiology-Paris*, vol. 108, pp. 28-37, 2014.
- [75] L. M. Giocomo, M.-B. Moser, and E. I. Moser, "Computational models of grid cells," *Neuron*, vol. 71, pp. 589-603, 2011.
- [76] N. Burgess, C. Barry, and J. O'Keefe, "An oscillatory interference model of grid cell firing," *Hippocampus*, vol. 17, pp. 801-812, 2007.

- [77] L. M. Giocomo and M. E. Hasselmo, "Computation by oscillations: implications of experimental data for theoretical models of grid cells," *Hippocampus*, vol. 18, pp. 1186-1199, 2008.
- [78] U. M. Erdem and M. Hasselmo, "A goal-directed spatial navigation model using forward trajectory planning based on grid cells," *European Journal of Neuroscience*, vol. 35, pp. 916-931, 2012.
- [79] O. Shipston-Sharman, L. Solanka, and M. F. Nolan, "Continuous attractor network models of grid cell firing based on excitatory-inhibitory interactions," *The Journal of physiology*, 2016.
- [80] Y. Burak and I. R. Fiete, "Accurate path integration in continuous attractor network models of grid cells," *PLoS computational biology*, vol. 5, p. e1000291, 2009.
- [81] N. Burgess, "Grid cells and theta as oscillatory interference: theory and predictions," *Hippocampus*, vol. 18, pp. 1157-1174, 2008.
- [82] M. J. Milford, J. Wiles, and G. F. Wyeth, "Solving navigational uncertainty using grid cells on robots," 2010.
- [83] A. T. Keinath, "The Preferred Directions of Conjunctive Grid X Head Direction Cells in the Medial Entorhinal Cortex Are Periodically Organized," *PloS one*, vol. 11, p. e0152041, 2016.
- [84] P. Gaussier, J. Banquet, F. Sargolini, C. Giovannangeli, E. Save, and B. Poucet, "A model of grid cells involving extra hippocampal path integration, and the hippocampal loop," *Journal of integrative neuroscience*, vol. 6, pp. 447-476, 2007.

- [85] A. Jauffret, N. Cuperlier, and P. Gaussier, "From grid cells and visual place cells to multimodal place cell: a new robotic architecture," *Frontiers in neurorobotics*, vol. 9, 2015.
- [86] "Grid Cell," *Wikipedia*, p. N.P, 2016.
- [87] J. Ventura and T. Höllerer, "Wide-area scene mapping for mobile visual tracking," in *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, 2012, pp. 3-12.
- [88] T. Strösslín, R. Chavarriaga, D. Sheynikhovich, and W. Gerstner, "Modelling path integrator recalibration using hippocampal place cells," in *International Conference on Artificial Neural Networks*, 2005, pp. 51-56.
- [89] F. Azizi and N. Houshangi, *Mobile robot position determination*: INTECH Open Access Publisher, 2011.
- [90] S. Thrun, "Robotic mapping: A survey," *Exploring artificial intelligence in the new millennium*, vol. 1, pp. 1-35, 2002.
- [91] K. Hardcastle, S. Ganguli, and L. M. Giocomo, "Environmental boundaries as an error correction mechanism for grid cells," *Neuron*, vol. 86, pp. 827-839, 2015.
- [92] R. Hayman and N. Burgess, "How Cumulative Error in Grid Cell Firing Is Literally Bounded by the Environment," *Neuron*, vol. 86, pp. 607-609, 2015.
- [93] V. V. Hafner, "Cognitive maps in rats and robots," *Adaptive Behavior*, vol. 13, pp. 87-96, 2005.
- [94] D. Massa, "Choosing an Ultrasonic Sensor for Proximity or Distance Measurement Part 2: Optimizing Sensor Selection," *Sensorsmag.com*, 1999.

- [95] S. Stewart, A. Jeewajee, T. J. Wills, N. Burgess, and C. Lever, "Boundary coding in the rat subiculum," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, p. 20120514, 2014.
- [96] A. Araneda, S. E. Fienberg, and A. Soto, "A statistical approach to simultaneous mapping and localization for mobile robots," *The Annals of Applied Statistics*, pp. 66-84, 2007.
- [97] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," *arXiv preprint arXiv:1304.1098*, 2013.
- [98] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots:: I. a review of localization strategies," *Cognitive Systems Research*, vol. 4, pp. 243-282, 2003.
- [99] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 2432-2437.
- [100] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, pp. 111-127, 2003.
- [101] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy Grid Models for Robot Mapping in Changing Environments," in *AAAI*, 2012.
- [102] S. S. Winter and J. S. Taube, "Head direction cells: from generation to integration," in *Space, Time and Memory in the Hippocampal Formation*, ed: Springer, 2014, pp. 83-106.
- [103] J. J. Knierim and D. A. Hamilton, "Framing Spatial Cognition: Neural Representations of Proximal and Distal Frames of Reference and Their Roles in

- Navigation," *Physiological Reviews*, vol. 91, pp. 1245-1279, 2011-10-01 00:00:00 2011.
- [104] N. Burgess, J. G. Donnett, K. J. Jeffery, and O. John, "Robotic and neuronal simulation of the hippocampus and rat navigation," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 352, pp. 1535-1543, 1997.
- [105] Y. van de Burgt, E. Lubberman, E. J. Fuller, S. T. Keene, G. C. Faria, S. Agarwal, *et al.*, "A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing," *Nature Materials*, 2017.
- [106] M. Milford and G. Wyeth, "Hippocampal models for simultaneous localisation and mapping on an autonomous robot," in *Proceedings of the Australasian Conference on Robotics and Automation, 2003*, 2003.
- [107] D. O. Hebb, *The organization of behavior: A neuropsychological approach*: John Wiley & Sons, 1949.
- [108] P. Orponen, "Computational complexity of neural networks: a survey," *Nordic Journal of Computing*, vol. 1, pp. 94-110, 1994.
- [109] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, vol. 25, p. 27, 1995.
- [110] C. M. Bishop, "Pattern Recognition," *Machine Learning*, 2006.
- [111] K. D. Miller and D. J. MacKay, "The role of constraints in Hebbian learning," *Neural Computation*, vol. 6, pp. 100-126, 1994.
- [112] W. Krauth and M. Mézard, "Learning algorithms with optimal stability in neural networks," *Journal of Physics A: Mathematical and General*, vol. 20, p. L745, 1987.

- [113] W. Gerstner and W. M. Kistler, "Mathematical formulations of Hebbian learning," *Biological cybernetics*, vol. 87, pp. 404-415, 2002.
- [114] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. LeCun, "Deep belief net learning in a long-range vision system for autonomous off-road driving," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 628-633.