Using an FPGA to Emulate Grid Cell Spatial Cognition in a Mobile Robot

Peter Zeno

Advisors: Prof. Sarosh Patel, Prof. Tarek Sobh

Interdisciplinary RISC Lab, School of Engineering

University of Bridgeport, Bridgeport, CT

Introduction

The ability for a mobile robot to perform simultaneous localization and mapping (SLAM) methodology, provides a means for it to be truly autonomous. SLAM is the capability of an entity to simultaneously create a map of its unknown environment and localize itself in that map. A ***classical*** SLAM navigation system relies mainly on *algorithms* for data association, visual data processing (e.g., recognition and comparison), path integration (PI) error rectification, etc. Whereas, a ***neurobiological*** based system relies heavily on intricately *integrated neural networks* to accomplish these same tasks.

Abstract

In theory, an autonomous mobile robot's ability to navigate with greater intelligence and agility in a dynamic environment would be possible if its navigation system was modeled after that of biological creatures. More specifically, a system that mimics the functionality of neurobiological navigation and spatial awareness cells, as found in the ***hippocampus*** and ***entorhinal*** cortex of a rodent's brain (Fig. 1). These navigation cells include: ***place cells, head direction cells, boundary cells, and grid cells***, as well as ***memory***. Our mobile robot, known as ***ratbot****,* uses inspiration from place cells for key locations, boundary cells for detecting significant errors in **path integration**, and grid cells for firing LEDs to aid in visual system debug. These three cells (logic interpretation), are implemented in a field programmable gate array (**FPGA**) silicon chip, to allow for parallel search.

Path Integration

Path integration (PI) was first suggested by Darwin [2], and confirming this hypothesis was shown in [3]. Fig. 2 illustrates the concept of PI used by animals, as well as the *ratbot*. In this figure, the rat leaves his or her home, travels around the enclosed area until it finds food, then returns home. The foraging/navigation task is accomplished by the rat continuously updating a return vector home approximation from the change in its *head direction* via **vestibular stimuli**, and *distance traveled* via **proprioceptive stimuli.** Similarly, the *ratbot's* "brain", an Arduino microcontroller board, uses distance traveled information gathered from the *ratbot's* motor **encoders** and the measured change in direction from a microelectromechanical systems (MEMS) based **gyroscope**, for calculating the return distance and direction to home, respectively, see Fig. 5 and Fig. 6a for physical implementation. The *ratbot's* vision (**ultrasonic sensor**) is for object avoidance only. This is similar to a rat foraging in the dark.

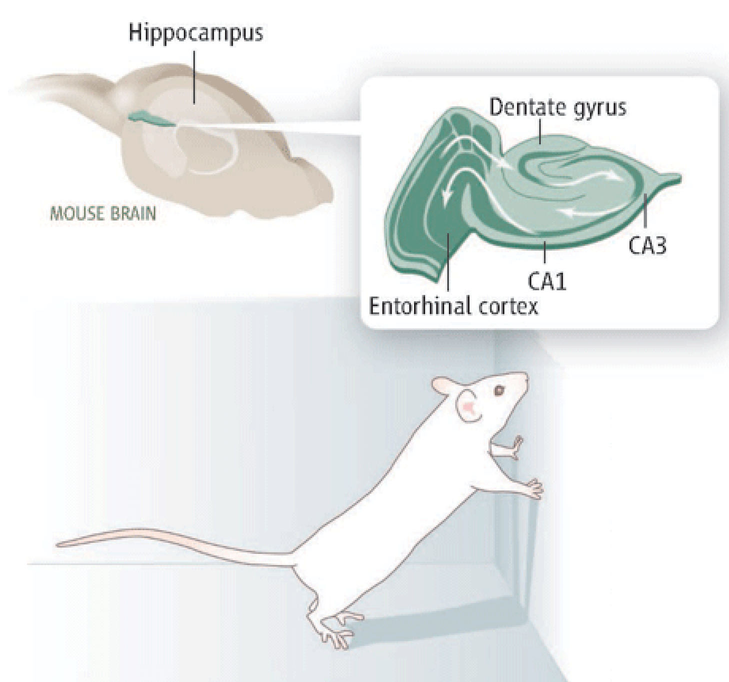

Figure 1: Rodent's hippocampus and entorhinal cortex [1].

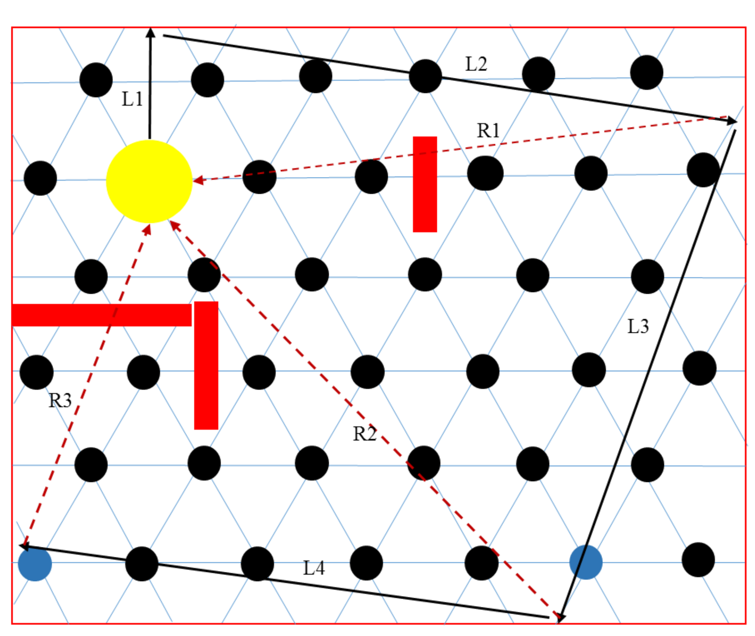

Figure 2: An example environment and travel scenario of the ratbot. For **PI**: The yellow circle is the ratbot's home (start and ending position). The black arrows (Ln) represents a travel leg of the journey. The red dashed arrows (Rn) are the return vectors calculated along the way (PI).

Analysis: PI Error

Due to sensor measurement errors, robot drift, and/or other possible external influences, the robot's true end location and heading are typically at odds with its true pose. This PI error accumulates with time as the robot continues to roam (Fig. 3 & 4). The *ratbot* experiences a slow enough accumulating PI error such that adding a simple allothetic sensor (e.g., pattern recognition camera) and dispersed salient distal cues, will bound this error.

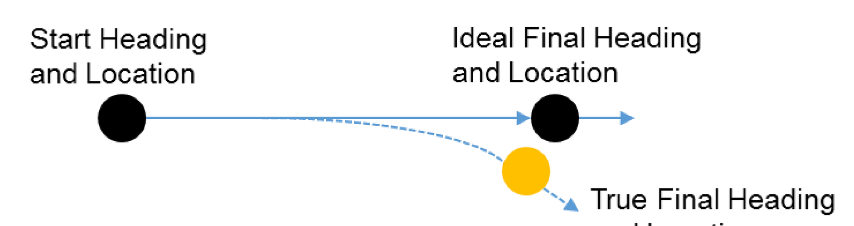

Figure 3: Accumulating PI error in a mobile robot and as seen with the *ratbot*.

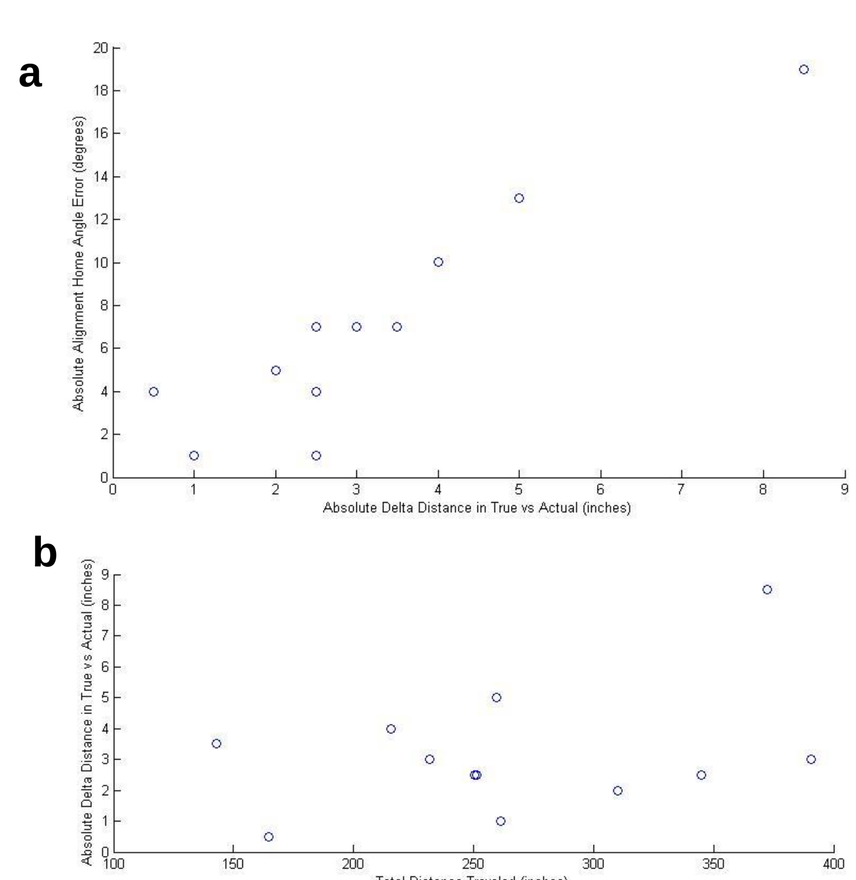

Figure 4: PI error data gathered from the *ratbot*. **a)** Correlation shown between final angle home and size of error in travel distance home. **b)** No apparent correlation between total distance covered by *ratbot* and magnitude of error in travel distance home.

Cognition Map and Route Planning

Through the use of an **FPGA** (Fig. 6b), the *ratbot's* environment is ***logically*** mapped into a two dimensional array of parallel processing units of place cells and boundary vector cells. Three grid cells of different phases are also used for visual PI error detection via LEDs. Routes are save as linked lists (e.g., G1->P1->G0).

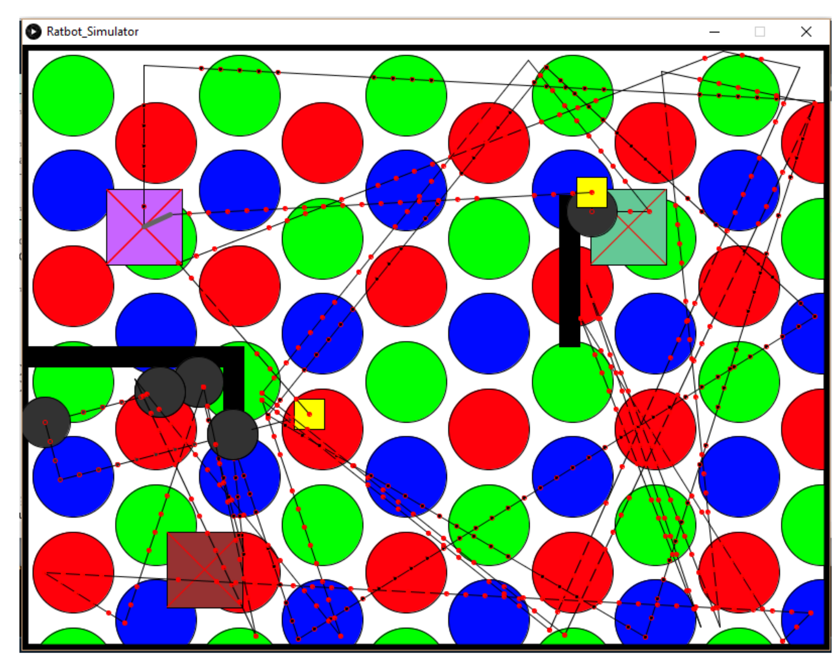

Figure 7: Ratbot Simulator output. Place cells are used for goal locations (large squares) and as a turn cell (yellow squares). Boundary vector cells are used for PI error detection via use of the cognitive map.

Sensor	Map	Possible Explanation(s)	Associated PI Error Color
Barrier Detected	No Boundary Cell	Not yet found. New barrier. PI Error.	(yellow)
Barrier Detected	Boundary Cell	Correct or slightly off in PI and nearby barrier section was previously recorded.	(green) (yellow)
No Barrier Detected	No Boundary Cell	Correct.	(green)
No Barrier Detected	Boundary Cell	Error in PI.	(red)

Figure 8: Map/PI error detection chart.

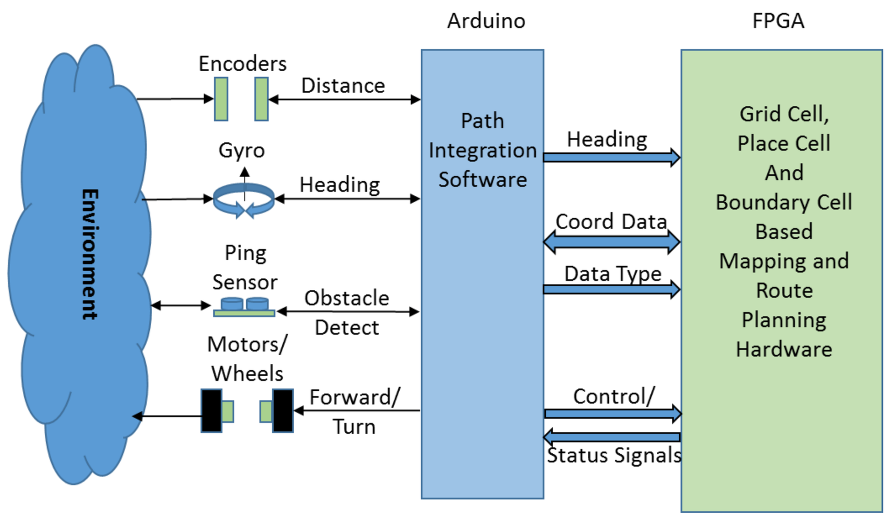

Figure 5: A top level block diagram of the path integration, mapping and route planning circuitry as implemented in the *ratbot* to simulate the specialized neurobiological navigation cells found in a rat's brain.

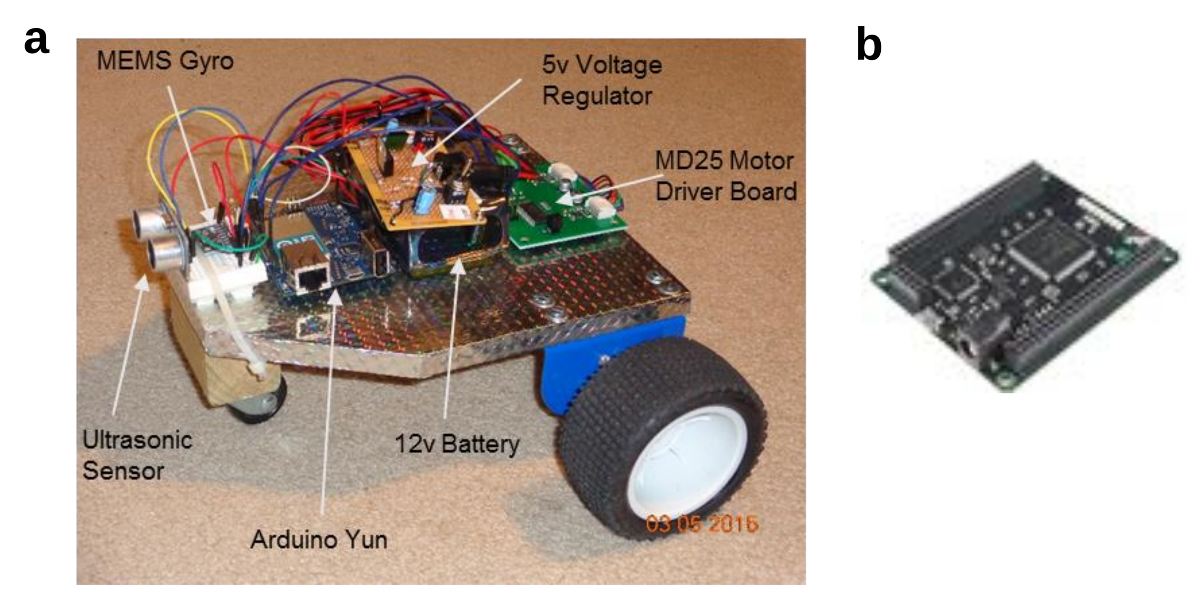

Figure 6: **a)** The *ratbot* and **b)** the Mojo Xilinx FPGA board.

References

[1] http://mindblog.dericbownds.net

[2] C. Darwin, "Origin of Certain Instincts," *Nature,* vol. 7, pp. 417-418, 1873.

[3] H. Mittelstaedt and M.-L. Mittelstaedt, "Homing by path integration," in *Avian navigation*, ed: Springer, 1982, pp. 290-297.