MODELS OF COCKROACH SHELTER SEEKING

IMPLEMENTED ON A ROBOTIC TEST PLATFORM

By

BRIAN ROBERT TIETZ

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Thesis Adviser: Dr. Roger D. Quinn

Department of Mechanical Engineering

CASE WESTERN RESERVE UNIVERSITY

January, 2012
CASE WESTERN RESERVE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

Brian Robert Tietz

candidate for the Master of Science degree *

(signed) Roger D Quinn
(chair of the committee)

Roy E Ritzmann

Michael S Branicky

(date) 11-7-11

*We also certify that written approval has been obtained for any proprietary material contained therein.
Table of Contents

List of Tables ........................................................................................................................................ 4
List of Figures ......................................................................................................................................... 5
Acknowledgements .......................................................................................................................... 8
Abstract ................................................................................................................................................ 10

Chapter 1 – Introduction and Goals ............................................................................................... 11
  1.1 Motivation for this project ........................................................................................................ 11
  1.2 Goals of this Thesis .................................................................................................................. 12
  1.3 Chapter Summaries .................................................................................................................. 12

Chapter 2 - Background .................................................................................................................. 14
  2.1 Robotic Approaches for Unknown Environments ................................................................ 14
  2.2 Inspiration from Cockroaches ................................................................................................. 16
  2.3 The Cockroach Shade Response ............................................................................................ 16
  2.4 Wall Following ........................................................................................................................ 17
  2.5 Robots as a biological test platform ....................................................................................... 18

Chapter 3 – Methods 1: Generating the Animal Behavior ........................................................... 20
  3.1 Animals .................................................................................................................................... 20
  3.2 The Arena ................................................................................................................................ 21
    ...................................................................................................................................................... 21
    3.2.1 Goal Selection .................................................................................................................... 21
    3.2.2 The Light Arena ............................................................................................................... 21
  3.3 Trial Procedure .......................................................................................................................... 22
    3.3.1 Postprocessing ................................................................................................................... 23
  3.4 Cockroach Behavior .................................................................................................................. 23
    3.4.1 Differences in Behavior along Wall .................................................................................. 24
    3.4.2 Wall Departures and Turnarounds .................................................................................... 25
    3.4.3 Arena Pivots ...................................................................................................................... 27
    3.4.4 Exploration Efficiency ....................................................................................................... 29
    3.4.5 Time Spent under Shelter ............................................................................................... 30
  3.5 Modeling Behavior in Simulation .............................................................................................. 32
List of Tables

Table 1: Summary of Cockroach versus Robot Data......................................................... 53
List of Figures

Figure 1: The cockroach *Blaberus discoidalis* ................................................................. 20
Figure 2: The arena, with dimensions .............................................................................. 21
Figure 3: The arena, in color ............................................................................................. 21
Figure 4: The overhead camera ......................................................................................... 23
Figure 5: Cockroaches’ speed decreases at close proximity to a wall ......................... 24
Figure 6: Cockroaches are more likely to depart the wall if they are facing away from the
shelter .................................................................................................................................. 25
Figure 7: The angle of the cockroaches' body to the wall it is departing ....................... 26
Figure 8: Cockroaches have some chance of turning around while on the wall ............ 27
Figure 9: There is a small probability the robot will continue straight to the shelter after a
pivot ...................................................................................................................................... 28
Figure 10: Pivot direction probabilities ............................................................................ 29
Figure 11: The exploration efficiency metrics of the cockroach with the shelter in the SC,
or mid-back location ........................................................................................................ 30
Figure 12: Heat maps of cockroach coverage .................................................................. 31
Figure 13: An example of the pivots and straights used to generate the RAMBLER model
data .................................................................................................................................... 32
Figure 14: The RAMBLER Algorithm ............................................................................ 33
Figure 15: The Various Barrier Lengths in the arena ...................................................... 34
Figure 16: The robot that is used as a test platform ......................................................... 36
Figure 17: Close up of the Robot’s acrylic antenna from above ....................................... 37
Figure 18: The robot in its arena ....................................................................................... 38
Figure 19: The forced fit calibration................................................................. 41
Figure 20: A block diagram of the wall following algorithm.......................... 45
Figure 21: The Left-Right decision making for pivots...................................... 47
Figure 22: Areas of shelter detection viewed from the robot’s camera............... 48
Figure 23: Unfiltered angular velocity values, $\omega$ ......................................... 50
Figure 24: The components of the RAMBLER algorithm that were implemented on the robot .................................................................................................................. 51
Figure 25: Distance walked along the wall, robot versus cockroach............... 55
Figure 26: Mean distance to the wall, robot versus cockroach.......................... 56
Figure 27: Robot wall departures rates (left) are consistently lower than cockroach wall departure rates (right) ........................................................................................................ 57
Figure 28: Distribution of robot wall departure angles...................................... 57
Figure 29: Percent coverage of the wall and the arena for the robot and cockroach.. 58
Figure 30: Time and distance to shelter, robot versus cockroach................. 60
Figure 31: An individual trace of cockroach barrier departure behavior........... 61
Figure 32: A robot following the barrier and then hitting the opposite wall. The robot did not make it to the shelter in this trial......................................................... 62
Figure 33: Results of the RAMBLER algorithm run in simulation with only those components implemented on the robot .................................................. 66
Figure 34: Cockroaches lose weight by dehydration over time according to a logarithmic function .......................................................... 71
Figure 35: Cockroaches mostly turn away from water (red x,) those that go to drink it start their approach at an average of 52 mm away (black dots)................. 72
Figure 36: Bias before the arena was shrouded

73
Acknowledgements

While no research occurs in a vacuum, this work is distinctly the result of much close collaboration. Even though I am receiving an engineering degree, the number of biologists I need to thank is roughly equal to the number of engineers.

The first acknowledgement must go to the tireless work of Kathryn Daltorio, for providing insights that helped bridge the gaps between animal behavior and robotic algorithms. She also gets my gratitude for reading the early drafts of this document.

Dr. John Bender’s support was incredible at the beginning of this project. In addition to his instruction in animal handling and inspiration to further investigate neuroethology, he was able provide lessons in computer programming and data presentation.

Drs. Roger Quinn and Roy Ritzmann have made the perfect advising team for this project. Having courses with both of them has been a pleasure, and both were able to provide key points of guidance throughout the project.

Dr. Michael Branicky also deserves thanks for serving on my committee, and for his support of Kathryn Daltorio’s work.

The students of the biorobotics course in 2010 and 2011, all deserve thanks for getting the robot working and proving additional biological inspiration. Special thanks go to Nicholas Szczecinski and Victoria Webster, for their work towards the robot’s wall following hardware, software and the deterministic behavior.

The logistical and social support of other members of the Quinn and Ritzmann labs also deserves mention. In particular, my work would have been much more difficult without
Alan Pollack’s advice and assistance. Dr. Richard Bachmann made himself available at some key times to help with robot repairs.

I also owe thanks to the NSF for grant NO. 0845417, HHMI for a fellowship through the SPUR program, DARPA for the M3 program grant W91CRB-11-1-0003 and NASA for the Space Technology Fellowship Program (#NNX11AN15H). These have provided my financial support over the past two and a half years.

Last but certainly not least I’d like to thank Mom and Dad for encouraging me to study engineering, and all of their support throughout my education. Love goes to Heather Lemire for bringing me coffee and other things to keep me awake and sane too many times to count while writing this.
Abstract

By

BRIAN ROBERT TIETZ

Animal behavior is often a model for robotic control, with benefits for both robotics and biology. This research covers a new animal behavior for this category: cockroach shelter-seeking. Cockroach behavior was tracked in a 91 cm by 91 cm arena, and significant trends were identified that form a stochastic navigation algorithm called RAMBLER. Components of RAMBLER were then implemented on a mobile robot, and compared with a deterministic model of the same cockroach behaviors. In the process of programming the robotic model, an interesting behavior was discovered when the cockroach loses contact with a barrier in the arena, posing new questions about animal behavior.
Chapter 1 – Introduction and Goals

1.1 Motivation for this project

In many robotic applications, environmental mapping is critical to success. Often, the robot is designed for an environment where most of the constraints are known by the programmer, therefore allowing for task-specific mapping techniques to be used. An example is the ION Autonomous Lawnmower Competition, in which the programmers are given a list of obstacles and can implement deterministic ways of avoiding them (Hughes et al., 2009) Some commercially available robots, such as the Roomba, are random, and react reflexively when a collision is sensed (iRobot). Robots like the Roomba are prone to getting lost without a map of their environment, while robots like CWRU Cutter can get stuck if their deterministic algorithms fail. Programmers are required to tune between deterministic and random methods for success.

Animals, unlike robots, are capable of quickly adapting to many different environments. Given their adaptive abilities one would expect animals to employ some method of searching for their goals that is simple and robust. Cockroaches have already found a successful balance between sensory input, mapping, and randomness through evolution. Their methods, once determined, could provide inspiration for more robust robotic navigation for applications such as search and rescue. These searching and surveying methods are not yet part of the otherwise large library of biologically-inspired navigation algorithms (Franz & Mallot, 2000)

In addition to providing inspiration for engineering applications, this research adds to the biological literature. Cockroaches are known to stop under shelter during an escape
response (Okada & Toh, 1998) and gather underneath shelters (Jeanson & Deneubourg, 2007), but how an individual cockroach reacts in a non-escape situation was previously unknown. Additionally, while cockroach strategies for negotiating barriers have been documented by Harley, English, & Ritzmann (2009), how the animals reorient to a goal afterward was unknown.

Finally, once these strategies are determined, implementation on a robot allows for comparisons to both deterministic robotic algorithms and the original cockroach data. The results of such testing provide insights that can suggest future biological experiments, furthering our understanding of the animals’ behavior.

1.2 Goals of this Thesis

This thesis develops a model of cockroach goal-seeking behavior, and describes a robotic test platform for cockroach behaviors. In pursuit of this goal I:

(1) Determined a suitable goal for the cockroaches, designed experimental procedures, and collected 177 trials for analysis.
(2) Developed a robotic test platform with tactile and sensory inputs.
(3) Implemented a model of the behaviors observed in (1), and compared the results on the robotic test platform to the cockroach behavior.

1.3 Chapter Summaries

The remainder of this thesis is presented as follows. Chapter 2 provides background in robotic models of animal behavior, as well as biological background in some relevant topics. Chapter 3 describes the cockroach experimental setup, and the behavioral model that resulted from these experiments. Chapter 4 describes the hardware and arena of the
robot. Chapter 5 discusses the specific behavioral models implemented on the robot, and Chapter 6 compares the performance of those models to the cockroach behavior. Chapter 7 discusses conclusions and future work.
Chapter 2 - Background

The first section of this Chapter deals with the challenges of unknown environments and some current engineering solutions. The second section describes previous work that inspired us to use cockroaches as a model animal. The third section deals with the cockroach shade response, our inspiration for a target. During experiments, we observed the cockroaches displaying two categories of behavior, based on whether they were able to contact a wall with their antennae (Bender et al., 2011; Daltorio et al., 2012). Thus the fourth section discusses wall following and obstacle navigation. The final goal of this thesis is to provide inspiration for further biological experiments; therefore, the final background section describes prior examples of this.

2.1 Robotic Approaches for Unknown Environments

Environments that are in need of mobile robots are typically either dangerous or inaccessible for humans, such as disaster or extraterrestrial areas. These are also environments in which it is less likely a human programmer or operator will have prior knowledge of conditions. This section highlights the difficulties of designing robots for such situations, and provides the motivation for developing a randomized, cockroach-inspired algorithm but it is not meant to be a comprehensive review of approaches in hazardous environments.

One approach for robots in these situations is for a human user to operate the robots remotely. Approaches of this kind have become quite sophisticated, allowing for haptic (physical, mechanical) feedback on the user’s controls, in addition to visual information (Glover, Russell, White, & Miller, 2009). However, an operator can typically only provide direct motion control to one robot at a time. If robots have more autonomy,
operators can control multiple robots at the behavioral or mission levels (Birk & Schwertfeger, 2009). The system developed by Birk et al. allows for operators to interact with the robot on all three levels, and allows for various autonomous behaviors to be used.

Strategies for autonomous robots vary based on application, and also by the level of task that they address. High-level tasks can have a robot attempt to cover an entire area (Choset, 2001) or dynamically map a changing environment (Milford & Wyeth, 2009). Lower-level behaviors can include seeking a specific goal using biological principles (Webb, 1995), or avoiding obstacles in the robot’s path, using a strategy such as a BUG algorithm (Ng & Bräunl, 2007). The robot will likely have multiple goals, such as finding its way to a target (goal 1), and avoiding collisions (goal 2). The robot will have to be programmed with a strategy to balance these goals, for example, the subsumption architecture (Brooks, 1986).

One problem with many of the above methods, especially the high-level methods, is their use of memory to keep track of previously visited areas and previously encountered obstacles. The cost of implementing these algorithms can increase rapidly, especially when multiple robots are involved. Work by Gage (1993) provides equations for tradeoffs in sensory effectiveness and cost for a multi-robot system where the robots have imperfect sensors. Gage shows that a stochastic search detects, on average, a larger number of targets than a coordinated search if the accuracy and, presumably, cost of the sensors in use decreases. Biological systems also have to deal with noisy sensors, balance goals, and run on limited memory requirements, suggesting they may provide inspiration for solutions.
2.2 Inspiration from Cockroaches

A review of animal navigation behaviors implemented in robots is provided by Franz and Mallot (2000), including trail following, aiming and path integration. The work they cite shows that a number of behaviors valuable to a robot in an unknown environment can be inspired by animal behavior. However, they claim the basic searching behavior has yet to be implemented in a robot. The question is how to tune between deterministic and random behaviors. Many animal behaviors could provide inspiration for this behavior, including foraging behaviors (Viswanathan et al., 1999) or home range behaviors (Börger, Dalziel, & Fryxell, 2008). Drawing inspiration from these sources in our analysis, we chose to collect data from cockroaches due to other insights they have already provided in experimental biology. These include wall following (Cowan, Ma, Cutkosky, & Full, 2005), and climbing or tunneling (Harley et al. 2009). In addition, cockroach locomotion has provided inspiration for legged locomotion in robotics (Quinn & Ritzmann, 1998) and the climbing versus tunneling decision making has also been implemented in a robot (Lewinger, Harley, Ritzmann, Branicky, & Quinn, 2005). Lewinger, Harley and Cowan’s papers show that inspiration from the cockroach’s antenna can provide simple and effective methods for navigating over, under, and around obstacles for a robot. Including the visual system could provide a multisensory, goal-directed behavior. In fact, Harley’s work foreshadows the next section on shade response, as cockroaches were found to be more likely to tunnel under the obstacle in the light.

2.3 The Cockroach Shade Response

Once cockroaches were selected as a model animal, I needed a suitable goal for them. Pheromone cues produce changes in walking speed and direction in Periplaneta
americana (Nishiyama, Okada, & Toh, 2007), but they have not yet been synthesized for our specific species of cockroach, *Blaberus discoidalis* (Ritzmann, personal communication). Water as a goal for dehydrated cockroaches was tested first, but was not acceptable in this study for reasons described briefly in Chapter 3 and in detailed in Appendix A. The following literature suggested a dark shelter in a light arena as a goal.

A shadow can cease escape response (rapid running) (Meyer David, J, Margiotta, Joseph F., Walcott, 1981; Okada & Toh, 1998). Ritzmann et al. also showed that the thoracic interneurons associated with escape response are sensitive to light (1991). This implies that there is a specific and rapid coupling from the cockroach’s visual system to its motor system. More systematic behaviors like goal-seeking might cue from similar inputs. More recent papers have shown that groups of cockroaches tend to congregate in dark places (Jeanson and Deneubourg 2007). Small robots, treated with cockroach pheromones and programmed to behave like cockroaches, can also influence the shelter selection site (Halloy et al., 2007). These papers show that cockroaches prefer to be under a dark shelter in groups, but it leaves the question of how an individual cockroach responds. This response was first documented in Daltorio et al. as a part of this project (2012).

### 2.4 Wall Following

A necessary component of exploring an unknown environment is navigating around obstacles. In nature, this is commonly achieved by tactile wall following. Wall following behaviors are especially common in blind animals (Sharma, Coombs, Patton, & Burt de Perera, 2009). As discussed above, vision could provide goal-seeking information. Modeling mouse behavior, Yokoi, Fend and Pfeifer (2004) have developed a whisker system that navigates around obstacles. They have not yet implemented an active
targeting system. Cockroaches use antenna to perceive a large number of cues, ranging from chemical to tactile to temperature sensors, are able to move their antennae in three dimensions, and receive inputs through both passive stimulation and active searching (Okada & Toh, 2006; Okada, Kanamaru, & Toh, 2002). Biologically inspired roboticists often look to cockroach antennae as an example of tactile wall following (Cowan et al. 2005).

The cockroach antennae use hair plates at the base of their antenna and the flagellum along the length for tactile input (Lee et al., 2008; Okada et al., 2002). The wall following behavior is quite fast, animals are capable of turning with each step, achieving up to 25 turns per second while running in response to the wall’s geometry (Camhi & Johnson, 1999). Cockroaches are also capable of sensing an object before a collision at escape speeds (Baba, Tsukada, & Comer, 2010). These behaviors were modeled using strain gauges to model the flagellum by Cowan et al. (2005). However, strain gauges can prove difficult to align and attach, since they measure forces in their vertical direction far better than the horizontal. Our somewhat simpler design uses potentiometers to model the hair plates (Szczecinski et al., 2012) and is discussed in Section 4.3.

2.5 Robots as tools for testing biological hypotheses

Biologically inspired robots can also provide a test platform for biologists. Webb (2001) discusses this extensively, and reiterates the importance of comparing a robot’s performance to that of the animal. This rigor in model construction often reveals the need for further data from the animal, which in turn makes the model more accurate and effective. Building a large robot based on cockroach measurements and kinematics lead to the discovery of the need for joint stiffening in anticipation of loading (Quinn and
Ritzmann 1998). While implementing stick insect and cockroach inter-joint coordination methods in a robot leg (Lewinger, Rutter, Blümel, Büschges, & Quinn, 2006), hypotheses were generated about descending commands for turning motions (Rutter, 2010). Hypotheses of fly flight fluid dynamics were tested on a scaled robotic model (Lentink & Dickinson, 2009). Implementing cockroach goal-seeking behavior on a robot could raise new questions about sensory integration and decision making in the animal.
Chapter 3 – Methods 1: Collecting the Animal Data

This Chapter details the methods behind collecting and analyzing cockroach data to discover a biologically inspired goal-seeking algorithm. Sections 3.1 through 3.3 detail the experimental setup and methods, Sections 3.4 and 3.5 show the results of those experiments. Section 3.6 covers data of cockroaches seeking shelter with a clear acrylic barrier, which we have only analyzed qualitatively thus far.

3.1 Animals

Female adult cockroaches (*Blaberus discoidalis*) were used in all experiments. Animals were selected at random from colonies in the Ritzmann lab, housed in five-gallon plastic barrels and given food and water *ad libitum* while on a 12-hour light-dark cycle. Animals were inspected for normal antennal length (approx. 2.5 cm) and only apparently healthy animals were chosen.

![Figure 1: The cockroach *Blaberus discoidalis*, which was used in experiments](image-url)
3.2 The Arena

3.2.1 Goal Selection

Our first attempt at a goal for cockroaches was placing dehydrated cockroaches in a 90 cm square arena with water. While cockroaches found the water in 13 out of 17 of the trials, they didn’t go directly toward the shelter until they were, on average, within 5.2 cm of it. We decided this was insufficient for the longer range navigation behaviors we were interested in, so we were inspired by Deneubourg et al. (2007) to try a dark shelter as an attractant. More information on the water trials is available in Appendix A.

3.2.2 The Light Arena

Our second arena is a 91.4 cm by 91.4 cm square, with walls 10 cm high (see Figures 2 and 3 above). The inside of the arena was spray-painted white. The top 3 – 5 cm of the walls were coated with Vaseline to prevent the cockroaches from escaping. There is a 10 cm square starting chamber on the side of the arena where the cockroach is placed before
trials begin. In Appendix A-2, we show that shrouding the arena by black cloth removes biasing visual cues. The shelter is a 15 cm by 15 cm square piece of thin red plastic film, taped to wooden mixing rods, and supported either on the wall or on pegs.

The arena was lit by four 100 W incandescent light bulbs, placed at each corner approximately 75 cm above the arena. A fluorescent, 90 W equivalent, flood lamp is placed 15 cm above the starting chamber to discourage cockroaches from staying there. Lighting conditions vary within the arena between 1200 and 2000 Lux, as measured by an EXTECH Light Meter, with the light in the starting chamber at 7600 Lux. Underneath the shelter the light was recorded at 300 Lux. Ambient conditions under fluorescent office lights in the room were between 490 and 710 Lux, and the cockroaches’ storage bin during trials was recorded between 50 and 110 Lux.

3.3 Trial Procedure

Cockroaches were placed in the testing room to acclimate to lighting and ambient air conditions one hour ahead of the trials. Before each trial the arena was cleaned with a detergent mixture and water, and then allowed to dry for four minutes. The shelter (as necessary) and gate to the starting chamber were then placed in the arena, and the cockroach was placed in the starting chamber. Afterward, the camera was started as soon as possible. Twenty seconds later the gate was removed and the cockroach was free to enter the arena. If the cockroach did not enter within four minutes the trial was stopped without taking data. Cockroaches entered the arena by their own decision at a median of 11.1 seconds after the gate was opened. Trials were started when the cockroach entered the arena and ended one minute later. In the event that the cockroach re-entered the starting chamber (which was outside of the camera’s view) or escaped the arena, videos
of at least 40 seconds were kept in the data set, while other trials were discarded. The cockroaches were never tapped or otherwise prodded to encourage movement.

3.3.1 Postprocessing

Trials were recorded at 20 Hz on a Basler™ Firewire Camera (Figure 4), suspended 140 cm above the center of the arena, and then recorded to the computer’s hard drive using the MOTMOT image acquisition package (Straw & Dickinson, 2009). Tracking was performed by CTrax, an open source software package developed for flies (Branson, Robie, J. Bender, Perona, & Dickinson, 2009). Additional post processing was performed in Fixerrors, a MATLAB toolbox developed with CTrax. Final processing and data analysis of the cockroach trials were done in MATLAB.

![Figure 4: The overhead camera, suspended above the arena.](image)

3.4 Cockroach Behavior

Our first attempt at modeling cockroach shelter-seeking behavior correlated the angular velocity of the cockroach with parameters like distance from the wall and angle to the
shelter. However, this fit became exceedingly difficult to implement on a physical robot (see Section 5.4 for additional details). Thus, we fit the curves of the cockroach in the open arena to a series of pivots (zero radius turns) and straight lines, which the robot could execute easily. Analyzing animal data in this fashion had been done previously by M.K. Tourtellot, R.D. Collins and W.J. Bell (1991). Additional details of the analysis algorithm are available in Daltorio et al. (2012).

3.4.1 Differences in Behavior along Wall

![Graph showing cockroaches' speed decreases at close proximity to a wall](image)

Figure 5: Cockroaches’ speed decreases at close proximity to a wall. This is due to a change in gait when the cockroaches are in antennal contact with the wall (Bender et al. 2011). The lines represent averages at individual speeds, while the boxes are 90% bootstrap confidences on the two threshold values, above and below 40mm (antennal contact distance) Figure from Daltorio et al. (2012).

Our data showed that when the center of the cockroach is within 40 mm of the wall, its speed drops dramatically (see Figure 5). The cockroaches in our arena spent, on average, 49.2% of their time along the wall. If a wall departure is considered a pivot, the
probability of a pivot along the wall is 1/3 to 1/10 of what it would be in the open arena. This caused us to consider the cockroaches’ behavior along the wall as a separate state.

### 3.4.2 Wall Departures and Turnarounds

The cockroach has a small chance of departing the wall (2 departures per meter walked) under normal conditions. This value increases if the cockroach is facing away from the shelter and decreases when the cockroach is facing within 90° of the shelter (see Figure 6, Daltorio et al. 2012).

![Figure 6: Cockroaches are more likely to depart the wall if they are facing away from the shelter. Boxes are 90% bootstrap confidences, with grey representing the control cases, and blue representing the average of all three shelter locations (Daltorio et al. 2012).](image)
Figure 7: The angle of the cockroach’s body to the wall it is departing, when it is 60 mm along the wall (1.5x the wall threshold). The distributions for the control (grey, black line), and with shelter (blue, blue line), are not significantly different.

There is also a small probability of the cockroach stopping and making a 180° turn along the wall. This probability decreases significantly when the animal is facing within ±33° of the shelter, see Figure 8.
Figure 8: Cockroaches have some chance of turning 180 degrees while on the wall, and this chance decreases dramatically when facing the shelter. Blue and grey boxes same as previous figures (Daltorio et al. 2012).

Behavior at the corners was slightly different from that on the walls, and was thus considered separately. When in the corners, the cockroach had a 29% chance of exiting along the same wall it came in on (Daltorio et al. 2012).

3.4.3 Arena Pivots

When in the open arena, we fit pivots to the cockroaches’ paths. The likelihood of pivots varied with the angle to the shelter, the rate drops from 11.7 to 5.7 pivots per meter walked if the cockroach is facing within ±12° of facing the shelter (see Figure 8). Additionally, there is a small probability (10%) that the animal will continue traveling straight until it reaches the shelter if it is facing within ±17°.
Figure 9: There is a small probability the animal will continue straight to the shelter after a pivot, without making any further pivots. Only one such case was recorded when a shelter was not present. Blue and grey boxes same as previous figures (Daltorio et al. 2012).

While the magnitude of pivots was not noticeably affected by the shelter, the direction was. Based on the fit parameters, cockroaches were likely to continue turning in the direction of their previous pivot (71%, Figure 10). However, if the cockroach is facing the shelter ±33°, the probability of turning in the previous direction is only 51%, if that direction would move the cockroach away from the shelter (Figure 10.)
3.4.4 Exploration Efficiency

In order to evaluate the exploration efficiency of the cockroach we compared the time to shelter, distance walked to the shelter location, and the percent of the arena covered for the control data versus the shelter cases. This is shown for the SC shelter location in Figure 11. The other two locations are shown alongside their respective robot trials in Figures 26 and 27 in Sections 6.1 and 6.2. In all cases, the presence of a shelter significantly reduces both the time to arrive and distance walked to the shelter location. Additionally, while the control and shelter cockroaches both cover similar percentages of
the open arena and the wall over the entire trial, the percentage covered before reaching the shelter location is lower.

Figure 11: The exploration efficiency metrics of the cockroach with the shelter in the SC, or mid-back location. A: The time to shelter is significantly lower when a shelter is present. B, C: The control and variable cases cover similar percentages of the wall and arena, but the variable cases cover less of the wall and arena before they reach the shelter. D: Distance walked during the entire trial is similar for each case, but the distance is reduced significantly before reaching a shelter location when a shelter is present. E: All of this is due to path, since walking speeds are not significantly different.

3.4.5 Time Spent under Shelter

Once cockroaches reach the shelter, they tend to pause there. When a shelter was present in the mid-back location, cockroaches spent an average of 2.9 ± 6.9 seconds there, as opposed to .48 ± 1.6 seconds when a shelter was not present. Additionally, they spend more time in the shelter locations than other locations in the arena as shown by the
bottom row in the plots below for the SC location (Figures 12). Red indicates more time spent in a region.

Figure 12: Cockroaches spend more time in the more red areas, which in the bottom maps indicate their preference for shelter. A, C, E, G represent control data, B, D, F and H represent variable data. “Before” coverage (C, D, G, H) maps indicate areas visited before the shelter is reached. Coverage maps (A-D) use an analog sum of time in location, visited maps (E-H) are digital for individual trials visiting a location, and then averaged over all trials.
3.5 Modeling Behavior in Simulation

The above trends were combined into a computer simulation, which fit a series of straights and pivots to the cockroach tracks, an example of which is shown in Figure 13. This fit produced an algorithm that guides the simulated agent toward the shelter. We called this algorithm Randomized Algorithm Mimicking Biased Lone Exploration in Cockroaches or RAMBLER, shown in Figure 14. Components of RAMBLER were implemented on the robot, and are discussed in Section 5.4.
Figure 14: The RAMBLER Algorithm combining the trends of Figures 5 - 10. Certain transitions are deterministic (encountering a wall or exiting the pivot state), but others are probabilistic, given by the distributions in the previous figures (Daltorio et al 2012).

3.6 Barrier Data

As part of a goal-seeking algorithm, the agent must be able to deal with obstacles between it and its goal. For the cockroach, we needed to maintain a visual goal of the shelter, while providing a physical obstacle. Thus, we placed clear acrylic barriers in the arena.

The barriers were made from 10 cm high clear acrylic walls. The barrier is 46 cm long, and has a 4 cm perpendicular extension on one end for support. Additional barriers 80 cm
and 70 cm long were used by Porr and Richards in their experiments, (see Figure 15, Szczecinski et al. 2012). All barriers were painted black on the top for increased overhead camera visibility. Barriers, when in use, were cleaned every 4-6 trials with plastic cleaner, before the arena was cleaned, and placed in the arena at the same time as the shelter.

The barrier data has not been modeled as quantitatively as the shelter-seeking behavior, but a few trends stand out. The cockroaches are successful in 40 out of 43 trials where a 45 cm barrier (half the length of the arena) is placed both in the middle of the arena and along the side wall. Richards and Porr tested an 80 cm barrier, and the success rate dropped to 6 out of 20. In 8 of the trials in which it didn’t reach shelter, they saw that the cockroach hit the barrier at a mean of 76 cm from the wall, with a standard deviation of 6 cm. In addition, in 12 of the 20 control trials, the cockroach crossed the barrier, versus half that percentage in the variable trials. Richards and Porr then made a barrier a standard deviation shorter than the contact point (70 cm barrier), and the success rate went back up to 16 out of 20. This behavior was modeled by Szczecinski and Webster.
(2012) in a deterministic robot algorithm. More information about the cockroach barrier trials is available in Section 6.4
Chapter 4: Methods 2 – Robot and Algorithms

4.1 Robot Hardware - Chassis and Motors

Figure 16: The robot that is used as a test platform.

The robot chassis was a standard Mini Whegs chassis purchased from BioRobots, LLC and then modified to better execute models of cockroach behavior. The chassis is 6.5 inches long and 4.5 inches wide, a 7.5 inch acrylic tail was added to assist with overhead video tracking. The wheel-legs caused vibration in the antennae and they were not needed on the smooth terrain in the experiments, so they were replaced with four 3.5 inch diameter acrylic wheels, with rubber bands as rims for added traction. The wheels are powered by two electric drive motors, one per side. Each side is kept in sync with a timing belt. Rechargeable lithium ion batteries provide onboard power.
4.2 Microprocessor and Electronics

A 500Mhz Blackfin microprocessor was purchased from Surveyor Robotics (surveyor.com) as part of the SRV-1 camera platform. The platform also includes a 1.3 megapixel camera that can be run from 160x128 to 1280x1024 pixels, hereafter referred to as the onboard camera. An SRV-1 RCM expansion board was purchased to allow for analog inputs, such as the antenna. The SRV-1 platform also includes a matchport wireless model that communicates over 802.11g wifi networks. This platform includes open source firmware.

4.3 Antenna

![Antenna Image]

**Figure 17:** Close up of the Robot’s acrylic antenna from above. The spring and top of potentiometer are also visible.

Initial digital antennae made of spring steel were constructed as part of the Spring 2010 Biorobotics Team Research course, taught by Profs. Quinn and Ritzmann. The original wheel-legs caused too much vibration in the spring steel antennae, so the Whegs were replaced with wheels. However, the spring steel still vibrated too much. Thus, the team in the Spring 2011 class took inspiration from the Cowan lab’s work on wall following (2005). Instead of using strain gauges to measure deflection along a flexible antenna the team decided that measuring deflection at the base of the antenna would be sufficient. Antennae that mimic the shape of the cockroach antennae during wall following were
constructed from acrylic, and then placed on 5 kΩ linear potentiometers. A torsional spring on each antenna restore them to a “home” position against which deviations can be measured. The potentiometers provide Coulomb damping which prevents excessive vibration and false readings. This setup, combined with the RCM board allows the robot to perform PID control on the position of the antenna, which is discussed further in Section 5.1.

4.4 The Robot Arena

Figure 18: The robot in its arena. The shelter is in the SC location.

Since the robot is five times larger than the cockroach, the arena needed to be scaled up as well (see Chapter 6 for chart 1 of comparison values). Due to the shape of the robot, I chose width as the appropriate scaling parameter for distance. The cockroach is an
average of 2.15 cm wide at the abdomen, while the robot is 11.7 cm wide. This leads to a ratio of 5.44:1, which I rounded to 5:1. This means the robot would need a 15 foot (450 cm) square arena. For the arena’s location, I chose Glennan 716, a classroom near the robotics lab. The chairs were easily movable, which allowed me to create sufficient space for the robot’s arena. Additionally, I was able to easily re-mount the off-board camera at the start of every day’s trials, allowing it to be used for biological experiments or stored in a secure location between robot tests.

The arena was surrounded by the room’s walls on two sides, and plywood walls on the other two. The shelter was black poster board on the floor and wall. Barriers were also made of plywood, since acrylic of sufficient length would have been prohibitively expensive.

There were two disadvantages to using this classroom. First, the chalkboard and a small brown strip between the wall and the floor provide another source of dark pixel values for the robot’s camera. More significantly, the 10 foot ceilings were too low for an overhead camera mount. This issue is discussed further in the next section on camera calibration.

4.5 Overhead Camera and Calibration

The same Basler™ Firewire camera used to track the cockroaches was used for overhead tracking of the robot. To connect to a laptop, this required a six-pin Firewire express card and a 12 v Ac/Dc adapter. The camera was mounted on an arm 11 ft in front of and 8.5 ft above the robot’s arena. It viewed the arena 7 feet to the camera’s right of the arena’s center. This placement, while awkward, was necessary to get the entire arena in the camera’s field of view.
A general issue in camera data, but exacerbated by the camera placement, is that of converting pixel coordinates to real coordinates. Camera calibration requires two sets of parameters – intrinsic parameters based on the properties of the camera’s lens, and extrinsic properties based on the position of the camera relative to what it’s recording (Bouguet, 2010). Given the angle of the camera relative to the robot arena, extrinsic parameters cause much more of the error than intrinsic parameters.

Since the camera had to be moved between every day’s robot testing, I required a faster solution than the typical calibration rig setup. At the expense of a small amount of accuracy, we discovered a linear algebra fit that allowed for a fast calibration setup. Calibration points were collected using the one-foot square floor tiles as coordinates. The functions below use four matrices:

1. Desired corners (DC), the coordinates in real space
2. Pixel corners (PC), the coordinates in pixel space
3. Pixel plus (PPT), a transform matrix
4. A, the combination of the transform matrix and the desired corners.

The calibration coordinates are used to generate A as follows:

\[
A = PPT^{-1} \cdot DC
\]  \hspace{1cm} (1)

This is represented as “A=numpy.linalg.pinv(pixel_plus)*desired_corners” in Python (Jones, 2001). Where PPT or pixel_plus is:

\[
PPT = \begin{bmatrix}
    x_i, y_i, \sqrt{x_i}, \sqrt{y_i}, \sqrt{x \cdot y_i} & 1 \\
\end{bmatrix}
\]  \hspace{1cm} (2)

Then calibrated real points can be computed using:
This transform produces a 0.3 foot mean squared error, which was deemed tolerable for this application. The results are shown in Figure 19.

**Figure 19:** The forced fit calibration. Black points are calibration points, blue points are test points. The red and violet lines are the fit with pixel points.

### 4.6 FlyTrax

In addition to recording video directly to a hard drive in a .fmf format, the Motmot package has a real-time tracking feature called FlyTrax. FlyTrax uses background subtraction against a static captured image to provide real time tracking of a single object, and is capable of relaying that data over a TCP/IP socket (Fry, Rohrseitz, Straw, & Dickinson, 2008). Outputs include x and y position in pixel coordinates, and slope of the
tracked object. The 180 degree information in slope can be converted into a 360 degree heading using information about the robot’s x and y velocities, since the robot is normally travelling forward. These values are set by the operator at the start of each track to ensure a proper initial orientation.

One problem encountered in this tracking setup is that the camera is not placed directly over the arena; therefore the heading of the robot can be obscured as its profile changes relative to the camera. This was mitigated by not using FlyTrax in the real time control of the robot for the final algorithm (RAMBLER).

4.7 Communication between the computer and the robot

For some of the deterministic algorithms, the computer needs to be able to provide information from the overhead camera to the robot. The Python programs running on the computer are capable of communicating with the picoC programs on the robot over the wireless network. Sockets are set up in Python for communicating with the robot’s IP address, and port 10001 (customizable in the SRV-1 firmware). The computer can send strings or integers, which are then parsed by the robot. Formatting is important, as the robot only interprets an integer after a non-integer is received, and a string after the ASCII null character (“char(0)” in Python) is received. It is more robust to send strings to the robot and parse them into integers, because if a packet is lost, then it is lost as a unit. If an integer is lost, it is difficult to determine the proper parsing of the next integer.

The robot can communicate back to the computer by using the “printf” function. This is normally just used to record the robot’s internal state, and can be easily written to a text file by the Python program. It is possible to parse out data and use it in the program, but
since everything is received in a string format this nearly requires try-catch statements to avoid frequent crashes.
Chapter 5 Models of Cockroach Behavior

In this Chapter, I discuss modeling two of the cockroaches’ sensory influences – tactile input from the wall and visual input from its compound eyes and ocelli. The hardware is discussed in Chapter 4, and each required its own set of algorithms. Producing a coherent behavioral output requires tuning between these two influences, two approaches are discussed in Sections 5.3 and 5.4.

5.1 Wall Following

The primary mode of wall following is with a proportional-integral-derivative (PID) feedback loop implemented on the robot by Szczecinski and Webster (2012). At the beginning of each trial and under certain conditions during the trial, the robot records a home position for the antennae. Deviations from this position, recorded as voltage differences, are then recorded as an error \( E_\theta \). Control can be done simply on this error (the P term), but additional stability and responsiveness can be obtained by also looking at the derivative and integral terms. The equation for this type of control is typically represented as (S. Bennett, 1984):

\[
C_A = P \cdot E_\theta + I \cdot \int_0^t E_\theta \cdot d\tau + D \cdot \frac{dE_\theta}{dt}
\] (4)

\( C_A \) is the total change to the motor speeds (in this case) and P, I and D are constant proportions that determine the weights of the respective position, integral and differential terms. Where Pure PID was not sufficient to solve the wall following problem, the robot was likely to get stuck during head on collisions and in corners. To solve this problem, if an antenna is ever stuck (displaced when the derivative term is zero) the robot backs away from the wall and then turns at an angle that allows it to follow the wall. While
backing away is not typical behavior for a cockroach under normal conditions, it is seen during the escape response (Baba et al., 2010). After a head on collision, the robot backs away from the wall and turns away from the antenna that was depressed further. This is
similar to cockroaches turning in the direction opposite the first antennal contact (Harley, English, & Ritzmann, 2009). A block diagram of this behavior is shown in Figure 20.

In the RAMBLER algorithm (Section 5.4) wall following is a discrete state, so the parameters originally implemented by Szczecinski and Webster were modified so the robot turned to meet the wall sooner, however, this increased the amount of time the robot spent in corners. There is also a certain probability of the robot executing a turn away from the wall. Finally, if the robot loses contact with the wall for twenty time-steps (value hand tuned, see Appendix B for more detail) it leaves the wall following state.

5.2 Vision – Onboard Camera

The robot contains several algorithms using its onboard camera for shelter-seeking, moving the robot toward the shelter in the arena, and shelter detection, the robot indicating it has found the shelter by pausing briefly. The onboard camera uses a YUV color scheme, with Y indicating the relative brightness of pixels, as calibrated by the initial camera view. The first algorithm used for deterministic shelter-seeking was the SRV-1’s built in blob finding algorithm, and was implemented in Szczecinski et al. (2012). The algorithm looks for the centroid of an area within a specific color bin, which for shelter-seeking can be all pixels below a certain Y value. This allows for the robot to do P control on the x position of the shelter’s center. Implementing I and D terms was determined unnecessary for this particular algorithm.

One disadvantage of the blob finding algorithm is it can be easily thrown off by noisy environments. For example, the classroom has a dark liner between the wall and the floor, which would extend the x values of the blob across the screen for certain shelter
locations. An alternative algorithm to bias the robot is to sum the Y values of all pixels within a certain area. There is a built in SRV-1 function to obtain the YUV values of a given pixel, which could be used in a loop to compute this sum. However, upon implementation this proved prohibitively slow (17 seconds for half the screen at 160 x 128 pixel resolution). I wrote the summing loop directly into the robot’s firmware, and this minor change led to 1000x faster performance (17 ms). For shelter-seeking, this meant I could bias the robot based on whether the left side or the right side of its screen was darker. This algorithm proved acceptably accurate in initial testing, but was not later quantitatively tested for false positives with data from actual trials.

Figure 21: The Left-Right decision making for pivots. This is based on the RAMBLER algorithm as discussed in Section 3.5. This is based on the cockroaches' pivot direction choices from Figure 10. The diagram for the previous turn on the right would have the same probabilities, left and right would need to be swapped in every case. *The "shelter not detected state" was not implemented in the robot data on this thesis, but is present in the data, see future work Section 7.2.
Shelter detection was performed by summing the Y values of regions at the bottom, left, and right sides of the screen, as well as the total of the entire screen. If the bottom, and the left or the right sums were lower than the mean over the total area than the robot anticipated it was on the shelter. If these conditions were still true 200 ms later, the robot pauses before continuing exploration.

Figure 22: Areas of shelter detection viewed from the robot’s camera. For the robot to declare an area to be shelter, the bottom area and either the left or the right side need to be darker than the mean of the whole frame, or the mean needs to be sufficiently dark.

5.3 Behavioral Weighting 1 – Deterministic

Three deterministic algorithms were implemented on the robot, based on the different sensory inputs: just the overhead camera, onboard camera with antennae, and overhead camera with the antennae.

Before the completion of the antennae, the simplest deterministic algorithms used P control on the robot’s heading. The computer provided motor left and motor right values to the robot. These algorithms were capable of directing the robot to a specific point in
pixel or real coordinates, and the robot would stop when it was sufficiently close to the target. With the addition of antennae, the robot needed to calculate its own motor speeds in order to have a fast enough response time. Thus, the input from the overhead camera was changed to a bias similar to that of the onboard camera in the algorithm by Szczecinski and Webster (2012).

The key of the Szczecinski and Webster algorithm is its dynamic sensory integration (2012). When the robot has a clear path to the shelter, it raises the influence of the visual system to three times that of the antennal system, directing the robot to the shelter. However, when a wall is between the robot and the shelter, the visual influence is cut to a third of the antennal influence. The specific weighting values were hand tuned.

The overhead camera was combined with the antenna by replacing the blob finding onboard camera with a bias signal the robot could parse. In addition to the left/right P control, the computer could also initiate pivots by telling the robot to allow negative motor values. This tightened the control when the robot was away from the walls.

### 5.4 Cockroach-Inspired Algorithm - RAMBLER

As discussed in Section 3.5, the cockroaches’ movements have been generalized into an algorithm called RAMBLER, which is shown in a state diagram in Figure 14. The components actually implemented on the robot are shown in Figure 24. A fit of pivots and straights was chosen because controlling the robot’s angular velocity proved unwieldy, since angular velocity is a derivative of angular position, and as mentioned previously the tracking setup has trouble with this. Figure 23 is a plot of the robot’s
measured angular velocity over time while the robot is turning at a constant motor speed, demonstrating how difficult it would be to filter or control based on these data.

![Unfiltered angular velocity values](image)

**Figure 23:** Unfiltered angular velocity values, obtained by tracking via the overhead camera, while the robot’s motors are set at a constant rate. This was deemed too noisy for accurate control.

Certain components of RAMBLER were straightforward to implement on the robot, while others were difficult given the limitations of the robot’s camera. The distributions of turns and wall departure angles were implemented using a weighted random number generator. The robot included the general case pivot probability in the open arena, but it was not possible to determine the shelter’s location from a distance using the Y-value averaging algorithms discussed in Section 5.2. The wall departure probability was constant for similar reasons. The robot’s code did include a “go directly to shelter” option when the robot thought it detected the shelter from a distance, and continued driving straight until it found a shelter or encountered the wall, as consistent with data in Figure
Finally, wall turnarounds and corner behavior were handled in a wall following state governed by the control system discussed in Section 5.1, and the specific turn-around probabilities were not used.

5.4.1 Unbiased Random Algorithms

The RAMBLER code can also be modified in a straightforward fashion to generate a correlated random walk as opposed to the biased RAMBLER algorithm. All this requires
is removing the influence of the onboard camera on pivot direction, and disabling the “go
directly to shelter” code. Other random walks can be generated by changing the pivot
distributions, but sufficient data was not taken with these settings.

5.5 Robot Trial Procedure

Robot trials were run in a similar fashion to the cockroach trials, except that there was no
need to clean the arena between trials. FlyTrax was used to track the movements in real
time, but the trials were also recorded as a backup. Cockroaches had sixty seconds to
explore the arena, but the cockroach also moves faster than the robot. The cockroaches’
median speed was 19 cm/s, thus they could cover their 91.4 cm arena in 4.57 seconds.
The robot is always run at its top speed of 1.92 ft/s, thus it would cover its 15 foot arena
in 7.82 seconds. This yields a scaling factor of 1.71, which rounds to the robot having
100 seconds to explore the arena.
Chapter 6 – Model Results

This chapter can be summarized in the chart below, which compares the behavior of the cockroach to the robot and simulation models. Additional details and figures for most metrics are provided in the further sections. The distance scale is 5:1, based loosely on robot width, and the time scale is 1.6:1, based on the robot’s speed. All robot data was analyzed using the same MATLAB code as the cockroach data.

Table 1: Summary of Cockroach versus Robot Data

<table>
<thead>
<tr>
<th></th>
<th>Cockroach</th>
<th>Robot</th>
<th>Robot (scaled)</th>
<th>Units</th>
<th>Ratio</th>
<th>Ratio (scaled)</th>
<th>Figure Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Length</td>
<td>49</td>
<td>165</td>
<td>71</td>
<td>mm</td>
<td>3.37</td>
<td>1.45</td>
<td>2, 14</td>
</tr>
<tr>
<td>Agent Width</td>
<td>21.5</td>
<td>114</td>
<td>28.5</td>
<td>mm</td>
<td>5.30</td>
<td>1.33</td>
<td>2, 14</td>
</tr>
<tr>
<td>Arena Size (length of one side)</td>
<td>914</td>
<td>4572</td>
<td>914.4</td>
<td>mm</td>
<td>5.00</td>
<td>1.00</td>
<td>2, 14</td>
</tr>
<tr>
<td>Antenna Length</td>
<td>24.2</td>
<td>127</td>
<td>25.4</td>
<td>mm</td>
<td>5.25</td>
<td>1.05</td>
<td>2, 14</td>
</tr>
<tr>
<td>Antennal Contact Distance Range (From center of body)</td>
<td>34.95</td>
<td>101.6</td>
<td>34.57</td>
<td>mm</td>
<td>2.91</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>Narrowest Corridor Navigable</td>
<td>21.5</td>
<td>228.6</td>
<td>45.72</td>
<td>mm</td>
<td>10.63</td>
<td>2.13</td>
<td></td>
</tr>
<tr>
<td>Horizontal Visible Angle¹</td>
<td>240</td>
<td>66</td>
<td>66</td>
<td>deg</td>
<td>N/A</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>Mean Arena Speed - excluding stops²</td>
<td>214.5</td>
<td>584</td>
<td>172</td>
<td>mm/s</td>
<td>N/A</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>Mean Wall Speed - excluding stops</td>
<td>119.13</td>
<td>431.2</td>
<td>127</td>
<td>mm/s</td>
<td>4.90</td>
<td>1.07</td>
<td></td>
</tr>
<tr>
<td>Radio - Arena Speed:Wall Speed</td>
<td>1.80</td>
<td>1.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stopped Time</td>
<td>15.77</td>
<td>7.328</td>
<td>4.58</td>
<td>s</td>
<td>N/A</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>Total Trial Time</td>
<td>60</td>
<td>100</td>
<td>60</td>
<td>s</td>
<td>1.67</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Time Along Wall</td>
<td>49.21</td>
<td>N/A</td>
<td>68.7</td>
<td>%</td>
<td>N/A</td>
<td>1.40</td>
<td></td>
</tr>
<tr>
<td>Percent of distance walked along wall</td>
<td>26.98</td>
<td>N/A</td>
<td>58</td>
<td>%</td>
<td>N/A</td>
<td>2.15</td>
<td>21</td>
</tr>
<tr>
<td>Mean Distance From Wall</td>
<td>148.49</td>
<td>800.85</td>
<td>160.17</td>
<td>mm</td>
<td>5.39</td>
<td>1.08</td>
<td>22</td>
</tr>
<tr>
<td>Mean Wall Departure Angle</td>
<td>0.603</td>
<td>N/A</td>
<td>1.09</td>
<td>rad</td>
<td>N/A</td>
<td>1.81</td>
<td>24</td>
</tr>
</tbody>
</table>

¹ For Periplaneta (Butler, 1973).
² Note that the time was scaled by 1.6. This is included in the scaled speed column (which comes from the tracked data.) The actual robot speed in the open arena is given in the un-scaled column, and the real wall speed is calculated by the tracked ratio.
### Wall-Following Results

#### 6.1 Percent of time and distance travelled on walls

The robot’s wall-following algorithm caused it to spend significantly more time along the wall, when compared to the cockroach. These data were measured by scaling the robot data to the cockroach arena, and thus the wall distance was set as the center of the robot’s body within 40 scaled mm. The robot, running the RAMBLER model, averaged 68.7% of its time long the wall and 58% of its distance walked (n = 82), while the cockroach averaged 49.2% of time but only 27% of distance walked (n = 177). The distance data are shown in Figure 25. The robot also spends more time in the corners than the cockroach, with a mean of 11.7% of the trials for the cockroach, and 25.2% of the robot trials. Since the corner time is also counted as wall time, this means the robot was spending an
additional six percent of its time along the wall but away from the corner (37.5% for the cockroach, 43.5% for the robot).

![Graph showing ratio of distance travelled along wall vs total distance walked for cockroaches and robots under different shelter conditions.](image)

Figure 25: Regardless of shelter location, the robot tends to spend 58 percent of its distance traveled along the wall, and the cockroach tends to spend 27 percent. All box sizes are 90% bootstrap confidence on the mean.

The robot’s mean, scaled distance to the wall is 160.17 mm, while the cockroaches’ is 148.5 mm. All shelter locations are shown in Figure 26.
Figure 26: The robot's mean distance to the wall lines up with two of the shelter cases (mid-back and side wall) but tends to be greater than the cockroach control.

For the deterministic model, with a barrier, these values increased even further, with the robot spending 75% of its time on the wall, or 77.6% if the barrier is included. This contrasts with 68.9% in the RAMBLER trials with a barrier.

6.1.2 Wall Departures

The robot is somewhat less likely to depart the wall than the cockroach, as shown in Figure 27. The robot departs at slightly larger angles than the cockroach, at a mean of 1.09 radians versus .6 radians (Figure 28 for the robot).
Figure 27: Robot wall departures rates (left) are consistently lower than cockroach wall departure rates (right).

Figure 28: Distribution of robot wall departure angles.
Values such as the antennal maxed out timer or the PID response terms could be tuned to change these results, and are discussed further in future work, Section 7.2 and Appendix B.

6.2 – Arena Exploration and Coverage

Due to the amount of time spent on the wall, the robot tends to cover a larger portion of the wall during its trials than the cockroach: 76.9% for the robot control versus 52.7% for the cockroach. The cockroach covers 59.2% of the arena versus 41.8% for the robot.

These trends are visible in Figure 29 for all shelter locations. The lower percent of coverage in the cockroach data before it reaches the shelter can be accounted for by the fact that the cockroach reaches the shelter sooner in the mid-back and front corner locations. The deterministic algorithm’s metrics are not significantly different from the RAMBLER algorithm.

Figure 29: Percent coverage of the wall and the arena for the robot and cockroach. All shelter locations are shown. The robot generally covers more of the wall and less of the arena.
6.3 – Shelter Seeking

6.3.1 Camera Algorithm Results

The camera algorithms written to work with RAMBLER had mixed success. The left/right turn algorithm is difficult to evaluate with in-trial data given both the large amount of data and the noise in the heading data. For shelter detection, out of 112 trials analyzed for these metrics with 91 shelter visits, the robot detected the shelter 118 times, or an average of 1.3 times per visit. However, there were also 5 visits in which the robot did not detect the shelter, and 17 false positives. The number of true negatives is very large (on the order of 220000), so computing statistical accuracy is not useful. The statistical precision is 87.4%.

The “go to shelter without pivoting” code was less successful. Out of 26 times that the code was called, the robot reached the actual shelter only 9 times. In 5 of the failures, the robot had visited the shelter 10 seconds prior and thus was traveling straight away from the shelter until it hit a wall; implying the code may need to be suppressed for some time after the robot visits the shelter.

Data on the biasing algorithm (comparing darkness of left versus right) is only available using the macro data on length of time and path to shelter.

6.3.2 Time and Distance Walked to Shelter

When there is not a shelter in the arena, the robot takes roughly the same amount of time to reach the shelter, although the path length is slightly shorter. When a shelter is present, the robot reaches the shelter in a longer scaled time than the cockroach for the mid-back and front corner locations, and also walked a further scaled distance to get there. For the
side wall location the scaled time and distance are roughly the same. These data are compared in Figure 30.

Figure 30: Time to shelter is similar for the cockroach and robot control cases, as is time to shelter in the side wall or EC position. The robot tends to travel a similar amount of total distance as the shelter trials, but only reaches the shelter in a similar distance to the side wall trial. Box sizes are 90% bootstrap confidence.

For the mid-back shelter location, the cockroaches approached from the wall 35% of the time, while the robot approached from the wall 25.2% of the time.

6.4 – Comparison of Barrier Behavior

The deterministic algorithm was designed specifically to mimic a cockroach navigating around a barrier. Without one, it reaches the shelter in a mere 7 scaled seconds by driving directly to the shelter. The algorithm mimics the behavior qualitatively observed by Richards and Porr – departing the wall when the agent sees a free path to the shelter. This strategy is successful for shorter barrier lengths (70 scaled cm, 29 successful out of 31), but like the cockroach trials discussed in Section 3.6, breaks down when the barrier is 80 scaled cm long (11 successful out of 21) (Szczcinski et al. 2012).
The RAMBLER code was tested using a barrier half the length of the arena, and displayed different behaviors from that of the cockroach in similar situations. Cockroaches will often make sharp turns when they lose a wall they are following, while the robot continues on a roughly straight path. These contrasting behaviors can be seen in a track of a cockroach’s path in Figure 31 and the robot’s path in Figure 32. This is because the robot’s wheels were prevented from being driven backward by the software in the trials displayed, limiting the radius of turning. This difference is also present in the deterministic algorithm. An adjustment of the wheel speed values can provide for this turning, but it upsets the stability of the other wall following parameters. The entire wall following behavior would have to be re-tuned in order to properly accommodate cockroach barrier behavior.

Figure 31: An individual trace of cockroach barrier departure behavior.
Figure 32: A robot following the barrier and then hitting the opposite wall. The robot did not make it to the shelter in this trial.
Chapter 7 – Conclusions and Future Work

As highlighted in Chapter 6, the current model of the robot matches the cockroach well in some areas (distance to shelter in control) but not others (time on wall, time to shelter in mid-back). Testing this model on the robot has also drawn attention to some biological behaviors (turns after losing antennal contact). This final section summarizes the similarities and differences and suggests future work to better align the model with the cockroach behavior. It then details the advantages of the robot model for biological experimentation and suggests new biological experiments based on the robot data so far.

7.1 Similarities between cockroach and robot data

Aside from the physical parameters I used to scale the robot model, the following values from chart 1 in Chapter 6 came within 10% of their target values: mean wall speed (7%), mean distance from wall (8 %), control time to shelter location\(^3\) (9%), time to shelter for side wall (9 %), and path length to the side wall location (8%). Percentage of the arena explored before shelter was similar for the side wall location (Figure 25). Total path walked was similar for the shelter trials in the front corner and side wall locations (see Figure 26). While differences ranged from 29% to 215%, these preliminary agreements show that the robot has promise as a test platform for future models.

7.2 Differences between cockroach and robot data, and possible improvement tactics

The largest differences between the cockroach data and the robot data are in stopped time (robot 29% of cockroach), time in corner and percent of distance walked along wall (215%). Stopped time is intentional, as the cockroach’s forward speed drops to zero.

\(^3\) Only mid-back was reported in chart 1, but the other control locations were also similar to the cockroach model.
roughly 27% of the time it is in the arena (Bender et al. 2011), while the robot moves continuously. Since the robot’s forward velocity is zero for its pivots and the cockroach can move in a range of forward and angular speeds, the robot needs additional movement time to make up the difference. In the worst cases (the control), distance walked only varied by 24%, so this programming already made up a large difference.

The corner and wall data are of greater concern. While the increased preference for the walls resulted in a 46% increase in the percentage of wall explored, the exploration of the arena was 29% lower. There is 4.8 times more area classified as ‘in the arena’ than ‘on the wall’, therefore the average cockroach explored 21% more scaled area than the average robot. There are two ways to get the robot spending more time in the arena, which should improve the robot’s exploration performance. Both may be necessary. First, since the robot’s actual wall departure rate was lower than that of the cockroaches, the robot’s programmed wall departure probability could be increased. Secondly, the wall following parameters could be tuned to decrease the amount of time the robot spends getting through corners. Previous tuning by Szczecinski was done qualitatively (2012, personal communication), so the quantitative feedback provided by the overhead camera and the MATLAB analysis code could provide a different tuning outcome. The tunable parameters and their probable effects are described further in Appendix B.

While it is possible that improvements in wall following would improve shelter-seeking in the front corner and mid-back cases, it is more likely that improvements need to be made to the vision algorithms. The blob finding algorithm used in the deterministic model takes time to establish a proper threshold for the shelter. The left-right pixel

---

4 1430 cm² of space was classified as wall space, while 6923 cm² was classified as arena.
summing used by the RAMBLER algorithm is likely to produce false positives, given the number of other dark objects in the arena and the difficulty of detecting the shelter from a distance. The precision of the left-right decision making could be quantitatively analyzed from the arena data. Additionally, this algorithm would need to add a state for “shelter not present” in cases where the difference between the left and right sides was not large enough. The robot could send back still camera images for evaluation of accuracy between tests. A future vision algorithm needs to strike a balance between speed and accuracy.

It is also worth noting again that these robot trials were run with some components of the RAMBLER algorithm, but lacked those adjusting the pivot and wall departure rates based on angle to the shelter. Without these components, RAMBLER still displays significant goal-seeking behavior (Figure 33). Additional insight could be gained by determining the influence of the specific components.
Figure 33: Results of the RAMBLER algorithm run in simulation with only those components implemented on the robot.

7.3 Advantages of Robotic Testing

In addition to the knowledge gained by creating and testing a robotic model of animal behavior, there are also advantages of testing new hypotheses on the robot. For example, while writing this thesis, I was able to run 55 robot trials in four hours, gather the data, and include them in the analysis within 24 hours of coming up with a hypothesis. For the cockroach data, trials are limited by the circadian rhythm, so experiments can only be run at specific times near the start of the cockroaches’ dark cycle (2 hours in this case). With the additional time to clean the arena and let the cockroaches acclimate to it, the circadian rhythm limits the cockroach trials to 14 or 15 trials per day. Some added speed for the robot analysis is currently provided by the robot trials using real-time tracking while the cockroach trials use post processing, but both options are available for both experiments. Additionally, an upper limit on cockroach trials is provided by the replacement rate of the colonies, whereas the robot is only limited by the recharge rate of its batteries.\(^5\)

Experiments on a robot model could refine experimental setups for new hypotheses, saving cockroaches and time.

Additionally, it is possible to obtain sensory data from the robot more easily than the cockroach. The robot can always report sensor values back to the computer, and then these can be evaluated in terms of the robot’s outputs. The robot could also send camera images back to the computer, to obtain agent-eye views of the situation. In cockroaches, it can be difficult to locate a specific neuron, or a probe can lose its position mid-trial.

\(^5\) Roughly 2 hours of testing per hour of charge time, multiple batteries are recommended.
The reliability of data coming from the robot could also be used to refine biological hypotheses and experiments.

### 7.4 Future Work in Biology

In addition to the future work to develop the robot model, the data in this thesis suggest additional biological experiments. As discussed in Section 6.4, the cockroach tends to depart the barrier at a large angle relative to its current path. This suggests a loss of antennal contact may be as strong of a stimulus as the initial antennal contact. Previous papers on wall following, for example Camhi’s paper (1999) only placed walls on the exterior of the arena, and typically discarded data away from the wall. Some insight may be gained by the response of campaniform sensilla, the cockroach’s load sensors. Ridgel et al. (1999) recorded the activity of the campaniform sensilla during forced loading and unloading of the leg and saw that different units responded to loading and unloading. Similar sensory systems may be at work in the antennae. The data involved with this thesis is only enough to distinguish behaviors at an organism level, since the antennae are not visible from the camera’s distance. Additional work could involve both recording the behaviors with a camera that can see the antennae, and neurological recordings. The T-Maze setup in the Ritzmann lab would be a viable option for this work.

Additional work could also include multiple shelters in the arena to provide multiple goals for the cockroach.

### 7.5 Conclusions

In pursuit of a model of cockroach goal-seeking behavior, this thesis has presented biological data, a hardware model of a cockroach, and two software models of cockroach
behavior. The behaviors have balanced two sensory influences (visual goal-seeking with tactile obstacle avoidance), and significantly directed an agent to a goal in simulation. The robotic test platform is now a viable option for testing future biological hypotheses, and has already drawn attention to a new cockroach behavior (turning with loss of antennal contact). I expect the sum of these contributions will ultimately lead to mission-capable robots finding goals more efficiently while exploring unknown terrain.
Appendix A – Additional Biological Data

This Appendix covers some biological data collected on the path to examining goal-seeking behavior, without directly contributing to the final result. This includes the water seeking trials (part 1) and the data for the bias removed by the shroud (part 2).

A-1 Water Deprivation and Seeking Data

Our first attempt at a goal for a cockroach consisted of having the cockroach seek water in a dark arena. The first section of this appendix details the relevant methods and results.

A-1.1 Methods

For our original experiments, we used female cockroaches from the lab’s colonies, and set them aside in individual containers with food, but not water. Cockroaches were weighed after they were removed from the Ritzmann lab’s colonies, and again the day of testing. The results of the water deprivation are discussed in Section A-1.2.

The water-based arena was also square and 3 feet (91.4 cm) to a side. This arena’s walls were 12 inches high. The original arena’s walls were faceted together with brackets and screws, but this left a significant amount of space that the cockroach would explore with its antennae or try to climb up. Thus, for the final trials, the walls were glued together with silicone sealant (similar to the current arena’s walls).

To remove chemical traces in this arena, I originally used ethanol, which required the arena to dry for five minutes. Then the water dishes (one full and one empty) were placed in the arena, and a lid was placed to cover the arena and prevent air currents. The water was then given five minutes to diffuse, and the cockroach was placed in the starting chamber for the same amount of time. The starting chamber gate was then raised, and the
cockroach was given five minutes to explore the arena. Trials were terminated when the cockroach found the water, as the cockroach would begin drinking and cease exploration at that point.

**A-1.2 Water Deprivation Results**

Most cockroaches lost between 4% and 35% of their body weight while being deprived of water before testing. A log fit to weight loss is shown in Figure 34. Additional weight was lost by those females that were carrying eggs and gave birth during the deprivation phase, but these data were excluded from analysis. Out of 160 cockroaches pulled for testing, only 8 (5%) died before their test, at a median of 25 days without water. The longest a cockroach was deprived of water before testing was 49 days, or seven weeks. A paper published soon after I moved to the light arena suggested this lifespan without water may be assisted by discontinuous breathing (Schimpf, Matthews, Wilson, & White, 2009).
Figure 34: Cockroaches lose weight by dehydration over time according to a logarithmic function.

A-1.3 Water Seeking Results

While the ability of cockroaches to survive water deprivation was impressive, their ability to find the water was not. While 13/17 tracked cockroaches drank water at some point during the five minute trials, they did not proceed directly to the water until they were within, on average, 5.2 cm of it (Figure 35). Other cockroaches got closer without drinking, and there was no correlation between time to water and weight lost.
Figure 35: Cockroaches mostly turn away from water (red x,) those that go to drink it start their approach at an average of 52 mm away (black dots).
A-2 Bias in the light arena

The first few months of trials in the light arena were done without the shroud surrounding the arena. The vertical range of vision of *Periplaneta* is 198, and I assume *Blaberus discoidalis* to be similar. Perspective cues above the arena’s 4 in walls could bias the cockroach. Figure A-3 shows a bias in the cockroaches’ left-right position as they travel down the arena, and the shroud fixing this bias in subsequent trials. Shrouded trials were used exclusively in the remainder of the thesis.

Figure 36: The first X value for a given Y value was recorded and averaged. The width of the spaces is a 90% bootstrap confidence interval.
Appendix B – Robot Parameters

This Appendix contains qualitative information about the robot’s code, and hopes to help anyone doing future work on the robot tune the values more efficiently and effectively. 41 parameters contribute to the wall following and the implementation of RAMBLER. 31 values are set in the variable declaration section of the robot’s picoC code, those that are set deeper in the code are indicated. All variables are of type int unless listed otherwise. The current picoC firmware does not support floating point math. Note that these are only the user adjusted parameters; additional variables are required to run the code:

- minSpeed – the minimum speed of the motors. Current implementation is at 5. Values lower than 0 will increase the robot’s ability to turn, but may also cause it to spin along a wall. Could go as low as -100.
- motorLeft, motorRight – the default speed of the motors. Currently set to 70, 70.
- maxSpeed – the maximum motor speed. Cannot go higher than 100; currently set to 100. If for some reason you want to prevent the robot from travelling at its maximum speed, set this lower.
- antennaSlop – the range over 12 bit integer readout over the 5 kOhm potentiometer, over which the antenna is ignored. Currently set to 200.
- collisionSlop - The range the antenna can be deflected without triggering the collision timer if the derivative term goes to 0. Currently set to 400.
- Goal_diff – establishing the goal position for the antennae for PID. Currently set to 500.
- Divisor – the divisor for the PID loop – allows for precision without floating point math. Adjust rate of convergence. Currently set to 1000000.
• P – the proportional component of the PID controller – currently set to 175000.
• I – the integral component if the PID controller – currently set to 1000.
• D – the derivative component of the PID controller – currently set to 500000.
• One antenna timeout – time in ms before a single antenna is judged to be in collision. Currently set to 300.
• Loop_dt – extra delay, if necessary, for the wall following code to ensure an even timestep. Currently set to 12 ms.
• outerLoopTime – sets the frequency of the pivot checking code, and the wall detecting code. Currently set to 50 ms (20 hz).
• shelterTimeIn – time between first shelter detection and the one that causes the robot to pause. Currently set to 200 ms.
• x_res, y_res: resolution values of the camera, for reference. Note that the resolution can be changed before each trial with a telnet command, and these do not set the resolution. The current default is 160x120 (the smallest setting) and the default can be set in the firmware.
• wallMotor – value of motor closer to wall for wall depart code. Currently set to 100.
• freeMotor – Value of motor closer to arena for wall depart code. Currently set to -10.
• leaveTime – not used – replaced with pivotArray
• pivotChance – the chance of a robot pivoting out of 10000. Currently set to 605, called every 50 ms (20 Hz).
• straightTime – replaced with outerlooptime.
- pivotMin, pivotExit – values used in method timeSelector to determine the length in time, and therefore the magnitude, of a pivot. Currently set to 16 ms; depends on the number of elements in the pivot magnitude distributions, sameArray, mismatchArray.
- wallExitTime – similar to pivotExit, except coupled with the pivotArray distribution.
- wallTransition – the probability of exiting the wall every 50 ms, currently set to 104.
- forwardSpeedLeft, forwardSpeedRight – the values to make the robot go straight during the straight segments. These should be re-checked any time a mechanical change Is made to the robot. Currently set to 80, 100.
- turnSpeed – sets the pivot speed of the robot with a positive and negative motor (will affect the values of pivotMin, pivotExit) currently set to 80.
- sameArray – array that sets the distribution for pivots on the same side as the previous pivot. Each bin in the distribution represents one timestep of pivotExit.
- mismatchArray - like sameArray, except for when the pivot is on the opposite side.
- PivotArray - like the last two, but for wall departures. Pairs with wallExitTime.

The following values are set in the code itself. Line numbers, as of 10/31/2011, are given.

- 106: vcam(0) – fixes the robot’s camera to a specific gain, so it doesn’t try to make dark areas brighter. Fully documented, with other options, on surveyor.com.
• 112, 113 – analog(18) and analog(28) call the value of the antenna’s position. 18 and 28 correspond to the pin positions on the analog input board.

• 140, 144 – motors(-100, 10) – calls a specific motor code to turn away from the wall in a collision situation. These can be adjusted to provide a specific radius of curvature for this behavior.

• 146: Delay(300) – the length of time the above motor command is used for

• 157: the collision timer is reset if another collision event (antenna outside collision slop and d=0) is not felt within 1.5*the timer minimum (currently 300 ms). Prevents collision code from executing during normal wall following.

• 197: offWallCounter >=20 – the number of wall following timesteps in which the antenna are ignored before the code reverts to an arena (straight-pivot) state. Corresponds to 240 ms.

• 297: dirThreshold – corresponds to the probability of pivoting in a different direction, differentiated by whether the previous direction agrees with the perceived shelter direction.

• 330, 332, 334, 336: shelter detection sum regions. The inputs to vmean2 are (Upper left X, Upper left Y, lower right X, lower right Y) of a square region to be summed. This function (like the firmware official vmean) outputs to the global Y1, U1, V1. These regions, then define the goal-seeking regions discussed in Chapter 5.

• 342: Delay(1500): the length of the pause, in ms, when the robot detects shelter.

• 362: vmean2 for values of the region that defines the “go directly to shelter” code.
References


