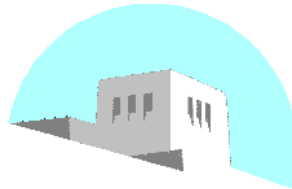


CS547/ECE547

NEURAL NETWORKS



SCHOOL OF ENGINEERING
UNIVERSITY OF NEW MEXICO

La Cucaracha: An Experiment in Embodied Neural Networks

Christopher E Davis

Department of Computer Science
The University of New Mexico
Albuquerque, NM 87131
e-mail: chris2d@cs.unm.edu

Richard Olinger

Department of Electrical and Computer Engineering
The University of New Mexico
Albuquerque, NM 87131
e-mail: rolinger6@comcast.net

Victor Vergara

Department of Electrical and Computer Engineering
The University of New Mexico
Albuquerque, NM 87131
e-mail: vvergara@ece.unm.edu

Final Report

Report Date: December 9, 2004

Abstract

This paper outlines, presents, and discusses the development strategy, design, results and analysis for a neural network implementation of a control system for an artificial being in an environmental simulator. Limited visual sensors and aural sensors are provided by the simulation. The main goal of this project is a well documented, long living robot.

Keywords

neural networks, back propagation, artificial life, simulation, control systems

1 Introduction

Neural networks have held the attention of researchers for over 60 years now because of their simplicity, robustness, and similarity to biological structures present in our own brains. Over the years researchers have applied neural solutions to many fields from medicine to weather forecasting including artificial life form simulation.

This paper covers the development process of an artificial life form driven by neural processes. The life form simulation was conducted within the FlatWorld simulator with the main goal of maximizing life span. The simulator provided sensory information to 32 visual sensors and 2 omni-directional microphones. Using these sensors the problem was to locate food, navigate the environment, and ultimately ; maximize the life span of the artificial life form.

Other researchers have investigated this area in some depth. Some of the most relevant research can be found in the field of “Animats”[12], Rod Brooks robotics lab[4][3], and the behavioral thought experiments of Valentino Braitenberg in “Vehicles”[2].

None of these approaches directly address the task as it was proposed. Animats focus on emulation of behavior and become entirely algorithmic in many cases, Brooks has spent a lot of research time on sublimation which focuses on emulation of behavior in special cases. Braitenberg discusses behavior of small robotic automatons but focuses on expected behavior without implementation or simulation.

What follows in this paper is a concise description of the development method, design, ancillary development tools, analysis of performance, and commentary.

2 Approach

2.1 Development Model

The software process model used in this project was a ‘waterfall’ model[10]. Due to the short timeline of the project it was of upmost importance that the cycle from development to testing to refactoring was quick. The typical software life cycle was shorter than 4 days.

2.2 Ancillary Development Tools

To facilitate a more natural and immediate understanding of the simulation, a 3D visualization tool , “La Cucaracha” (the namesake of our project) was developed for Flatland[6]. It shows a red headed roach immersed in a land full of food. (See Figure 1) The FlatWorld environment dictates the location of the roach and the food. As the roach moves in the simulation the visual representation of the roach scurries around within the visualization. La Cucaracha is completely independent of the simulation and the communication between the two is done by opening a TCP/IP network session. This has the advantage that the visual environment can be driven by a distant computer. The visual environment was of great help to analyze and debug the neural network programming and learning, since it allowed the designers to detect bad movements and behaviors in a more natural way.

Further information on how to run this application can be found in Appendix A.

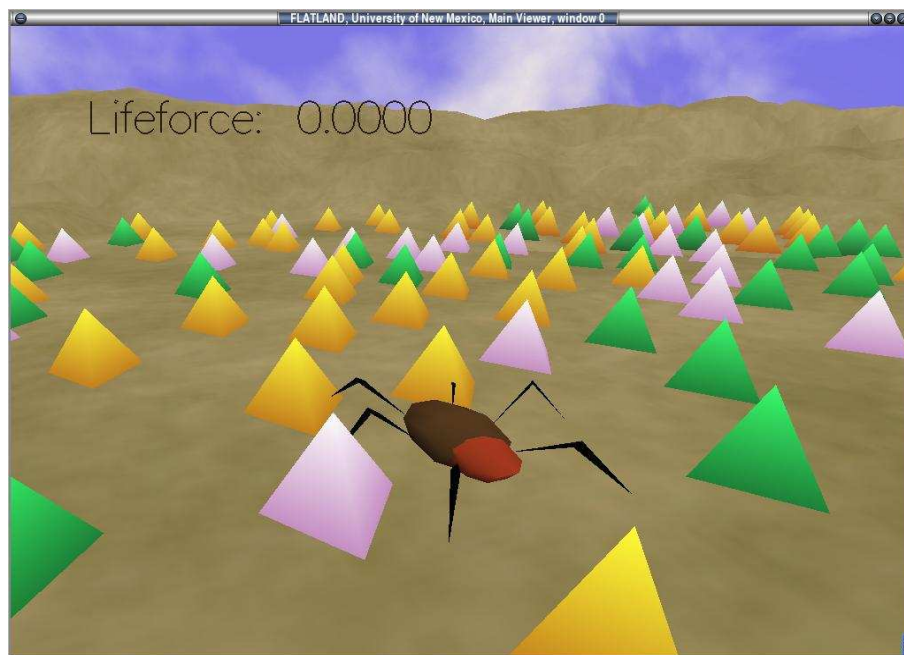


Figure 1: Screen shot of the 3D visualization tool developed in Flatland showing the roach awaiting a connection from a client.

2.3 Sensor Layout

The sensor layout underwent three major changes throughout the evolution of the project. Initially the visual sensors were laid out evenly distributed around the entire circumference of the robot and the microphones were placed at ± 90 degrees (perpendicular to the main axis of the robot). The robot often oscillated due to a lack of direction. Taking the advice of Valentino Braitenberg's book "Vehicles"[2] and Animat papers on artificial vision [12] the visual sensors were moved to the front 180 degrees of the robot. The final sensor layout change involved moving the microphones to the front and back of the robot. This allowed the robot to steer itself towards the noise of objects like Valentino's vehicles moved towards objects[2].

2.4 Neural Design

A main goal of this project was to have a good performing solution to the problem within the time allotted. Given the constraints a simple starting point was a totally random neural solution. Namely the neural network receives all the inputs from the environment simulation, ignores them, then outputs a random activation level. This "random bot" will serve as a comparison model for more advanced models.

The segmented and modular design of the project was tantamount to achieving a successful final design.[10][1] It was clear that the robot controller could not be simultaneously and instantly designed with any hope of success.[5] The problem was decomposed into several smaller problems: movement, classification, and decision to eat. Each of these problems was then addressed by a neural network with the goals of maximizing performance of its task, and as a result the whole robot, while minimizing the overall network-to-network connectivity and communication.[1]

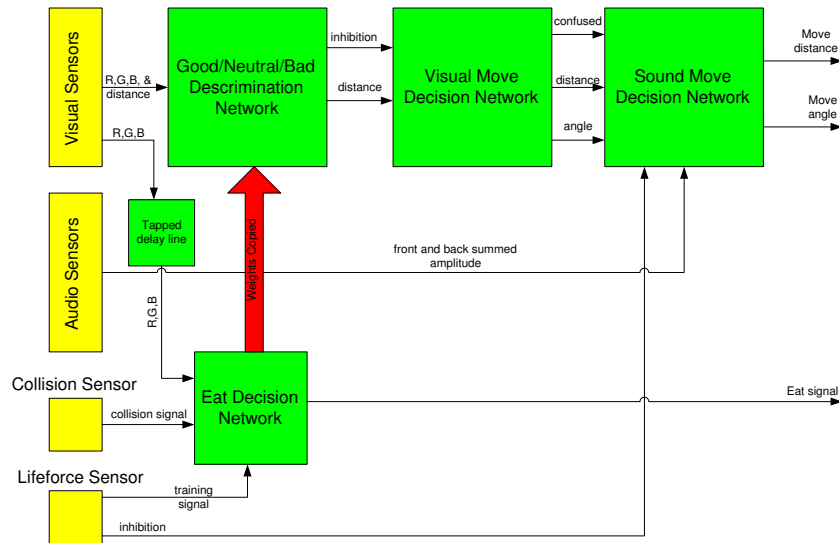


Figure 2: Overview of the entire architecture.

From the “random bot” the next step was to implement a back-propagation network.[9] The initial design uti-

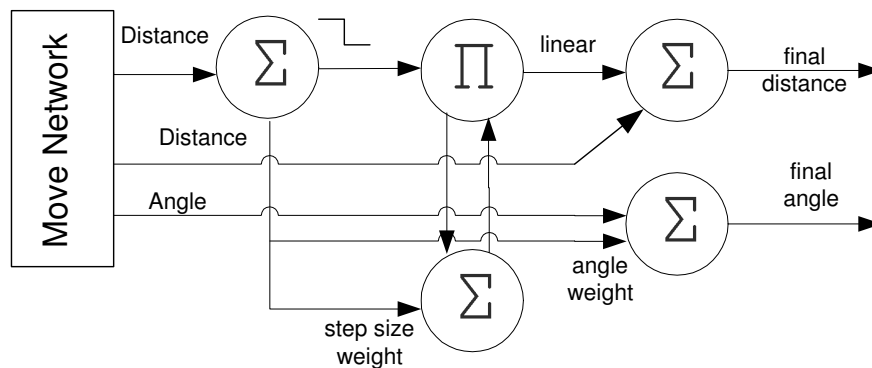


Figure 3: Diagram of a spiral neural component that implements the approach taken in the controller. Note the feedback loop which amplifies while active.

lized only the visual sensors in order to simplify the design process. Visual sensor signals were processed through a parallel collection of back-propagation networks. Each network shares the structure and weights of the “Eat Decision” network. The eat decision network is trained off of the current life force change and the previous set of visual inputs via a tapped delay line structure. A neural implementation of this structure was not implemented because it is trivial to see that a tapped delay line structure can be easily implemented via neurons that simply repeat their input. The movement network was a non-learning Π network that utilized a inhibition signal from the “Good/Neutral/Bad network” as well as a excitation signal. (See Figure 4)

Using these two signals the “Move network” was able to mainly consider the good and neutral objects as potential destinations. The move network then outputs the distance and angle corresponding to the closest good or neutral object to the simulation API.

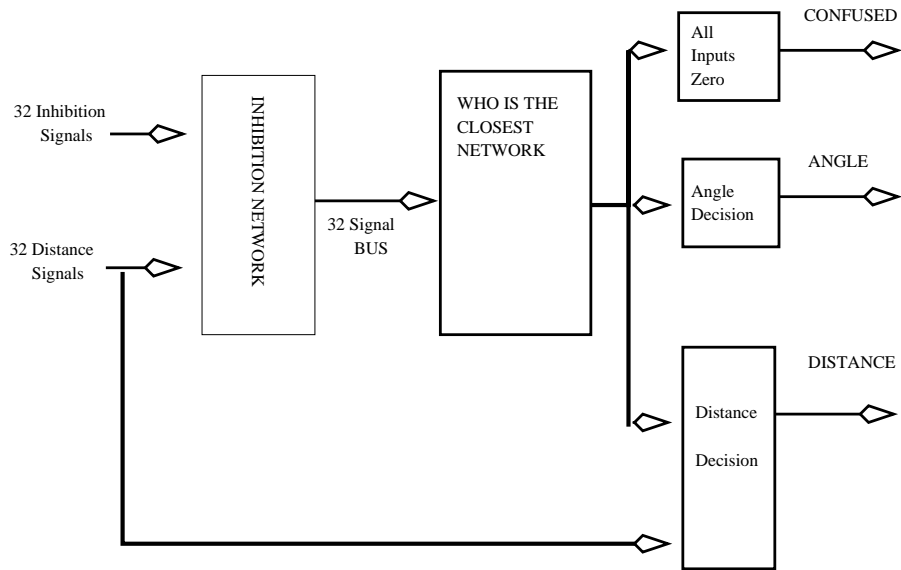


Figure 4: Diagram of the visual move decision network.

Once the visually based robot had been tested and verified the robot was often becoming “stranded” in a circle

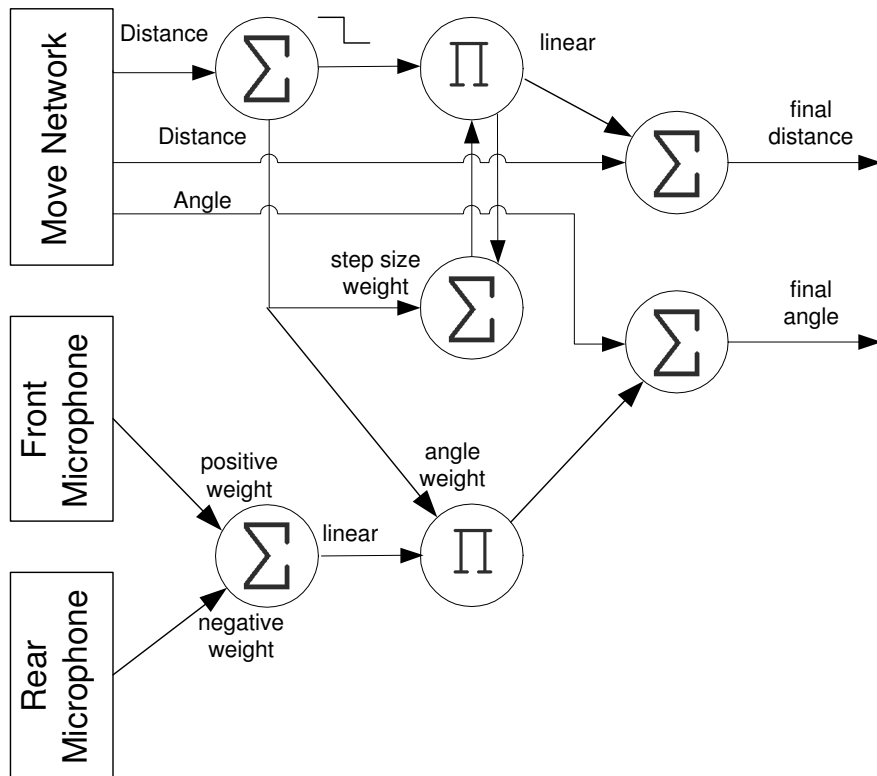


Figure 5: Diagram of an aural processing network that implements the approach taken in the controller. The spiral feedback structure is still present and a directional component is added.

of poison objects spinning in one spot until the life force ran out. To counter this problem another behavior was introduced. When the robot did not sense any visual objects of interest it would turn a pre-set number of degrees and take a step forward. If nothing was sensed again the robot took another turn of the same number of degrees and took an incrementally larger step forward. Although this was implemented in an algorithmic form in the project, it could have been implemented in a neural form utilizing a feedback loop for the step size and a bias node with a pre-fixed weight for the angle signal. (See Figure 3)

After the spiral was introduced, part of the design was re-factored to account for potentially different resource distributions. Audio information was integrated to allow for better performance in the testing environment where the distribution of food is unknown. The audio signal was utilized by the robot only when there was no visual signal to take an action on. When audio navigation was used it helped to guide the robot towards louder areas of the environment in the hopes that it would be able to visually sense another object worth eating. This navigation was accomplished by taking a difference of the front and back microphones total audio amplitude, then using the resulting value as an indicator of whether the robot was facing towards or away from the loudest areas of the environment. The above mentioned spiral was then initiated towards the loudest signal. This was also implemented in algorithmic form although it could have been implemented neurally via a neuron to sum each microphone's amplitudes then a neuron to take the difference of the front and back via a negative weight and summation. (See Figure 5)

The step size and angle size for the spiral were determined by a manual selection of values which were then tested in the environment for efficacy.

With each design several measures were taken to allow for comparison. Each model was run until no further improvement was measured for a large number of epochs. For measuring purposes 360 epochs were run for each model.

The final form of the neural controller and La Cucaracha can be found in Appendix A along with compilation and running instructions.

3 Results

All simulations were performed on poi.hpc.unm.edu a workstation running Linux 2.6.8 on Dual Intel PentiumIII 1GHz processors.

The software was written in C and compiled with gcc using glibc 2.3.2. Results were processed with Matlab 7 version 14. The document was typeset in \LaTeX .

3.1 Random Bot

After running 10 simulations the average life span of the random robot was 228.8 time steps. The longest lived robot lasted 283 time steps, and the shortest lived robot lasted 162 time steps.

There were 6 good eats, 1 bad eat, and 3 neutral eats in the best simulation and 0 good, 2 bad, and 2 neutral in the worst.

Figure 6 shows the life force levels for the longest surviving random robot.

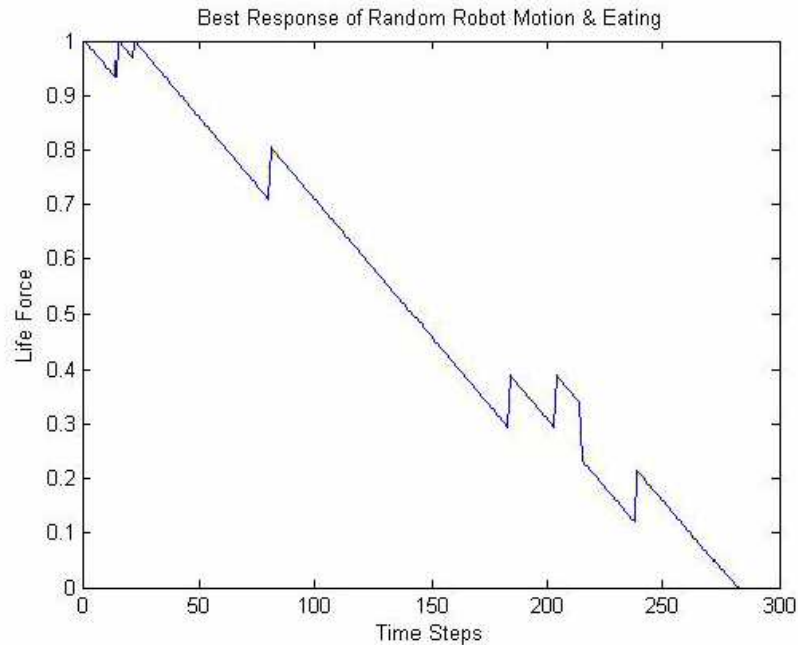


Figure 6: Plot of the longest surviving random bots life force showing slowly decreasing life force over time as well as good and bad reactions to eating

3.2 Back-Propagation with Visual Sensors

After running 7 individual simulations of 360 training epochs each the longest living robot lasted 1878 time steps. It ate 89 good, 1 bad, and 93 neutral pieces of food.

Figure 7 shows a plot of the life force level during the best performing simulation.

3.3 Back-Propagation with Visual & Aural Sensors

After running 7 individual simulations of 360 training epochs each all robots achieved a maximum life span of 2039 time steps. The average training epochs to achieve a life-span of 2039 was 290.4 epochs. The slowest learning robot achieved a life span of 2039 after 360 training epochs. The fastest learner achieved a life span of 2039 time steps after 227 training epochs.

The robots that lived to 2039 time steps ate 99 good, 2 bad, and 98 neutral food objects.

Figure 8 shows a plot of the training epochs for the simulation that reached 2039 time steps the fastest.

Figure 9 shows a plot of the life force within a single training epoch for a robot surviving to 2039 time steps using visual and aural senses.

Figures 10, 11, and 12 are all based off of a single simulation run that was deemed typical. The course of all the simulations for this model followed roughly the same path, with minor offsets in time. It was determined that a typical run in this context was the simulation that achieved a life span of 2039 closest to the average number of epochs for all simulations. The simulation chosen achieved a life span of 2039 after 286 training epochs while the average number of training epochs to 2039 time steps was 290.4 training epochs.

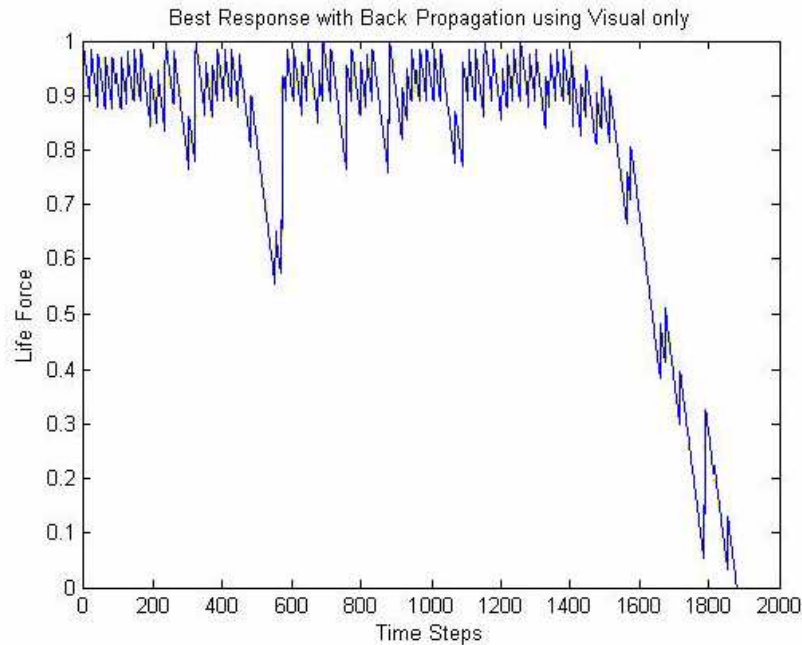


Figure 7: Life force plot from an individual training epoch of the best performing visual navigating robot. This robot uses visual sensors only to navigate and survived to a maximum of 1878 time steps.

Figure 10 shows results obtained by taking the ratio of the number of bad eats over the number of good eats as training progresses through all epochs in a typical simulation.

In Figure 11 a comparison is made between the percentage of moves that use visual sensors vs the percentage of moves that use aural sensors for all training epochs in a typical simulation.

Figure 12 shows a plot of the surviving time steps per training epoch in the typical simulation.

4 Discussion

One strategy for this problem is to have no strategy and simply stay stationary. In this way you guarantee that you will eat no bad, and survive to exactly 200 time steps. From the results above it is apparent that a random strategy is better than no strategy at all as the random robots lived on average 14% longer than a stationary robot. It is apparent, however, that while a random strategy is on average better than no strategy, there were occasions on which the random performance was worse than a stationary strategy, therefore a more advanced strategy is called for.

The visual only robot, which will now be referred to as “the roach”, was much better suited to survive in this environment[5]. The visual roach was capable of learning good from bad, and therefore capable of eating more good than bad, a drastic improvement over the random strategy. It was also capable of food detection and locomotion to facilitate immediate and deterministic collision with the food. Comparing the best performances of the visual roach and the random robot an improvement of over 500% in survivability was measured.

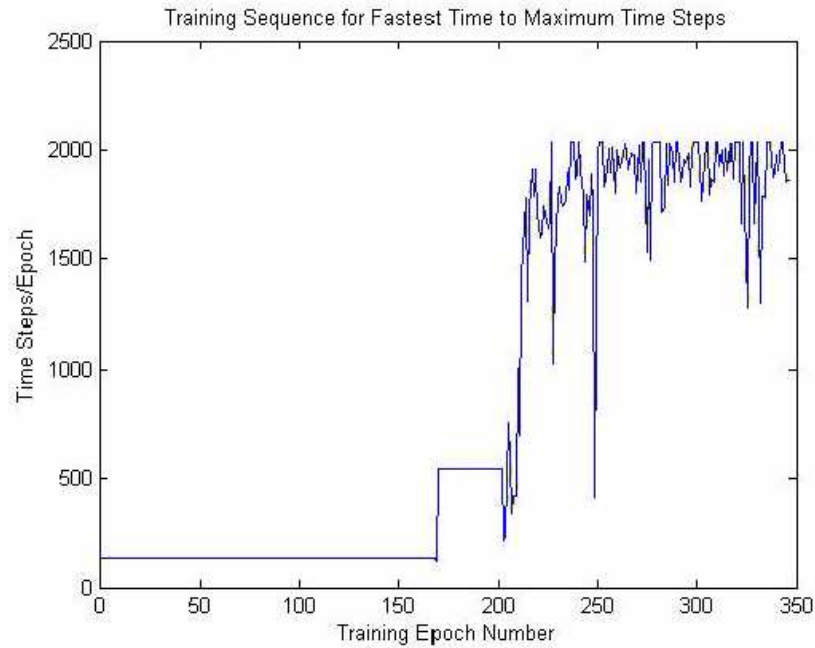


Figure 8: This simulation achieved a maximum life span of 2039 time steps in 227 epochs. This robot was the fastest learning simulation observed.

As noted in the results, this version of the robot ate 89% of the good food. It was determined that the robot was missing food because he was unable to see it due to visual occlusions caused by poison objects.

To overcome this problem aural sensory data was integrated into a network that allowed the roach to navigate towards the loudest areas of the environment. The integration of aural senses also proved useful in keeping the roach on the distribution of food, rather than allowing it to wander off into empty space in the environment. This addition resulted in an 8% increase in survivability over the visual only robot. Although this seems to be a small improvement, the addition of the aural senses allowed the roach to find 82% of the remaining good food items missed by visual only.

Examining Figures 8 and 12 the roach initially performs worse than the stationary robot, dying in 133 time steps for multiple training epochs. Eventually as the roach's networks begin to learn good from bad a dramatic increase in performance can be seen. These innovations are periodic in nature, so it was important to run many training epochs to ensure that maximal performance was achieved.

From Figure 8 you can see that the roach waits to eat until its life force drops to below 0.9. Initially there is a plethora of good food available, so the roach is able to easily survive until it has eaten all the food in its immediate visual vicinity. At this point the life force begins to drop while it is searching for more good food using its aural senses. Once it finds the next cache of good food the life force returns to above 0.9. Eventually the food in this region is depleted and the robot is forced to search again, not finding more food until its life force is below 0.5. At this point comparing Figure 8 with 7 it is noted that this is the point where the two approaches diverge. The visual only roach is unable to locate more food and dies, while the aurally enhanced roach is able to locate several small remaining areas of good food resulting in a longer life span.

Figure 10 shows that initially the roach eats more bad items than good. At the same time that an increase in

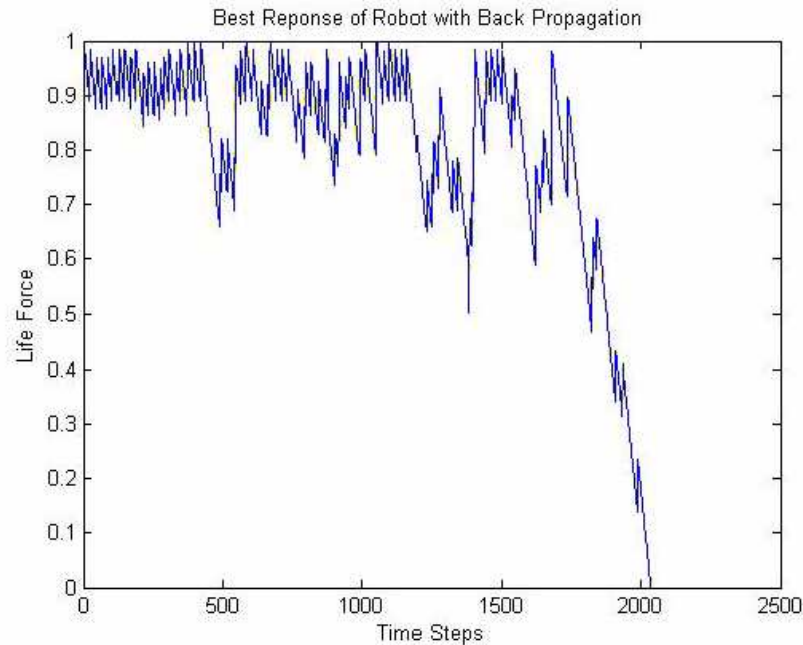


Figure 9: Plot from a single epoch of the life force for a back-propagation network using visual and aural senses. This shows the effects on the robot's life force as it searched for food to survive. This is a representation of the longest surviving robot which survived for 2039 time steps.

survivability is seen in Figure 12 a decrease in the ratio of bad eats over good eats is seen in Figure 10. This is interpreted as the roach starting to differentiate good and bad items, resulting in avoidance of the bad food. Over time a significant decrease in the ratio of bad eats over good eats is seen, signifying the roach has learned to almost completely avoid bad objects.

As the training epochs progress the roach relies on visual and aural sensory to differing degrees. During early training the roach relies heavily (75% of the time) on visual information. At approximately the same time as a significant innovation is measured in survivability in Figure 12 a correlated decrease in usage of the visual senses and an increase in usage of the aural senses is noted in Figure 11. Once the roach has reached the maximal performance range it relies almost equally upon visual and aural senses.

5 Summary and Conclusion

In summary, this project can be deemed a success because all design goals were met. The final performance of the robot is within 8% of the maximal potential life span, it seems robust to changes in data distributions, and the whole process has been well documented in preparation for this paper.

The visualization tool proved to be very beneficial during the design and testing process. Figures 13 and 14 show beginning and ending states of an individual training epoch within the visualization tool. The visual feedback from this tool allowed problems, such as not being able to see any good food, to be quickly diagnosed and ad-

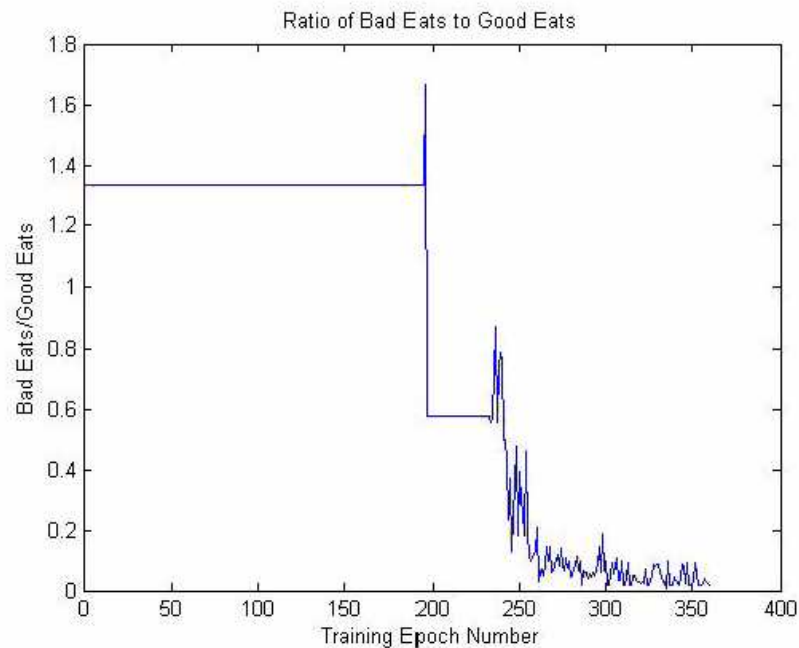


Figure 10: Plot of the ratio of bad eats/good eats for a typical robot, showing over time the robot initially ate more bad than good but improved to the point of eating significantly less bad than good.

dressed.

Although there are more advanced neural approaches available, back-propagation proved to be more than adequate for this environment. Future improvements to this design could include better use of the aural information to classify and locate specific types of objects.

To date the next closest competitor that La Cucaracha faces relies upon a hard-coded spiral that it returns to after visiting potential food items in the environment. Their design is susceptible to changes in the underlying food distributions. For example, if the environment was changed such that there was a “trail of crumbs” that stretched through the environment, La Cucaracha should quickly and easily adapt to this, while the spiral bot will die in a long spiral.

In conclusion, while there are many possible approaches to this problem, this solution has proved to satisfy the main objectives of the project while not becoming overly complex or unwieldy.

6 Acknowledgements

The group would like to acknowledge Dr Thomas Caudell for his instruction, guidance, and consultation on this project. Flatland was developed by Dr Caudell’s research group at the University of New Mexico.

The group would also like to acknowledge the countless cockroaches observed while residing in New Mexico. They have provided a convenient representation and analogy for conversation about this project.

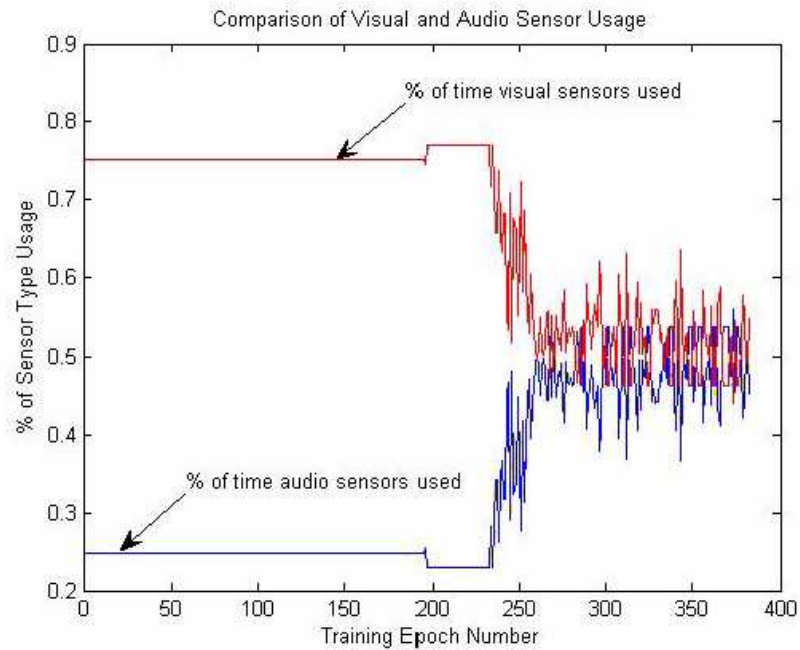


Figure 11: Plot of visual sensory usage vs aural sensory usage by epoch. Initially the robot used visual information the majority of the time and eventually came to use both visual and aural senses about the same amount.

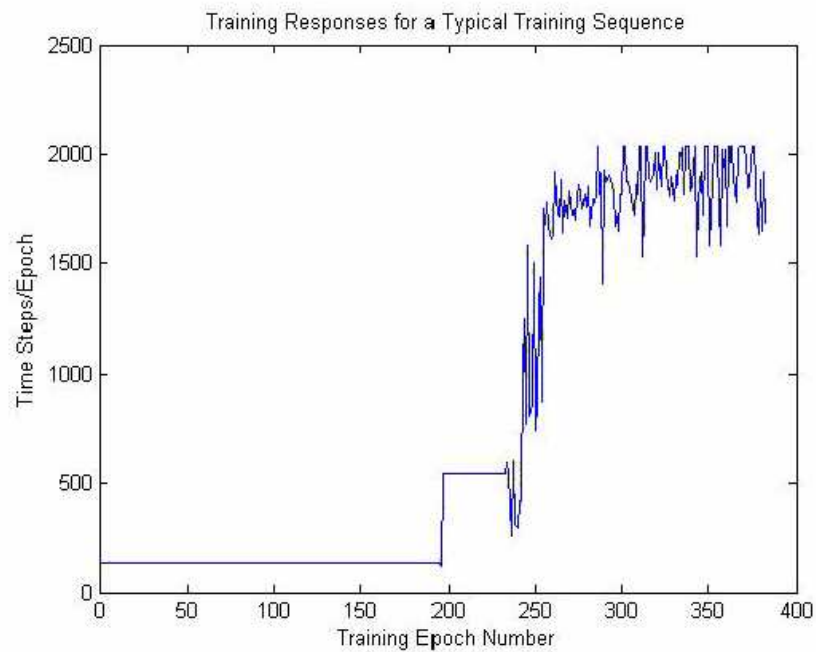


Figure 12: Plot of the simulation for which the training sequence reached 2039 closest to the average of 290. This simulation was deemed a typical representation of all simulations.



Figure 13: Screen shot of La Cucaracha waiting for a client to connect. All food is shown as colored cones, good is gold, bad is green, and neutral is pink. The colors are derived directly from the world description file.

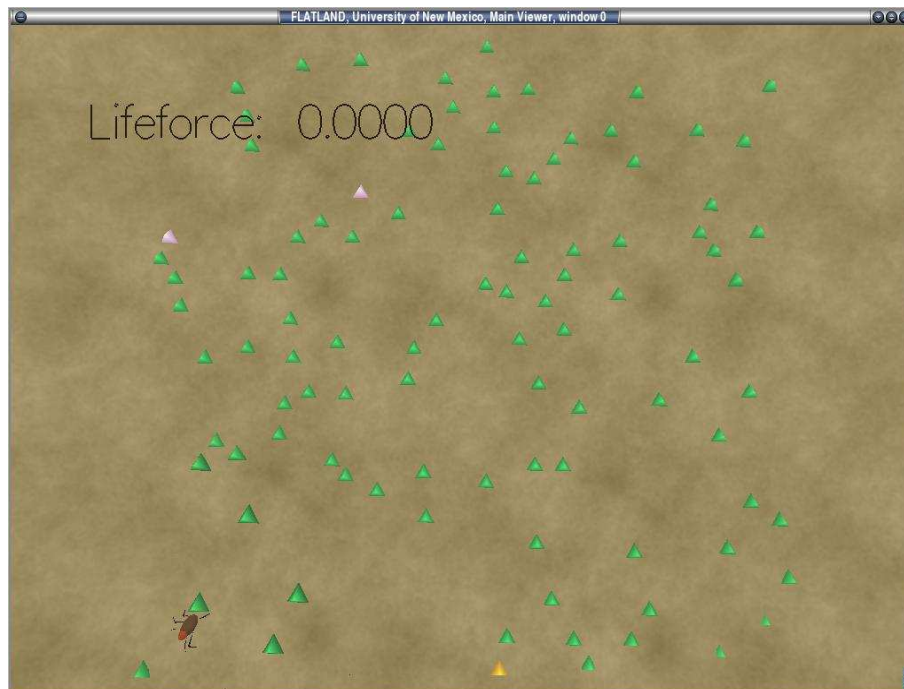


Figure 14: Screen shot of La Cucaracha after completing an epoch showing almost all neutral and good food items eaten and an environment full of poison.

7 Appendices

A Cucaracha Guide

The application is called ECE547. It is launched as any other Flatland Application. The World Description file being used should be indicated in the configuration file (Flatland.config). An example command line instruction is as follows:

```
Flatland -c usr_modules/ece547/Flatland.config.ece547
```

After startup the application enters a wait state (as a server) ready for a user to connect.

In order to provide TCP/IP services to the simulator the following files need to be included during compile and link time:

- SDLnet.c
- SDLnet.h
- SDLnetTCP.c
- SDLnetUDP.c
- SDLnetselect.c
- SDLnetSYS.h
- libSDLmain.a
- libSDLnet.a
- netutil.c

Once these files have been correctly included there are only 3 lines that needs to be added. Here is an example taken from our project.

```
/* This is a modified version of the supplied Main file */
/* This is a test file used to test the FlatWorld API */
/* Created by Jorge Parra and Thomas Caudell */
/* Modified by Chris Davis */
/* Flatland Communication Modifications by Victor Vergara */
#include "H_BraveWorld.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

Final Report

```
... etc ...
int main (void) {
... etc ...
  /* Initialize Network to communicate with Flatland */
  N_InitNet(BraveWorld);

  for (i=0 ; lifeforce >0 ; i++)
  {
    printf("\nSim Cycle #: %d \n",i) ;
    /* Update Flatland */
    N_SendUpdate(BraveWorld);
    ... etc ...
  }
  /* Tell Flatland that the simulation is done */
  N_SendDone();
}
```


References

- [1] Christopher Alexander. *Notes on the Synthesis of Form*. Harvard University Press, 17 edition, 1964.
- [2] Valentino Braitenberg. *Vehicles: Experiments on Synthetic Psychology*. The MIT Press, 1987.
- [3] R. A. Brooks and A. M. Flynn. Fast, cheap and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, pages 478–485, October 1989.
- [4] R. A. Brooks and L.A. Stein. Building brains for bodies. *Autonomous Robots*, 1(1):7–25, 1994.
- [5] Charles Darwin. *The Origin of Species*. Bantam Books, 1859.
- [6] University of New Mexico Dr Thomas P. Caudell. Flatland. Open Source, 1999.
- [7] Daniel Drubach. *The Brain Explained*. Prentice-Hall Inc, 1 edition, 2000.
- [8] Steven J.C. Gaulin and Donald H. McBurney. *Psychology: An Evolutionary Approach*. Prentice Hall, 1 edition, 2001.
- [9] Simon Haykin. *Neural Networks: a Comprehensive Foundation*. Prentice Hall, second edition edition, 1999.
- [10] Ian Sommerville. *Software Engineering*. Addison-Wesley, 6 edition, 2001.
- [11] Lauren Sompayrac. *How the Immune System Works*. Blackwell Publishing, 2 edition, 2003.
- [12] Demetri Terzopoulos and Tamer F. Rabie. Animat vision: Active vision in artificial animals. *Videre: Journal of Computer Vision Research*, 1(1), Fall 1997.