

Using Robotics to Create an Intelligent Agent

Susan Imberman

1. Introduction

According to Russell and Norvig in *Artificial Intelligence: A Modern Approach*, “An agent is anything that can be viewed as perceiving its environment through sensors and acting upon the environment through actuators.” Intelligent agents can be created with varying levels of complexity. Simple reflex agents incorporate a lookup table to determine the agent’s response to its environment. Model based agents are also use a lookup table but are aware of the current state of their environment and can hence differentiate between states that appear similar. Goal based agents have a state that they strive to be in. This allows the agent to have some measure of its “happiness” with respect to achieving a goal. Utility based agents use a utility function to allow it to evaluate between varying degrees of “happiness”. States that are preferred over other states have a higher utility value. Learning agents explore their environment. The agent receives feedback from an outside source, allowing the agent to modify its performance. Thus the agent learns new behaviors that push it toward the desired goal.

Low-cost, low platform robotics has been recognized as an invaluable tool for learning concepts in Artificial Intelligence and other courses in the computer science curricula [Greenwald et. al. 2004; Klassner et. al. 2002; Kumar et. al. 1998; Parsons et. al. 2004].. Students are able to create embodied agents that illustrate many AI concepts. Most platforms today can be programmed using variations of the common languages taught in most Computer Science curricula, such as C, C++, C#, VB.net, and JAVA, to

name a few. Hence, students can implement many of the algorithms taught in most undergraduate AI courses using robotic agents. In the process of embodying agents, students learn basic robotic concepts and techniques as well.

There are many low cost robotic platforms available today. [Dodds et. al. 2006] The LEGO® Handy Board robotic platform has been used extensively in the undergraduate curriculum. [Martin F. 2001, <http://www.handyboard.com>] Our preference for the Handy Board is its ability to support up to seven analog sensors and nine digital sensors without an expansion board. With an expansion board the number of sensors supported nearly doubles. Notwithstanding, this project can use other platforms, such as the LEGO® NXT, with minor modification.

2. Project Overview

The purpose of this project is to build and program intelligent robotic agents. Students, by creating and programming these intelligent agents, will be able to see the differences between different types of agents while experimenting with various AI concepts. Students will create a simple reflex, model based, goal based, and utility agent and deploy these agents in a modified soccer game. Ultimately, students will create a learning agent that uses a neural network to learn path following behavior.

2.1 Project Objectives

The overall goal of this project is to allow students to embody a robotic agent so that it exhibits behavior that would classify it as a lookup agent, a model based agent, a

utility agent, and a learning agent. The other portion of this project implements a neural network into a robotic agent, allowing it to learn path following.

The desired pedagogical objectives for this project are to:

- Enable students to see how AI concepts offer viable solutions to hard problems when incorporated in real world situations.
- Understand that “simpler” agents are not necessarily “worse” agents
- Understand the basic neural network back propagation algorithm.
- Understand the difference between training a neural network and the trained neural network
- Learn basic robotics
- Realize the problems inherent in working "in the real world"

3. Project Description

This project will take place in several phases. The first phase deals with the construction of the robot. Phase 2 has students programming the robot into the simple reflex, model based, goal based, and utility agents. Phase 3 creates the learning agent.

Phase 1 – Robot and Soccer Field Construction

Overall Problem Statement: To design and build a working robot.

Materials

Axles and Extenders

LEGO Beams (2-3 pkgs per robot)

Connectors and Bushings

LEGO Plates (1pkg per robot)

Wheels and Hubs (2 or 4 per robot depending on the architecture)

8 tooth gear wheel (for ~5 robots)

24 and 48 tooth gear (for ~5 robots)

2 Yellow 1 x 11.5 Double Bent Thick

Handy Board

Photo Sensitive Detectors (2)

Touch Sensors (4)

Range Finder (1)

Servo motors (optional)

9 volt DC motors (2)

28 gauge wire (10 feet ~ 4 robots)

heat shrink tubing (10 feet ~ 10 robots)

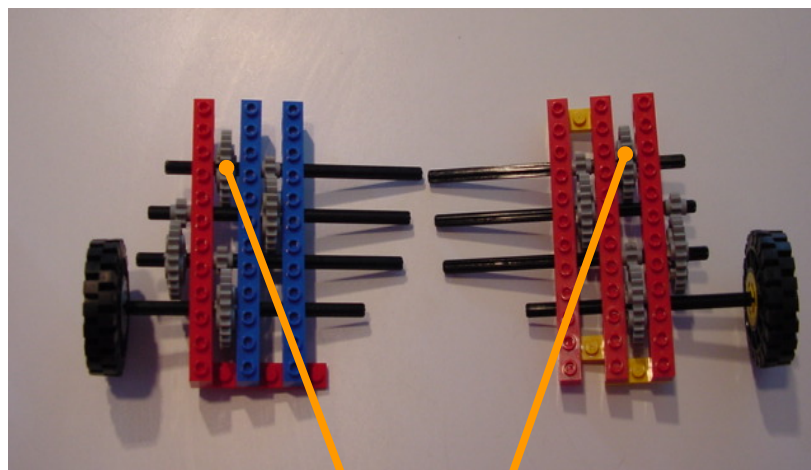
0.1-inch male headers

Constructing the Robot

Step 1 - Construction of the Gear Box. Proper gear box construction is the foundation for a properly working robot. One must constantly check gears during construction to make sure they turn smoothly. If the gear box is too tight, once the weight of the Handy Board is added to the robot, movement may appear very slow or nonexistent. Detailed step-by-step instruction on building a gear box can be found at:

<http://www.cs.csi.cuny.edu/~imberman/ai/Build Gear Box.htm>

Step 2 - Robot construction is finished by connecting the two gear boxes and threading the motors onto the gear indicated in the diagram.



Motors thread here

Make note as to which gear to thread your motors on. The motor gear is placed on the "fastest" moving gear in the gear box when turning the attached tires. If you thread the motor on the wrong gear you will notice the robot won't have enough power to move.

All robots have to have at least two motors and two photo sensors. Two photo sensors need to be located at the "front" of the robot for the learning agent.. Make sure the sensors are placed so that they are about ½ inch apart.

Gears should be constantly checked to see that they move freely the robot is being built. Incorporate the two technic liftarm pieces into the front of the robot as skidders.

Alternate Construction:

1. Start with a basic robot base.
2. Add to the base so as to be able to manipulate the soccer ball (Add LEGOS!!) and detect the field markers (Add sensors!!!)
3. Feel free to customize your robot as you wish. Keep in mind, bigger is not necessarily better!!!!

Step 3 – Test your robot by writing a simple interactive C program. Different brand/types of photo sensors and motors behave differently. You need to determine, for your robot, the power needed to move the robot SLOWLY!! The robot boards have 32K of memory and have a 2 MHz processor. Therefore it takes them time to process information. Write a simple program, using Interactive C that has the robot move forward for 1/2 second, turn right and then move forward, 1/2 second and then move left.

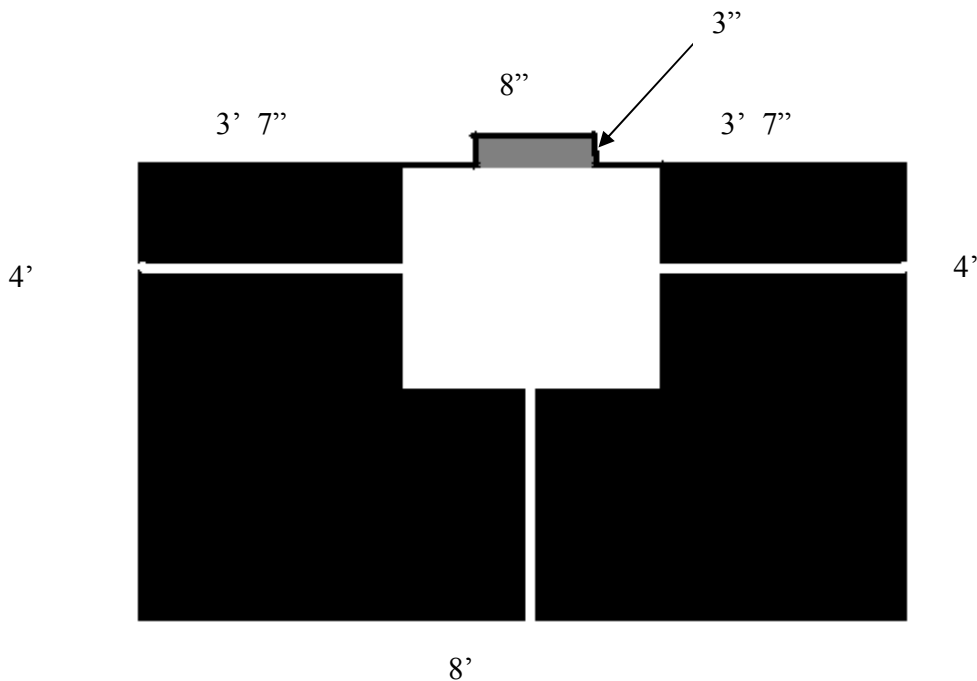
Try to find the minimum motor power to do this. Note how much power you need to make the turns. Use the Interactive C Manual, <http://handyboard.com/software/icmanual/icmain.html>, as a reference for the different C commands you will need. Instructions on how to use Interactive C with your robots can be found at: <http://www.cs.csi.cuny.edu/~imberman/ai/startinginteractiveC.htm>

Deliverable: One working robot

Phase 2 – Creating Soccer Playing Agents

The Soccer Field

Field Construction – Our “soccer field” is modified in that instead of two goal’s there is only one. Basic materials consist of 1 piece of ~4 foot by 8 foot masonite or particle board with “walls” around the edge. Walls can be made from lumber or PVC tubing.



Photosensors measure the reflected light. Hence we used matt black paint and glossy white paint to differentiate areas on the soccer field. Both paints were mixed to color the goal area grey. The white lines give the robots markers for navigation. Flooring can also be constructed of Plexiglas and black electrical tape.

Students can now program their robots to embody the various agent models, simple reflex, model based, goal based, and utility agent. Simple Reflex agents can be programmed using if-else rules where the consequent can be a single statement or function that embodies a behavior. Intrinsic to this agent is that there are no set goals, and previous states are not remembered. A model based agent knows its previous state but still can't make decisions with respect to its goal. A goal based agent can make decisions about a goal but given a set of choices can't discriminate between them. A utility based agent can apply a function to the choices to see which behavior might be better to achieve the goal.

Once programmed, each agent will play a modified form of soccer. As a standalone, the goal of the soccer game is to move the ball (golf ball) into the goal. The amount of goals accomplished in a given time period, say 5 minutes, is recorded. The one-in-one game, pits one agent against another. First one to score a goal is the winner. Students will be asked to draw conclusions as to the relative performance of each of the agents.



Soccer Playing Agent

Phase 2 Deliverable

1. Copies of all agent code.
2. Written paper discussing the students results and observations

Phase 3 - Creating a Learning Agent

Soccer playing robots can use lines in the soccer field to move toward the goal. Can a robot learn this behavior?

Prior to the creating the learning agent students are given a lesson on neural networks after which they are given an assignment to modify a 2 input, 2 hidden node, 1 output neural network to a 2 input, 2 hidden node, 2 output neural network. Below are the assignment instructions given the students.

Write a program that implements the back propagation algorithm discussed in class. Modify the generation5 neural net code to implement a neural network with 2 input nodes, 2 hidden nodes, and 2 output nodes. You will need this code to program your robots. Hand in one program per robot team. The source code and an explanation of back propagation can be found at:

<http://www.generation5.org/content/2000/cbpnet.asp> . The code here will train a neural net to implement a 2 input 1 output function. The source code is downloadable and will run.

Important consideration

- the inputs for the functions learned in this example are either 0's or 1's
- Output of the robot analog sensor function is between 0 - 255
- The result of the neural network is also 0 or 1 (round the decimals)
- Your neural network code needs to output values that will be an input to the IC motor function.
- Motor function inputs range from 0 - 100 since we are only going forward. Therefore you will have to multiply by a scalar amount to get your results to be of the correct order of magnitude.

These are actual training examples used by a former student:

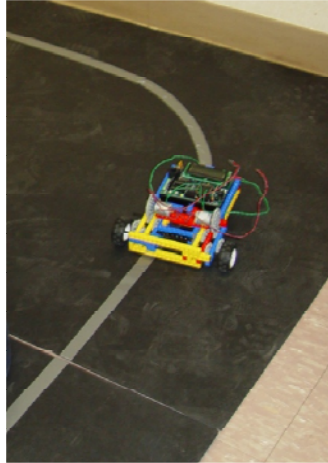
```
// straight
    bp.Train(112,107,.025,.025);
    bp.Train(120,115,.025,.025);
// off to the right
    bp.Train(107,36,.050,.020);
    bp.Train(108,36,.050,.020);
// off to the left
    bp.Train(54,111,.020,.050);
    bp.Train(63,107,.020,.050);
```

You may have to do things differently. The robots are very sensitive to the amount of light available. We also have many different kinds of motors and the respective power requirements will differ. Notwithstanding, these are good enough to see how your neural network works. Also, this student multiplied by a scalar of 550 on the outputs from trained neural network, rather than 1000, before he input them to the motor function. Why did he do this?? Because he needed to slow the robot down more and didn't feel like redoing his training examples!!!

Creating the Road

Because we use photosensors, we have found that the difference in the reflectivity of the background to the actual road black was important. We used silver duct tape on a black

poster board background. The curves in the road were configured to be “gentle curves rather the hairpin. Robots had the most difficulty navigating the curves.



Creating the Learning Agent

Step 1 – Students will write an Interactive C program that will display readings from both photo sensors. These readings are used to create training examples for the learning agent. Given the way the robot is positioned on the road, and the sensor readings from the robot, students estimate the parameters needed for the left and right motor functions to control the rear wheels. Therefore, each training example will consist of two inputs (the sensor readings), and two outputs (the values passed to the right wheel's motor function and the left wheel's motor function).

Step 2 – Students use their modified Generation 5 code, created prior to this phase, to program their robot with a neural network. They then take the trained neural network function, and modify it for use in Interactive C.

Step 3 - Once trained, students then download their trained neural network into another robot with the same sensor type as the robot you trained on. This is to illustrate the robustness of the neural network.

Step 4 - The training portion of the program is executed in the desktop environment, with the Microsoft Visual C++ compiler. Once students have the weight values for the neural network equations, they can incorporate these into the Interactive C neural network and "road test" the robot. Steps 2 and 3 may have to be repeated several times before the robot reasonably follows the road. One important thing to remind students of is to make their robots move SLOW enough so that it has time to take readings from the road and act upon these readings.

Phase 3 Deliverable –

1. Interactive C code for trained neural network
2. Completed Task Check off sheet, a copy appears on the next page.

Team
Members: _____

Robot Name _____

Robot Gender (Male/Female, or Undecided) _____

Instructor Initials	Robot Project
	<p>Task 1 - Build a robot. Use fehmbot as your model. Remember the body of the robot has to be strong enough and big enough to hold the handy board. Demonstrate that you have a working robot. (5 points)</p> <p>Read pages 773 - 785 in your text Answer the following questions. (Only hand in one set of answers per robot team)</p> <ol style="list-style-type: none"> Describe your robot's environment. What type of "real world" tasks might a robot like the one you've built be good for? What types of effectors does your robot have and what are they used for? How many actuators does your robot have? Describe your robot's actuators. What is a nonholonomic robot? What is a holonomic robot? Which one of these describes your robot? Is your robot statically stable or dynamically stable? Why? Our robots have very simplistic sensors. Choose one sensor type, as described in your text, and describe how it could be used to help solve the your robot 's problem task. <p>(1 point)</p>
	<p>Task 2 - - Have the robot use its photo sensors to follow a path. The path is defined by silver tape on a black background. Use the neural network software you modified to teach your robot how to follow a path. (3 points)</p>
	<p>Task 3 - After all tasks and extra credit is completed, dismantle your robot! (Unless your robot makes it into the "Robot Hall of Fame"!) (1 point)</p>
	<p>THERE WILL BE NO EXTRA CREDIT ASSIGNED UNTIL THE NEURAL NETWORK PORTION OF THE PROJECT IS FINISHED.</p>
	<p>Extra Credit - Hard code your robot to follow the silver tape path. (No neural net, lots of if-else statements!!!) Evaluating the performance of an intelligent agent allows one to improve the agent. Did your robot follow the track better when trained with a neural or when it was hard programmed? (1 percentage point added to your overall grade)</p>
	<p>Extra Credit - Use decision tree software to teach your robot how to follow a path. Compare the decision tree solution to the neural net solution. Which performed better? What reasoning can you give for any differences? If they performed equivalently, what reasoning can you give for this?(2 percentage point added to your overall grade)</p>
<p>Total Score:</p>	

References

- Dodds Z. , Greenwald L., Howard A., Tejada S., and Weinberg J., Components, Curriculum, and Community: Robots and Robotics in Undergraduate AI Education, *AI Magazine* 27 (1): Spring 2006, 11-22.
- Greenwald, L., Artz, D. 2004 Teaching Artificial Intelligence with Low-Cost Robots, *AAAI 2004 Spring Symposium Series Report*, Stanford, CA. March 22-24
- Generation5.org, *Back-Propagation: CBPNet*, Available WWW, <http://www.generation5.org/cbpnet.shtml>
- Interactive C, Kiss Institute, Available WWW. <http://www.kipr.org/ic>
- Klassner, F., 2002 A Case Study Of LEGO Mindstorms™ Suitability For Artificial Intelligence And Robotics Courses At The College Level, *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*, February 27-March 03, Cincinnati, Kentucky
- Kumar, D. and Meeden, L., 1998 *A Robot Laboratory For Teaching Artificial Intelligence*, In *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*
- Martin, F., The Handy Board. Available WWW. <http://www.handyboard.com>
- Martin, F., 2001 *Robotic Explorations A Hands-On Introduction to Engineering*, Prentice-Hall, Inc., ISBN 0-13-089568-7.
- Parsons, S., Sklar, E., 2004 Teaching AI using LEGO Mindstorms, *AAAI 2004 Spring Symposium Series Report*, Stanford, CA. March 22-24,.