Using RFID and a Low Cost Robot to Evolve Foraging Behavior

Abe Howell & Roy McGrann
Mechanical Engineering Dept.
State University of NY at Binghamton
Binghamton, NY
+1-607-777-6676
abe@abotics.com
mcgrann@binghamton.edu

Richard Eckert
Computer Science Dept.
State University of NY at Binghamton
Binghamton, NY
+1-607-777-4365
reckert@binghamton.edu

Hiroki Sayama & Eileen Way
Bioengineering & Systems Science Dept.
State University of NY at Binghamton
Binghamton, NY
+1-607-777-2135
sayama@binghamton.edu
way@binghamton.edu

ABSTRACT
The process of developing genetic algorithms, genetic programs or training neural networks is a time consuming task. When the target device is an autonomous mobile robot, this development is often performed using software simulation. Software simulations are a cost effective tool and provide researchers with the ability to test out multiple algorithms quickly and efficiently. However, the end result is that the optimized algorithm(s) must be implemented and tested on an actual robot to evaluate performance in the real world. Significant cost can be associated with this final step. In this paper we propose to leverage Radio Frequency Identification (RFID) and a low-cost RFID capable mobile robot with the intent of creating basic foraging behavior. Additionally, we will present experimental results that demonstrate the effectiveness of using Genetic Programming (GP) and a low-cost RFID capable robot to create foraging behavior.

Categories and Subject Descriptors
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multi-agent systems.*

General Terms
Algorithms, Design, Experimentation.

Keywords
Microcontrollers, radio frequency identification, mobile robot, genetic programming.

1. INTRODUCTION
Foraging behavior is easily simulated when expensive sensors and hardware are modeled in software. However, when it comes to direct implementation on an actual robot, researchers must have access to a robot capable of foraging. A foraging capable robot must have the ability to detect food, manipulate food and perform simple to complex navigation tasks in an unknown or known environment. Traditionally this has been achieved through the use of a suitable camera system, gripper attachment and expensive robot platform. Most commercially available robots that meet these foraging requirements are fairly expensive and cost prohibitive when used in swarms [3,6]. By leveraging RFID we have developed another method for satisfying the foraging requirements. This new system comprises passive RFID tags, a RFID chip and antenna, and a low cost mobile robot. Food is stored virtually on passive RFID tags that can be disseminated throughout a robot's environment. The RFID chip and antenna are integrated with the low cost robot so that the agent can explore its environment while searching for RFID tags. Upon discovering a tag the robot acquires food by simply reading the amount of virtual food that is available.

In this paper we will first describe the robot, its sensors, and its capabilities. After that we will explain how RFID can be leveraged to develop foraging behavior. Finally, foraging experiments we carried out using GP will be discussed.

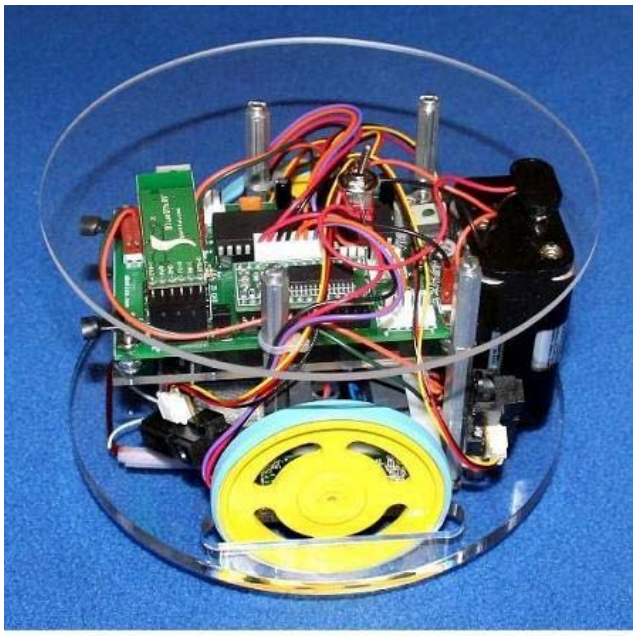

FIGURE 1. BIObot

GECCO'06, July 8–12, 2004, Seattle, WA, USA.
ACM 1-59593-186-4/06/0007.

2. Low Cost Robot

To help solve the issue of foraging requirements we are using a unique low cost robot that was developed under an NSF CCLI grant for a new Bioengineering curriculum at Binghamton University [5]. This robot is appropriately named BIObot and contains several different types of sensors, which allow it to explore and interact with its environment and also execute basic and complex foraging behaviors. The BIObot robot is shown in Figure 1. Currently BIObot is being used in a Bioengineering course, Autonomous Agents, where students are exploring concepts in control theory, fuzzy logic, neural networks, and genetic algorithms. Hands-on interaction with BIObot engages students, makes learning fun, and elevates interest in the concepts under investigation. For most of the semester, students work with BIObot, handheld computers and specially designed software. However, near the end of the course students develop code in Mathematica and must leverage at least two of the covered algorithms to solve a specified problem. The work described in this paper will enable us to develop lab modules for a genetic programming section.

2.1 BIObot's Sensors

BIObot utilizes a total of ten different sensors: five Sharp GP2D120 Infrared Rangers, two cadmium-sulfide light cells, two quadrature wheel encoders, and one RFID chip and antenna. Basic obstacle avoidance and wall following is achieved with the Sharp IR sensors, of which there are three in the front and two in the rear of the agent. The two frontward-facing light sensors support behaviors that are associated with light such as light tracking or avoidance. BIObot is able to move through the environment using three different control methodologies: open loop wheel velocity, closed loop wheel velocity, and position control. Position and closed loop velocity control both utilize the quadrature wheel encoders to implement their respective proportional, integral, and derivative (PID) controllers. This feature helps BIObot to execute semi-precise movements such as rotating 90 degrees or traveling forward 12 inches. Such rotational and translational motions are required capabilities when performing simple or complex navigation. Finally one of the more important sensors, the RFID chip and antenna, allows BIObot to detect, obtain, and transport virtual pieces of food.

2.2 Controlling BIObot

BIObot's onboard controller is able to integrate most micro-controller based Bluetooth® transceivers [1, 2]. A higher-level controller such as a Bluetooth® equipped laptop, desktop, handheld computer or cell phone can wirelessly control BIObot when both devices are paired. The onboard controller runs a specially designed firmware that provides the user with an easy to use set of functions that are accessed via asynchronous serial commands across the Bluetooth Serial Port protocol. Users can command BIObot to move using open loop wheel velocity, closed loop wheel velocity, and position control, to retrieve any of the analog to digital (A/D) sensor readings, to check for in-range RFID tags, to set and get the states of digital pins, to set sensor reflexes or to access a host of other functions that are built into the firmware.

2.3 Programming BIObot

By using simple asynchronous serial commands BIObot opens up the world of programming to virtually any language that supports serial communication. Visual Basic .NET, C# .NET, Java, Python, Mathematica, C, C++ are some of the more predominant software development tools that are compatible with BIObot. However, the end target hardware must also support serial communication and have built-in Bluetooth® or be able to add USB or serial Bluetooth transceivers. Operating system (OS) transparency is also achieved by using simple serial commands

3. Foraging with RFID

With a RFID chip and antenna built-in, BIObot is able to roam through an environment and detect tags as it passes over them. Tag detection is accomplished by continuously polling the RFID chip. We have installed the tags underneath the environment floor so that they will not interfere with the motion of BIObot. Since we are using a wireless Bluetooth connection there is a minimal amount of latency associated with the serial data transmission. Using a serial baud rate of 19,200BPS we realized approximately 120-140ms of latency for the roundtrip transmission of two characters. When polling the RFID chip there is an added delay, which brings the total delay to 250ms on average. Having the robot's micro-controller communicate with the RFID chip via asynchronous serial commands at 19,200BPS adds this extra time delay. Once the RFID chip receives a command it must power an antenna, which in turn supplies energy to the passive in-range tag. Once powered by the antenna, the tag is able to return the requested reading to the higher-level controller. Even with a latency of 250ms we have not observed any degraded performance during our testing of simple foraging behaviors. Future work will investigate the possibility of performing the RFID polling local to BIObot so that the time delay can be minimized.

3.1.1 Virtual Food

Virtual food is stored on tags, which are then disseminated throughout the environment. Each tag can have the same or varying amounts of food. In this way, the amount and location of food sources can be controlled and set up for various experiments. BIObot can interact with the tags and procure virtual food by simply reading an in-range tag. Heuristics and/or other algorithms can be used to determine how much food BIObot is allowed to obtain. Additionally, tags can be used as infinite or finite food supplies. To use them as finite supplies BIObot first reads the tag, determines how much food to take, subtracts this amount from the tag amount, and finally writes back the new depleted amount to the tag. Finite food supplies will eventually expire, unless there is a mechanism for renewal. It should be noted that tags can also be mounted in places other than the floor, so long as the RFID antennae is mounted on BIObot such that it can be positioned parallel to approached tags.

3.1.1.1 Structure of Virtual Food

The Q5-T5555 RFID tags that we are using can store up to a total of 32-bytes of data [10]. Tag data are stored in 4-byte blocks, which results in a total of 8 storage blocks. However, the first two blocks are not to be used because they are set aside to store the configuration settings and a start sequence. The first block stores configuration information with regard to the tag, modulation type, maximum number of blocks to be transmitted, and other settings.

Block two stores a unique 4-byte start sequence, which allows the RFID chip to know where the start of the tag data stream begins. The eighth block can be configured and used as a password for the tag.

In the case of BIObot we are only using the third block in a tag to store virtual food. Since we have a total of 4-bytes available in each block we can use block#2 to store a 32-bit unsigned integer that represents pieces of virtual food. The remaining 5-blocks could be used for storing additional environmental information such as x-y coordinates of the tag, number times the tag has been read or written, or even identification numbers that correspond to robots that have read the tag. Currently there are tags available that comprise 31-storage blocks for a total of 124-bytes of storage space.

3.1.2 Foraging Navigation

Simple and complex navigation can be accomplished using the five infrared range sensors, two light sensors, three motion control methodologies and Sensor Reflexes. Sensor Reflexes allow BIObot to navigate obstacles without waiting to receive instructions from a higher-level controller. Using Sensor Reflexes offloads the task of monitoring sensors so that a higher-level controller can spend time performing other computations and not have to provide continuous instruction to the robot. To use Sensor Reflexes the user must define (2) different thresholds: Analog/Digital (AD) Reflex Level and Light Reflex Level. After setting these levels, the user then prescribes which sensors are to be used for the Sensor Reflexes. Now that the Sensor Reflexes have been set, BIObot can safely move through the environment and when one of the thresholds is exceeded by the appropriate sensor(s), BIObot will halt and await further instruction. A higher-level controller can periodically check to see if BIObot has triggered a sensor reflex and then take the necessary action to correct the situation. For example, suppose we want BIObot to explore its environment and avoid obstacles, but do not want to have the higher-level controller continuously poll the robot for its current sensor readings and prescribe motor speeds. In this scenario we would need to use the three front IR sensors and set the AD Reflex Level to an appropriate value. We can determine the necessary AD Reflex Level by knowing the maximum speed of the robot and also how far away from the obstacle we want the robot to stop. After setting this value we then set the Sensor Reflexes for the front left, front center, and front right IR sensors. BIObot will now drive with prescribed motor speeds until any one of the three front IR sensors exceeds the AD Reflex Level.

3.1.2.1 Simple Navigation

BIObot can utilize simple logic, fuzzy logic, genetic algorithms, genetic programs or neural networks to create simple navigation behaviors. Simple navigation means that BIObot has no a priori knowledge of its environment and does not generate or store knowledge of its environment during exploration. Having BIObot randomly roam through an environment while polling for RFID tags and avoiding obstacles or other robots would be an example of simple navigation.

3.1.2.2 Complex Navigation

Complex navigation takes place when BIObot is provided with information about its environment prior to exploration, leverages simple logic, fuzzy logic, genetic algorithms, genetic programs, or neural networks and/or generates new information and/or modifies its existing knowledge during the exploration phase. In this scenario BIObot can be supplied with a map of the environment that not only includes the location of obstacles, but also that of the food tags and coordinate tags. Coordinate tags consist of the same basic food tag, but instead of containing food they hold coordinate information. BIObot can use the coordinate tags to update its current location status by simply referring to its map of the environment. There are several published papers that reveal how RFID can be used to perform localization on a mobile robot [4,7,9]. By using a map, BIObot can better exploit its environment and over time learn where the best food supplies are located.

4. Genetic Programming Proof of Concept Foraging Experiment

Basic foraging behavior was investigated using BIObot, simple GP based navigation with Sensor Reflexes, eight passive RFID tags, and a small four-foot by eight-foot walled environment. Figure 2 shows the high-level control program, (GP Robot Control), which resides on a Bluetooth® equipped laptop or desktop computer and controls BIObot while evolving simple genetic programs that control the navigation behavior of BIObot.

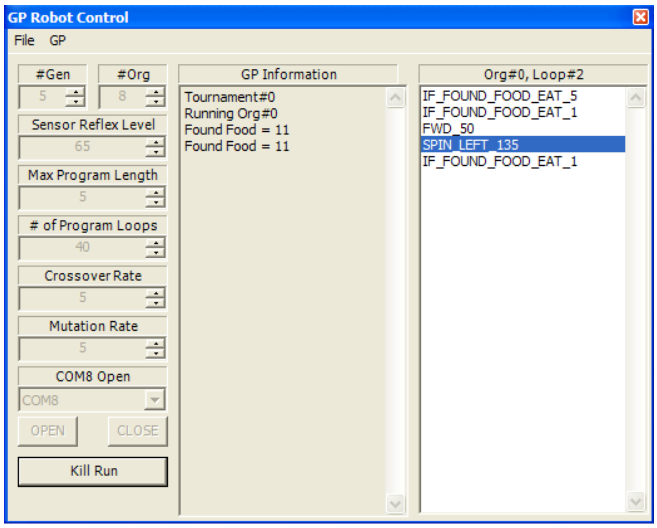

FIGURE 2. GP ROBOT CONTROL PROGRAM

4.1 Initial Setup

Each of the eight RFID tags was randomly loaded with varying amounts of virtual food in the range of 5-20 pieces and then placed under the environment floor. The list of random virtual food values was as follows: 11, 9, 17, 8, 6, 15, 13, and 20. Tags were spaced 16 inches apart in the four-foot direction and 19.2 inches in the eight-foot direction. Figure 3 illustrates the environmental layout for the eight RFID tags.

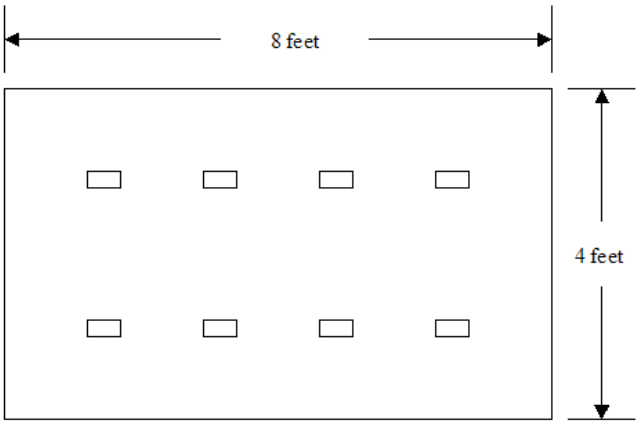

FIGURE 3. RFID LAYOUT

4.2 GP Robot Control Program

The GP Robot Control application allows a user to generate a population of simple linear genetic programs based upon the following predefined functional and terminal sets:

T = {FWD_100, SPIN_LEFT_90, SPIN_RIGHT_90, SPIN_LEFT_135, SPIN_RIGHT_135}

F = {IF_FOUND_FOOD_EAT_1, IF_FOUND_FOOD_EAT_5, IF_FOUND_FOOD_EAT_ALL}

FWD_100 drives BIObot forward for 100 encoder ticks, which is equivalent to 6.4". SPIN_LEFT_90 and SPIN_RIGHT_90 rotate BIObot 90 degrees in the respective direction. Likewise, SPIN_LEFT_135 and SPIN_RIGHT_135 rotate BIObot 135 degrees in the specified direction. The IF_FOUND_FOOD_ functionals allow BIObot to procure food from the RFID tags. If an IF_FOUND_FOOD_ functional is evaluated and food is available then BIObot is able to collect a single piece, five pieces or all the pieces of available food depending upon which functional is being evaluated.

The number of generations, number of organisms, AD Sensor Reflex Level, program length, number of program loops, crossover rate and mutation rate are parameters that the user has control over. Sensor Reflexes ensure that BIObot will not collide with the environment walls. The initial population of organisms is generated randomly with fixed length as specified by the user. Organisms are evaluated live on BIObot in four-organism tournaments. At the end of a tournament, the two organisms with the highest fitness are copied into the two least fit. Next, crossover and mutation are performed on the two least fit based upon crossover and mutation rates. Single point crossover is performed since the genetic programs are linear. Our mutation operator utilizes single point mutation and can replace a functional with a terminal or vice versa. Fitness is determined entirely by the amount of food collected during a specific tournament run.

4.3 Experimental Costs

All major experimental expenditures are displayed in table 1.

Table 1. Experimental Costs

Component Description	Cost
Bluetooth & RFID Equipped BIObot	**$250.00**
USB Bluetooth Adapter for Desktop/Laptop	**$30.00**
Eight Passive RFID Tags	**$40.00**
4 x 8 Foot Environment with Perimeter Walls	**$40.00**
TOTAL	**$360.00**

4.4 Experimental Results

An initial population of eight organisms each with a fixed length of eight instructions was randomly generated. The AD Sensor Reflex Level was set to sixty-five for each of the three frontward facing IR sensors to ensure that BIObot would avoid collisions. Crossover and mutation rates were set to 50% and each program was allowed to run for thirty loops during execution. The small population size was chosen due to the limited amount of functionals and terminals, but also because the genetic programs were limited to a linear structure.

The initial population of eight genetic programs evolved for a total of four and a half hours and only covered two and half generations. We decided to cease evolution when BIObot's battery voltage dropped below 7.0 volts, which is close to the discharge knee of BIObot's rechargeable NiMH cells.

Table 2. Results of Experimental GP Runs

	Pieces of Food Collected
AVERAGE	**20.92**
MIN	**0**
MAX	**104**
STD. DEV.	**31.8**

Results of the experimental GP runs are shown in Table 2. On reviewing the results it can be seen that on average, BIObot was able to collect 20.92 pieces of virtual food. The genetic program with the highest fitness, organism#3, was able to collect 104 pieces of food even though this particular program generates a circular trajectory when executed. The program for organism#3 is listed below.

SPIN_LEFT_90
IF_FOUND_FOOD_EAT_1
SPIN_LEFT_90
SPIN_RIGHT_135
FWD_100
IF_FOUND_FOOD_EAT_ALL
IF_FOUND_FOOD_EAT_1
SPIN_RIGHT_135

We decided to let the evolutionary process run continuously, meaning that each organism is evaluated immediately after the previous organism in a tournament. Using this configuration makes it possible for a circular program to begin execution within close proximity to a food tag if the previous organism was terminated close to a tag. In this situation, the executing program is able to collect relatively large amounts of food in comparison to a program that traverses a larger amount of the environment's total area.

5. Conclusions

In this paper we presented a basic framework for utilizing RFID, a low-cost robot, and genetic programming to create foraging behavior. The low-cost RFID and Bluetooth equipped robot was introduced and discussed in some detail. Furthermore, we provide the favorable results from of our proof of concept experiment with a single BIObot, eight passive RFID tags, a four by eight foot environment, and evolution of a genetic program. Future work will investigate more complex foraging behaviors by leveraging fuzzy logic, neural networks, genetic algorithms, genetic programming, complex navigation, and possibly a multi-agent schema with several BIObots communicating over a wireless network.

6. ACKNOWLEDGMENTS

This project was funded by NSF CCLI EMD, DUE-0442887.

7. REFERENCES

[1] BlueSMiRF, http://www.sparkfun.com

[2] EB500, http://www.a7eng.com

[3] Garcia Robot. http://www.acroname.com

[4] Hahnel, D., Burgard, W., Fox, D., Fishkin, K., Philipose, M. Mapping and Localization with RFID Technology. *IEEE International Conference On Robotics and Automation (ICRA), New Orleans, LA.* April 26- May 1, 2004.

[5] Howell, A., McGrann, R., Way, E., and Woods, R. Autonomous Robots as a Generic Teaching Tool. Submitted for publication to the *36th ASEE/IEEE Frontiers in Education Conference, San Diego,* CA (Oct. 28-31,2006).

[6] Khepera II Robot. http://www.k-team.com

[7] Kulyukin, V., Kutiyanawala, A., Jiang, M., Gharpure, C. Surface-Embedded Passive Radio Frequency Exteroception in Robot Navigation. Technical Report USU-CSATL-2-01-06, Computer Science Assistive Technology Laboratory, Department of Computer Science, Utah State University. January 24, 2006.

[8] Nara, T., Ando, S. Localization of RFID Tags from Measurement of Complex Gradients of Electromagnetic Fields. Graduate School of Information Science and Technology, University of Tokyo.

[9] Seo, D., Won, D., Yang, G., Choi, M., Kwon, S., Park, J. A Probabilistic Approach for Mobile Robot Localization under RFID Tag Infrastructures. *International Conference on Control, Automation and Systems (ICCAS), Gyeong Gi, Korea*. June 2-5, 2005

[10] SONMicro Electronics, http://www.sonmicro.com