A Web-based Platform for Distributed Robotics Research

Gabriel Loewen, Jack O'Quinn, James Weston, and Ashraf Saad*

*Associate Professor and Department Head of Computer Science

Armstrong Atlantic State University, Savannah, GA

Abstract

We present a web-based platform and simulation system for IntelliBrain robots. The IntelliBrain is a small Java programmable robot which has a set of attached sensors. These sensors are utilized by the IntelliBrain in order to navigate and provide feedback about its environment. The platform is developed to take advantage of these sensors in order to provide a rich web-based experience for developing behaviors for the robots. Behaviors are designed to control a robot or swarm of robots within a grid environment. By enabling the service over the web, users are able to load behaviors into the robots and observe behavior, using live streaming video or by using a virtual environment embedded within the web application. In addition, users are able to develop more complex environments using the virtual environment which acts as a behavior simulator. The simulator allows grid space to be larger and provides additional functionality.

Keywords: Robotics, IntelliBrain, Breadth First Search, Particle Swarm Optimization, Java, Applet, Algorithms, Behaviors, Simulation

1 Introduction

This paper presents a distributed system of robots interconnected with a host machine. The purpose of the robots is to operate within a grid space, or arena, according to a defined set of behavior algorithms. The properties of this system allow developers to define various behaviors according to certain specifications in order to visualize actions produced by these behaviors either physically, or in an virtual environment. IB robots have an initial position within the arena and operate by communicating messages to and from a host machine. Messages received by an IB consist of a command which corresponds to the next location that an IB must move to. Messages sent from an IB to the host machine consist of a list of possible locations that an IB may move to which are calculated using its attached sensors. The combination of sending and receiving messages to and from the host machine allow an IBB to operate independently from other robots within the arena because each robot has a separate dedicated connection to the host machine. In addition, the host machine is charged with managing a remote database which holds the immediate data from each individual IB. The database also contains commands being sent from the web application which are then available for the host machine to read and process in order to distribute messages to each IB.

2 Robot

2.1 The IntelliBrain-Bot

There are many robotic kits available in the hobbyist robotics market. When deciding on a robot for our project we looked at price, level of complexity, programming language, kit features, documentation, and product support. Ultimately, We chose the IntelliBrain-Bot (IB) by Ridgesoft [1]. The IB is a relatively inexpensive Java programmable educational robot kit. It supports a variety of sensors and effectors, which can be arranged on the chassis, and the documentation is exceptional, offering API docs, user-guides, tutorials, and example programs.

2.2 Hardware

Each IB is equipped with one Sharp GP2D12 2X infrared range sensor (IRS) and four Fairchild QRB1134 reflective object sensors (ROS). The purpose of the IRS is to determine the proximity of an object in the IB's path. The IRS determines proximity by radiating an infrared beam outward from the emitter, which bounces off a nearby object and is caught by the receiver. When the infrared beam is received, the IRS calculates the angle that the beam entered the receiver. The distance between the emitter and receiver is known, so the distance of the object from the sensor can be triangulated. . A ROS functions like an IRS, but instead of computing distance a ROS determines the reflectivity of a object. Rather than using time as a measure for reflectiveness, the amount of reflected light captured by the receiver determines the sensor output value, which represents degree of reflectivity. In addition to its sensors, each robot contains two Fubata S148 continuous rotation servos and wheels that provides a variable speed driving mechanism for moving the IB forward, backward, and making turns. Communication between the IB and the host machine is provided by an AIRcable [2] mini serial-to-Bluetooth adapter attached to the IB's COM1 serial port, and serves as means for sending and receiving data during program execution and loading programs onto the IB. An LCD display attached to the IB's controller board for displaying runtime information.

3 Experimental Platform

The experimental platform consists of a physical arena which is composed of black vertical and horizontal lines forming a grid over a white background. The large contrast difference between the black and white helps to ensure that no errors happen while detecting surface reflectivity. Robots are placed upon the grid at a predefined location and are initiated by listening for a remote connection with the host computer. Once connection has been established the operator of the host computer may load behaviors and begin experimentation.

3.1 Virtual Arena

The virtual arena is a representation of the physical arena within the applet. This arena is configurable while the physical arena is static. The benefit of using the virtual arena is that it is possible to apply behaviors on a larger set of robots as well as within a larger and more complex arena.

3.1.1 Methods For Arena Customization

The virtual arena is a representation of a grid by means of a set of nodes. Each node has a

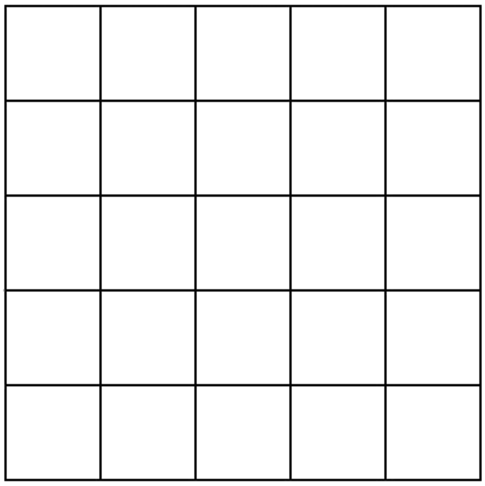

Figure 1: Layout of the physical arena

coordinate such that the set of all nodes construct the grid. In general form,

$$S = \{a_0, a_1, ..., a_n\}$$

Where a_n is a point, (x, y), such that the system constructs a grid of n nodes from S. For instance,

$$(\text{Let } S = \{(0,0), (0,1), (1,0), (1,1)\})$$

Where S is a set of points such that a 2×2 grid is constructed. These nodes are displayed graphically as a grid with connections between each node and its adjacent nodes. In addition to the position of these nodes within the arena each node receives parameters such as color, status and robot occupation. Node status indicates whether the node has been visited or not.

3.2 Navigation

Physical robots use a line following algorithm in which their line sensors are always positioned along the sides of a black line. Sets of ROS sensors are attached to the robot, an inner set and an outer set. The inner set are utilized by the IntelliBrain controller, as stated in section 2.2, to detect variance in reflectivity upon the surface that it is navigating. The reflectivity being tested determines the color of the surface directly beneath the sensor which is interpreted as either white or black. The objective is for the sensors to straddle a black line and to correct the robots movement if reflectivity deviation is detected. For instance, if the left most sensor detects black then the robot will move left until white is detected. Whenever an outer ROS sensor detects black the robot is prompted to stop and exchange data with the host machine. The data exchange allows the robot and the host to communicate actions needed for continuing navigation.

3.3 Host Application

The host application resides on the machine which is responsible for establishing direct communication with the robots via BlueTooth. Also, the host application monitors the database for commands being sent by the applet as well as sending data from the robots to the applet. Whenever data is received from the robots the host application updates the database which is then read by the web application for display. Similarly, whenever the database is updated by the web application the host machine reads the updated commands and transmits them accordingly to the robots. Essentially the host application acts as an intermediate entity in between the database and the robots.

3.4 Web Application

In order for user good interaction we have written the web application as a Java applet to provide a level of abstraction which creates the illusion that operators have direct access to the robots without an intermediary system. It resides on a remote server and allows users to control the robots using a set of behaviors. The user has full control of arena management as well as being able to monitor robot activity remotely via live streaming video as well as a graphical representation of the arena within the applet.

4 Communication System

The distributed nature of the IB robots depend upon a communication system which connects the host machine to the robots as well as to the applet and remote database. Each section of the communication system work in tandem to form a communication loop which allows constant communication between a user, or user defined behavior, and a set of robots.

4.1 Host-Robot Communication

Communication between the robot and the host application is facilitated by means of a serial-to-Bluetooth communications module. The module allows standard serial RS232 3-wire data transfer over a wireless connection. Once we have

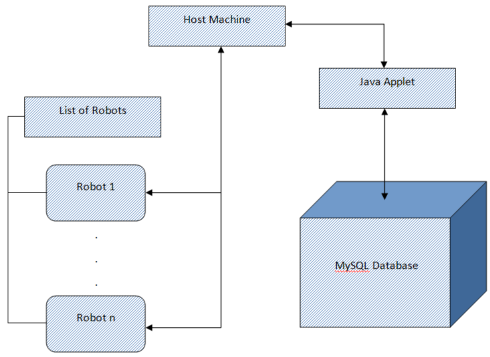

Figure 2: Diagram of communication system

established a wireless link between the IB and the host computer the subsequent connection is utilized by the IB to transmit sensor data to the host application. The host application uses two-way connection to transmit commands and data to the IB. Communications between the host application and the robot work as a series of reads on an open input stream and subsequent writes on an open output stream. Because of hardware limitations neither the host nor the robot may read and write at the same time. During the communications process a sequence of verification methods are used to ensure data has not been lost during the transfer.

4.2 Handshaking Protocol for Host-Robot Communication

The serial stream used for communication between IB's and the host machine is an unreliable connection which can fail due to loss of communication. Handshaking is performed using set of methods provided by RidgeSoft [1]. The RidgeSoft mailbox communication structure provides a reliable means for passing messages over the serial-BlueTooth data stream. The protocol ensures that messages are delivered in the order that they are sent and also maintains data flow.

4.3 Host-Applet Communication

The web application, or applet, resides on a web server. Communication between the host computer and the applet uses a database connection which is accessed by both the host application and the web application for two way communication. The database maintains separate relations for each IB operating within the experimental test-bed. Each of these relations contain data which is specific to an individual IB and are updated as soon as new data becomes available.

This communications model updates data within the shared database which is vital for the applet to function properly, such as robot position and current directional heading. Also, the applet uses the database connection to issue commands to the host application. These commands are then transmitted to the IB using the Host-Robot communications model described in section 4.1.

5 Live Video Streaming

Mounted above the physical arena is a video camera which is connected to the host computer. This video camera captures and streams video of the IB during an experiment. The live video streams as users access the system using the web interface. During the process of an experiment the experimenter may view the streaming video in order to verify that the robot is operating correctly.

6 Behaviors

Behaviors are objects which are designed to navigate a robot within the arena. The host application instantiates behaviors within a separate thread which allows each behavior to operate concurrently. Behaviors may be written to operate a single IB or multiple IB's depending on the design of the behavior. The cardinal directions behavior is an example of a behavior designed for a single IB, and the particle swarm optimization behavior is an example of a multi-robot behavior.

6.1 Behavior Abstraction

Behavior objects are implemented abstractly in order to provide a uniform layout for all behaviors to follow. In this manner every behavior is accessed in the exact same way. Every behavior is given an input, performs a verification on that input to ensure that the input is correct, and produces an output using an algorithm. The methods called between verification and production depend on the type of behavior. Because of the algorithm-centric nature of behaviors the production of the output may come from a generated path or, in the case of particle swarm optimization from analysis of adjacent nodes. For instance, a swarm intelligence algorithm would perform a set of operations which are quite different than a singular robot algorithm but because of the abstraction the output is standardized so that robots can interpret the commands given.

6.2 Graph Searching Algorithms

The representation of the robots grid space is interpreted as a graph. Because of its graph structure, graph searching algorithms allow robots to calculate an optimal path while avoiding obstacles. The implementation of these techniques allow robots to perform complex manoeuvring in order to traverse a node structure, such as a maze. The benefit of using graph searching algorithms with the robots are that given a set of obstacles they will always be able to find a path to the target location while providing feedback to the host machine. For instance, a robot may traverse this path and along the way it could be providing useful data about its environment.

6.2.1 Breadth First Search

Breadth first search is a graph searching algorithm which begins at the root node, which is the starting location for a particular robot, and explores all unexplored neighboring nodes until the target node is found. For an unweighted graph breadth first search is optimal because it always provides a path with the least number of edges between the root and the goal. However, if the graph is weighted breadth first search may not find an optimal path. For instance, if each node represents a location on a map they may each be weighted by distance such that travelling from node a to node b represents a shorter distance than travelling from node a to node c even though node b and node c are both adjacent to node a. In the case that nodes are weighted by distance it is evident that breadth first search may not produce an optimal path.

6.2.2 A*

A* (pronounced *a star*) is a best first search algorithm which produces the shortest path between the root node and the target node. Unlike breadth first search A* does take account for weighted nodes. In this respect A* will produce the most optimal path within a weighted graph. However, the optimality of A* within an unweighted graph is comparable to a breadth first search. (See figure 3).

7 Current Research

Distributed robots have many applications which have not been realized by this system. Current research has an emphasis on swarm robotics and expansion of available behaviors.

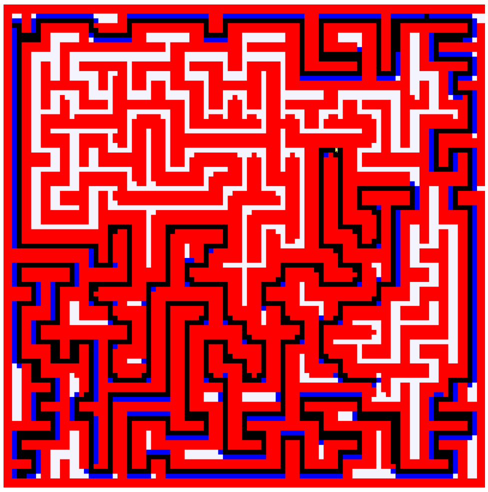

Figure 3: A* Algorithm: A robot is traversing a maze.

7.1 Spatial Pattern Recognition

Integration of the platform technology with learning algorithms creates a robust system to conduct research in spatial learning and spatial pattern recognition. By taking advantage of various sensors attached to the IB, such as the infrared range sensor and the line sensor, spatial patterns can be identified within the arena space where the IB operates. The patterns that are recognized are used as a reference while the IB operates, which allow it to navigate more intelligently. For instance, if the user instructs the IB to find specific intersection points within the grid it may perform an exhaustive search before finding all points. However, an IB which has identified a pattern within the grid will perform a more efficient search by referencing the data gathered during previous attempts.

7.2 Particle Swarm Optimization

Particle swarm optimization, applied to the IB robots, is an algorithm that operates by testing parameters within the arena with a fitness function. This fitness function may be written in many ways in order to solve a number of different tasks. For instance, the fitness function may test soil samples for nitrogen content or could test the air for CO_2 levels. Given the result of the fitness function a local best location, or node, is updated accordingly. After a robot has updated its local best node it announces the current fitness by pushing it into a synchronized queue. This queue is read by all robots independently and a global best node is updated accordingly. Given a local best node and a global best node a velocity is determined for each robot. This velocity determines whether the IB will move in the X or Y direction within the arena. The implementation of a particle swarm optimization algorithm allows multiple IB's to operate together by sharing information and simulates inter-robot communication. Because particle swarm optimization relies on a large set of robots working together on a task results are produced faster than with a singular robot behavior and it is more efficient. (See figure 4).

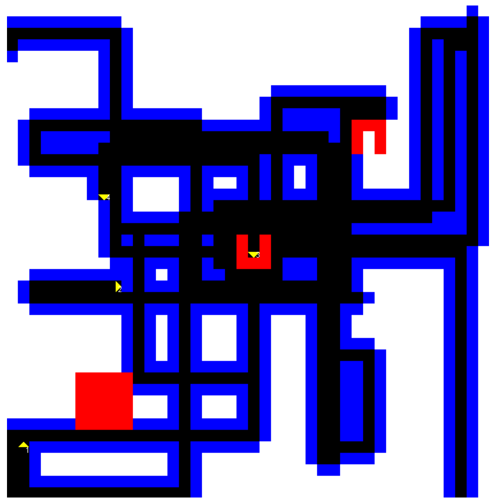

Figure 4: Particle Swarm Optimization: Robots are seen travelling through their gridspace while searching for their target location. Blue represents locations possible for travel, black represents locations already travelled, and red represents obstacles.

8 Future Research

Goals of the research are to produce a system capable operating multiple robots within any environment using a large set of behaviors. Behavior research is aimed at the development and implementation of learning algorithms which form the basis of higher cognitive functions such as perception and intuition. We are working towards the development of an outdoor robot which has onboard GPS navigation as well as a variety of sensors to aid environment exploration and spatial pattern recognition.

8.1 Learning Algorithms

The manner in which the IB's environment is constructed allows for a reinforcement based learning algorithm. Each node in the arena can be constructed as goal nodes in which the robot receives credit for traversal. Over time the robot accumulates data about its environment and patterns emerge which help the robot to find goal nodes in a more efficient manner. Reinforcement learning requires a large number of trials in order for the system to produce accurate predictions about the data set.

Write something about possible classification methods and genetic programming and/or fuzzy reasoning techniques

8.2 Dead Reckoning

As patterns within the arena are being recognized the IB will become increasingly efficient in locating target points within the arena grid. Given that the IB has generated a map of its operating space, research will be conducted to determine the efficiency of an IB in returning to its starting location. This behavior can be observed in populations of bees. After travelling long distances bees are able to accurately and efficiently return to their hive. The postulate is that an IB which has mapped its operating space will be able to efficiently return to its starting location after travelling a long distance. For instance, an IB which has not fully mapped its surroundings might come into contact with obstacles within the arena which prevent it from travelling. In this situation it might attempt to move around the obstacle in order to return to its origin. However, an IB which has gathered data about its operating space will be able to more efficiently determine a path to its starting location without any unnecessary contact with obstacles.

8.3 GPS Enabled Navigation

The current method of navigation is limited by the size of the arena in which the IB's operate. GPS enabled navigation would allow an IB to navigate in an outdoor setting without the need for line sensors. Also, by using a GPS enabled navigation system in combination with an algorithm, such as particle swarm optimization, it would be possible for a swarm of robots to perform a defined task. For instance, a swarm of robots could perform the task of finding a burning building with the aid of a smoke detection sensor.

9 Education

Using the IntelliBrain robots along with the platform introduced in this paper it is very conceivable to teach both undergraduate and graduate courses. For instance, students would benefit from using the platform while taking a course on algorithms because of its algorithm-centric nature. Also, students taking an introductory programming course could also benefit by introducing the notion of programming embedded systems. Undergraduate computer science students who might not have a clear understanding about their possible career options might see robotics and embedded programming as an exciting option for further study.

10 Conclusion

The platform introduced in this paper builds upon the notion of distributed robotics taking advantage of algorithm-centric behaviors. Properties of this system allow developers to take advantage of sensors attached to the robots in order to implement algorithms and observe the robots behavior. A simulated environment within the web application allows developers to expand upon the robots grid space in order to create a larger and more complex environment for the robots. The implications of this system are that a developer can implement any algorithm designed to work within a grid, or graph, structure and provide a visualization using the IntelliBrain robots. The behavior functionality provides a rich educational experience as well by allowing students to study broad computer science topics such as algorithms, data structures, and embedded programming. The combination of research and educational implications make this a unique and beneficial platform which bridges the gap between computer science education and research.

11 Acknowledgements

The authors would like to acknowledge contributions to this research by Dr. Bradley Sturz and Marcel Manning of Armstrong Atlantic State University.

References

[1] RidgeSoft, LLC. Educational java robotics. http://www.ridgesoft.com/.

[2] Wireless Cables Inc. Bluetooth, intelligent, sensor interfaces with range. http://www.aircable.net/.