An Implementation of Robot Formations using Local Interactions

Ross Mead*, Jerry B. Weinberg*, and Jeffrey R. Croxell[†]

Southern Illinois University Edwardsville *Department of Computer Science [†]Department of Electrical and Computer Engineering Edwardsville, IL 62026-1656 qbitai@gmail.com, jweinbe@siue.edu, jcroxel@siue.edu

Abstract

Coordinating a group of robots to work in formation has been suggested for a number of tasks, such as urban searchand-rescue, traffic control, and harvesting solar energy. Algorithms for controlling robot formations have been inspired by biological and organizational systems. In our approach to robot formation control, each robot is treated like a cell in a cellular automaton, where local interactions between robots result in a global organization. The algorithm has been demonstrated in simulation. In this paper, we present a physical implementation.

Introduction

Robots organizing and working in formation has been suggested for a number of tasks, such as systematic searchand-rescue (Tejada, et al 2003), automated traffic cones for road construction (Farritor & Goddard 2004), and construction of a large orbiting solar reflector for harvesting solar energy (Bekey, et al 2000). Work on formations has been inspired by biological and organizational systems, such as the flying patterns of geese or marching bands (Fredslund & Mataric 2002, Balch & Arkin 1998).

Our approach is to treat the formation as a type of cellular automaton, where each robotic unit is a cell (Mead & Weinberg 2006). The robot's behavior is governed by a set of rules for changing its state with respect to its neighbors. By selecting one of the robots as an "initiator", human intervention would change its state, which would propagate to its neighbors, instigating a chain reaction.

A desired formation is defined as a geometric description, which is sent to some robot *i*, designated as the *seed*, or "initiator". The formation definition is communicated to neighboring robots; relationships are determined by calculating a vector from *c*, the formation-relative position (x_i, y_i) of the robot, and the intersection of a function *F*, which defines the formation, and a circle centered at *c* with radius, *r*, where *r* is the distance to maintain between neighbors in the formation [Figure 1].

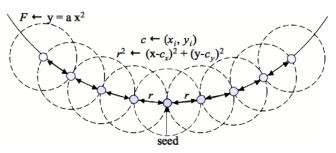


Figure 1: Robots calculate neighborhood relationships.

Once a neighborhood is established, relationships and states are communicated locally within that neighborhood. Using sensor readings, robots then maintain the calculated relationships with their neighbors. Despite only local communication, these relationships result in the overall organization of the desired global structure. It follows that a movement command sent to a single robot would cause a chain reaction in neighboring robots, which then change states accordingly, resulting in a global transformation.

The formation algorithm was initially implemented in simulation (Mead & Weinberg 2006). The simulator demonstrated the scalability and generality of the system, with over 3,000 agents conforming to a variety of different formations. This current work is a demonstration of the feasibility of the approach on a physical platform.

Robot Platform & Implementation

We have developed a platform to evaluate the algorithm in the physical world [Figure 2]. Each robot is built upon a Scooterbot II base (<u>www.budgetrobotics.com</u>). The Scooterbot is 7 inches in diameter, and is crafted from expanded PVC, making it durable and light. Two modified servo motors are employed for differential steering.

The formation control algorithm is implemented in Interactive C (<u>www.kipr.org/ic</u>) and runs on an XBCv2 microcontroller (<u>www.botball.org</u>). The XBC utilizes back-EMF PID for accurate motor control. It also features a camera, capable of multi-color, multi-blob simultaneous tracking. We mounted the camera to a servo system that provides 360° of rotation.

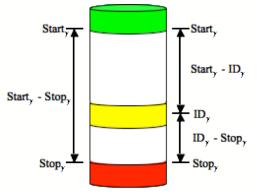
Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Figure 2: The robot platform.

Neighbor Localization

For a robot to determine its state, it must be able to identify and track its neighbors. This is easier said than done, as each unit looks identical. We alleviated these problems by utilizing a colored bar-coding system [Figure 3]. Each robot features a three-color column; the unique vertical location of the ID color bar (in relation to the start and stop color bars) is proportional to the identification number of the robot. Similarly, the perceived distance between the start and stop color bars of a robot is proportional to the actual distance to that robot.



 $Robot_{10} = ID_{exer} * (Start_y - ID_y) / (Start_y - Stop_y)$ Figure 3: Colored barcode used for neighbor localization.

Communication

A radio communication module is needed to share state information within a robot's neighborhood. Our first logical choice was Bluetooth; however, we quickly found this option limiting. Instead, we chose the XBee (ZigBee/IEEE 802.15.4 compliant; <u>www.maxstream.net</u>) for its rich feature set; the module does not require a host/slave configuration, allowing for mesh networking and packet rerouting. The XBee also scales well for large applications, using 16-bit addressing to provide for over 65,000 nodes. The low-power model offers a range of 100 meters, which is equivalent to class 2 Bluetooth. The chip communicates using a TTL-level UART, while the XBC uses RS-232 levels. To overcome level translation and to interface between the XBee and the XBC, we designed our own PCB. We utilized all surface mount parts, with the exception of a 9-pin D-Sub male plug. The result is a board smaller than the XBee itself that directly plugs into the XBC's 9-pin serial port.

Evaluation

The formation control algorithm is currently being tested on three of these robots; however, nine additional robots are in production. When all twelve robots are working in formation, a series of experiments will be conducted and evaluated based on the criteria discussed in Fredslund & Mataric (2002) for reasons of comparison and analysis.

Future Work

If the robots are not initially put in a formation, then a neighborhood must be established dynamically. This will be accomplished by implementing a market-based auctioning method, where a robot is chosen to be a neighbor based on its distance to the desired relative location in the formation description.

References

Balch, T. & Arkin R. 1998. "Behavior-based Formation Control for Multi-robot Teams" IEEE Transactions on Robotics and Automation, 14(6), pp. 926-939.

Bekey G., Bekey I., Criswell D., Friedman G., Greenwood D., Miller D., & Will P. 2000. "Final Report of the NSF-NASA Workshop on Autonomous Construction and Manufacturing for Space Electrical Power Systems", 4-7 April, Arlington, Virginia.

Farritor, S.M. & Goddard, S. 2004. "Intelligent Highway Safety Markers", IEEE Intelligent Systems, 19(6), pp. 8-11.

Fredslund, J. & Mataric, M.J. 2002. "Robots in Formation Using Local Information", The 7th International Conference on Intelligent Autonomous Systems, Marina del Rey, California.

Mead, R. & Weinberg, J.B. 2006. "Algorithms for Control and Interaction of Large Formations of Robots", The 21st Conference on Artificial Intelligence, Boston, Massachusetts, pp. 1891-1892.

Tejada S., Cristina A., Goodwyne P., Normand E., O'Hara R., & Tarapore S. 2003. "Virtual Synergy: A Human-Robot Interface for Urban Search and Rescue". In the Proceedings of the AAAI 2003 Robot Competition, Acapulco, Mexico.