Developing a Low-Cost Robot Colony

Felix Duvallet, James Kong, Eugene Marinelli, Kevin Woo,
Austin Buchan, Brian Coltin, Christopher Mar, Bradford Neuman

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{felixd, jyk, euge, kwoo, abuchan, bcoltin, cmar, bneuman} @ cmu.edu

Abstract

Taking inspiration from nature, we have developed a colony of small, low-cost robots. We have created a robotic base which is inexpensive and utilizes simple sensors, yet has the capabilities required to form a colony. To overcome computational limitations, we have developed custom sensors and algorithms that enable the robots to communicate, localize relative to one another, and sense the environment around them. Using these noisy sensors and simple local rules, the colony as a whole is able to develop more complex global behaviors, similar to the emergent intelligence of colonies in nature. We present our work developing an autonomous robot colony and algorithms for efficient communication, localization, and robot behaviors. We also highlight recent developments that enable our Colony to recharge autonomously.

Introduction

A colony of robots has many advantages over a single robot. Multiple agents can often accomplish tasks better, faster, and more robustly than a single agent can. Groups of cooperating robots have proven to be successful at many tasks, including those that would be too complex for a single robot to complete. Multi-robot systems have been applied to such diverse tasks as complex structure assembly (Heger and Singh 2006), large-object manipulation (Trebi-Ollennu et al. 2002, Sugar and Kumar 2002), distributed localization and mapping (Fox et al. 2000), sensor networks (Hundwork et al. 2002), multi-robot coverage (Cortes et al. 2002), and target tracking (Hundwork et al. 2002). Furthermore, whereas a single complex robot can be crippled easily by damage to a single critical component, the abilities of a colony can degrade gracefully even if individual agents are disabled, even in hostile environments. Some of the most successful organisms in nature survive by working in groups.

Many colonies of robots have successfully demonstrated cooperative actions, most notably in (McLurkin 2004) and (Howard et al. 2006). But while much of this existing research has centered on highly-specialized, expensive robots, we have aimed to create a colony that consists entirely of simple, small, inexpensive robots.

Our research seeks to develop a robot colony with three primary goals:

- low-cost robots
- homogeneous organization
- distributed system

We will continue by presenting an overview of the robot platform we have developed, including its sensors and the infrastructure which allows the Colony to communicate and localize. We then introduce several intelligent distributed behaviors. We highlight recent work on making Colony robots recharge autonomously, which will enable the Colony to execute tasks autonomously for longer durations, and we discuss ColoNet, an interface between the Colony and the Internet. We conclude with some remarks about the project's future direction.

Our Robot Colony

Our Colony consists of small, low-cost robots that are homogeneous in nature. Each possesses the necessary sensory, computational, and communication capabilities to interact with the world as well as other robots.

Robots

Colony robots are small, oval-shaped robots approximately 16 cm long and 14 cm wide, shown in Figure 1. They stand about 8 cm tall on two wheels and a caster, and each robot is powered by a small 6V NiMH battery. Each robot costs around $350. The robots use low-cost microcontroller boards based on an Atmel ATMega128 microprocessor custom-designed in partnership with Botrics LLC. In addition to driving the two DC motors, the board interfaces with the various sensors and user I/O devices located on the robots. These include two buttons, a potentiometer, two RGB LEDs, a piezo buzzer, two encoders, and five IR rangefinders. A further four analog and six digital ports are available for custom homebrew sensors, and an LCD module is supported for displaying text directly on the robot. The robot can communicate via USB, I^2C, and SPI ports, as well as through an integrated XBee wireless module. Mounted on each robot is our custom localization sensor, the Bearing and Orientation Module.

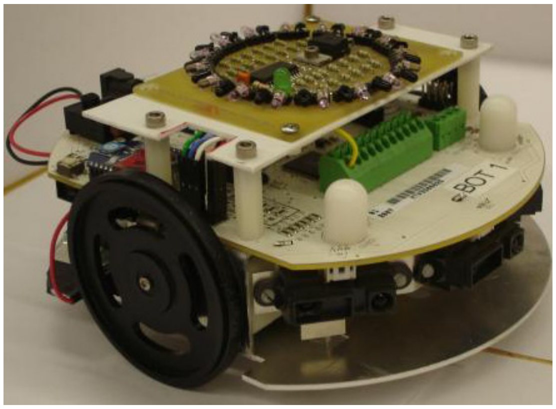

Figure 1: *A single Colony robot*

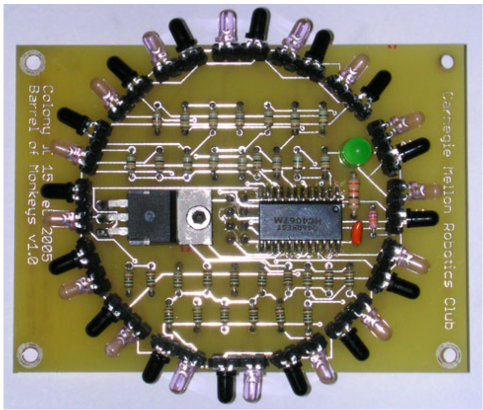

Figure 2: *Bearing and Orientation Module (BOM)*

Bearing and Orientation Module. The Bearing and Orientation Module (BOM) is a custom-designed sensor developed in our previous work (Atwood et al. 2005). It consists of a coplanar ring of 16 IR emitters/detector pairs, shown in Figure 2. It operates in two modes: a beacon mode and a receiver mode. In beacon mode, the sensor emits a plane of IR light in all directions. Since all robots are homogeneous and operate within a plane, every robot that is within line of sight will receive this flash of infrared light. In receiver mode, the robot polls each of its 16 IR receivers to determine which has the highest intensity reading, which indicates the bearing to that robot.

This sensor gives us bearing information between pairs of robots, but is unable to provide any range information as there is no reliable way to gain this information from unmodulated IR. Note that in order for this sensor to be of value, the robots must coordinate to ensure that only one robots emits light at one time (so that no destructive interference occurs), and that robots know when another robot has emitted so that they can enter receiving mode at the right time. This coordination is achieved by the wireless and localization infrastructure.

Infrastructure

The Colony wireless network combines a fully-connected network with a token-ring coordination scheme. The network is ad-hoc, and robots can join or leave the network at any time. With this architecture, robots continuously pass a token around the ring. To ensure that no two robots send messages or flash their BOM at the same time, only the robot with the token is allowed to send wireless packets or flash its BOM. Each packet contains the sender's ID, the robot ID of the token's recipient when it is sent, data used for behaviors, localization data, and a checksum to ensure packet integrity. By coordinating BOM flashes with wireless communication, robots that receive wireless messages know the relative bearing of the sending robot by detecting the highest BOM intensity reading. Communication and localization are thus fused into one problem, leveraging the strengths of both the wireless network and the BOM operation.

Each robot maintains a matrix storing the bearing between any pair of robots. Since the robots are constantly updating their bearing measurements, whenever a robot receives the token, that robot sends a wireless message with its updated bearing measurements to each of the other robots. As the token is passed around, all other listening robots update their bearing matrices with the data received from each sending robot. In this way, the bearing matrix is propagated amongst all robots. This bearing matrix provides the robots with a method for topological localization.

Robot Control Architecture

Our architecture for robot control is a layered architecture similar to the one proposed in (Simmons et al. 2002). At the highest level, a robot can be executing any one of a series of tasks, which are high-level constructs of desired actions. These tasks are decomposed into actions that are executed by the executive layer. These actions interact with the base control layer, which actuates the motors and other I/O devices. Finite State Machines (FSMs) are used to control program execution and store robot state.

Figure 3: *Several robots performing obstacle avoidance*

Robot Behaviors

Our early research with Colony robots focused on emergent behaviors in which each robot acts independently using simple local rules to form a complex colony-wide behavior. The simplest of such behaviors that can be demonstrated is swarming. In this behavior, each robot uses its rangefinders to avoid obstacles (see Figure 3). This simple obstacle avoidance behavior requires no inter-robot communication and can easily scale without degrading performance as more robots are added.

Adding wireless communication and bearing-only localization allows colony robots to perform a lemming-like behavior in which each robot will follow another robot in a chain. One robot assigns itself as the leader and all subsequent robots follow the robot in front of it. This allows the behavior to scale to arbitrarily long chains of robots.

More recent research has focused on leveraging colony robot capabilities to perform more intelligent and cooperative behaviors. Using localization data provided in the bearing matrix, Colony robots can effectively seek a target independently and then converge on the target as a group once it has been found. Figure 4 shows one such scenario in which robots are placed in an unknown environment and work collectively to seek a goal. Once a robot has found the goal, each robot is capable of finding the shortest line-of-sight path to that robot by running a simple graph search on its bearing matrix.

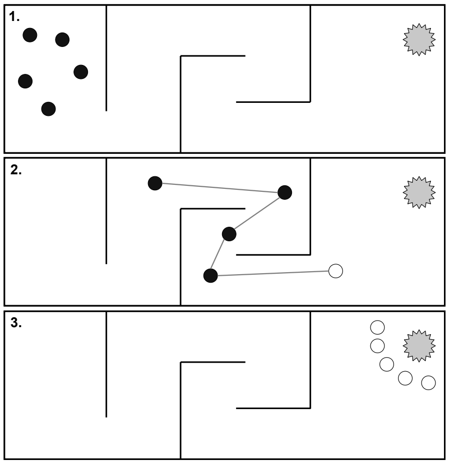

Figure 4: *Cooperative maze solving. 1) Robots begin to randomly seek the goal. 2) One robot finds the goal, and alerts its teammates. Each robot finds a line-of-sight path and moves towards the robot that spotted the light source. 3) The entire colony of robots can converges on the goal.*

Autonomous Recharging

Power management is an issue rarely addressed in mobile robotics. However, this becomes key problem in large groups of robots, especially those that will potentially run autonomously for extended periods of time in distant and hostile environments. Robots operating without human intervention for a long time must be able to recharge their batteries autonomously. In exploring an autonomous recharging solution for our Colony, we have strived to utilize as much of the existing sensors and infrastructure as possible. We have augmented the current system with three custom pieces of hardware that comprise our charging solution: 1) a charging station, 2) a medium-range homing sensor, and 3) a battery charging board.

Charging Station The charging station, shown in Figure 5 consists of three primary components: a standard ATX power supply that provides power (12V) to charge the batteries, a controller board which communicates with the Colony and schedules charging, and pairs of charging bays that robots physically dock with. The power supply and the controller board are typically stored outside the robot environment, while the charging bays line the edges of the environment. Up to eight charging bays can be daisy-chained together and controller by a single controller. Each charging bay provides a docking area for a robot with charging contacts to power the robots, as well as two different beacons that facilitate the homing process. An IR beacon very similar to the BOM provides long-range (0.5m – 2m) bearing measurements, and a pair of modulated IR emitters enable precise mid-range (5cm – 50cm) guidance. Physical grooves guide the robots into their final charging location (see Figure 6).

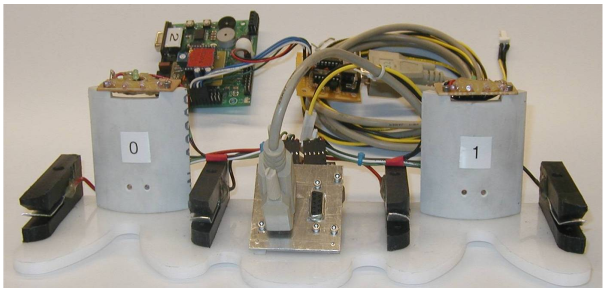

Figure 5: *Charging station with two docking bays*

Battery Charging Board Each robot is equipped with a custom daughter board used to control battery recharging (see Figure 7). Software and hardware on the recharge board regulate the current flowing into the battery and decide when to terminate charging. The board monitors the voltage, current, and temperature of the battery. Current is held at a constant level until either the voltage or the change in temperature reach a peak value, indicating the end of the charge cycle. The charge board also facilitates docking by interpreting the homing signal from the midrange IR sensor used by the charging bay. The board communicates this sensor data as well as status events concerning charging to the robot using the I^2C protocol. The robot scheduler processes these events to control the recharging and homing processes.

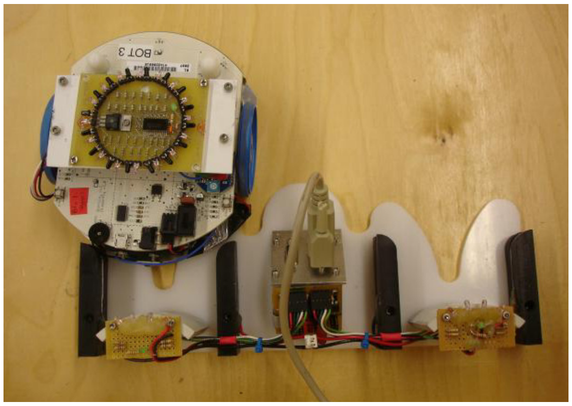

Figure 6: *Robot docking with a charging station bay*

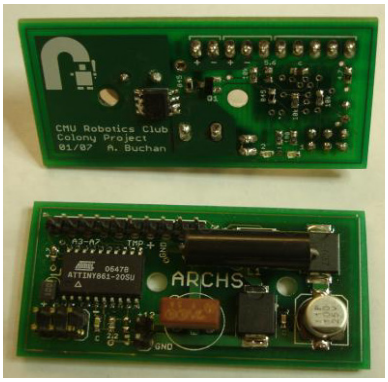

Figure 7: *Charging board*

Task Scheduler and Bay Allocation

Recharging is a complex task involving many actions that the robots can execute. When a low battery signal is detected, the robot will switch from its default task to the recharging task, and it will remain in this state until charging has completed. In the charging task, the robot sends a charging request over wireless to the charging station. If the charging station is full or if another robot is currently attempting to dock, the station denies the request and the requesting robot waits before requesting again. Otherwise, the station assigns the robot a bay using its bay allocation algorithm.

The bay allocation algorithm ensures that multiple bays are not assigned to a single robot. If possible, it also attempts to minimize the likelihood of collisions during homing by maximizing the distance between allocated bays. By intelligently managing its docking bays, the charging station can effectively manage a limited number of resources (charging bays) that must be shared between a large number of robots.

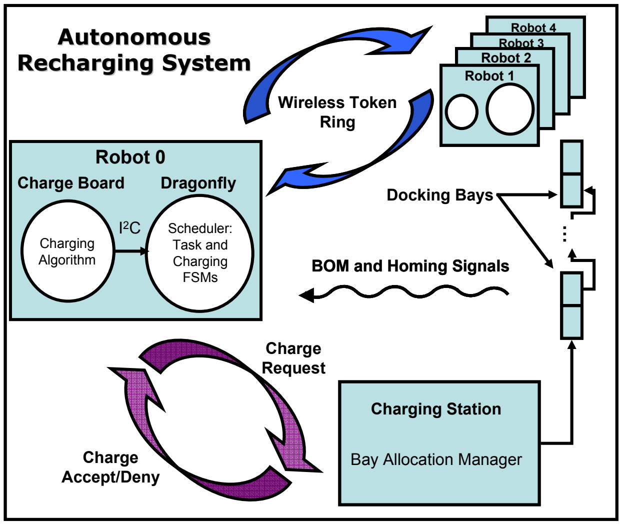

Figure 8: *Overview of the Autonomous Recharging System*

Once a robot receives an acceptance message from a charging station with a bay assignment, the robot begins to home in by first seeking with the BOM. This allows the robot to determine the general direction of the charging bay and then move towards it. When the robot is close enough, it will detect the midrange IR homing signal which it will then use to accurately dock with the charging station. Once the robot makes contact with the charging pads, the battery charging board will trigger a notification to the robot that instructs it to stop moving. The charging board will automatically begin to recharge the battery and will then send another notification instructing the robot to move away from the docking bay once the battery is fully charged. After the robot has left the docking bay, the recharging task relinquishes control of the robot, and the robot resumes its regular task. Figure 8 illustrates the entire autonomous recharging system.

ColoNet

ColoNet is an interface between the Colony and users across the Internet. The core functionally of ColoNet allows a user to send commands and requests from a client terminal via the Internet to robots and returns responses from the robots back to the client. ColoNet passes client information via TCP/IP to a server physically near the robot environment. The server then relays that information wirelessly to the robots. Combined with live webcam feeds and a universally accessible web application, ColoNet allows users to remotely monitor and control a large number of robots from anywhere in the world. ColoNet consists of three primary modules: the robot library, the server, and the client (see Figure 9).

The robot library maps high level server requests to basic robot functions and provides an interface for robots communicating over ColoNet by handling wireless communication to and from the server. The wireless interface between the robots and the server is implemented using a Colony robot programmed to relay wireless data to and from a computer via a USB port and with a server-side wireless library.

The server application handles requests from the robots and then relays information to the client through a TCP/IP connection. Conversely, the server application routes requests and commands from connected clients to the wireless interface used by the robots. The server can accept an arbitrary number of TCP/IP connections with clients, allowing for multiple users to control different robots at the same time.

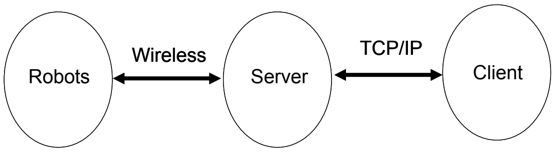

Figure 9: *ColoNet modules and interconnections*

ColoNet provides an application that allows the user to monitor and control the robots in real-time. The current client application is a browser-based Java applet which includes an overhead camera feed of the Colony environment and simple controls allowing the user to select a robot and steer it, set its orb color, turn its buzzer on and off, and report its battery life.

Future implementations of ColoNet will enable task queuing and bi-directional remote procedure calls. Task queuing will allow users to send high level tasks to the Colony which in turn will be executed by the robots. Remote procedure calls will allow robots to request services from ColoNet and vice versa.

Conclusions and Future Project Direction

We have developed a robot colony consisting of small agents that can autonomously accomplish intelligent tasks. Our Colony is unique in that it consists entirely of low-cost agents. We have demonstrated software algorithms and custom hardware that enables the robots to communicate, localize, and perform complex tasks such as autonomous recharging.

Currently, ongoing work seeks to streamline the wireless network and localization infrastructure by updating the wireless network library and creating a new, upgraded version of the BOM that will enable us to receive both range measurements as well as communicate over infrared. This new sensor and communication device has been named the RBOM, or Range Bearing and Orientation Module. The new wireless library will enable multiple robots to send packets simultaneously using mesh networking instead of requiring full connectivity. This new network will be faster and more reliable than the current wireless network implementation.

Using our autonomous recharging capabilities, we envision the Colony to be able to operate for extended periods of time (on the scale of weeks). Although complete autonomy is always desired, dealing with the inherent limitations of the Colony means that robots will occasionally tangle against each other, or fail to dock properly. These unexpected and unpredictable events are very difficult to detect autonomously, whereas a user can easily identify and fix the problem. Since constant user monitoring for week-long trials is not feasible, our ColoNet interface will enable us to monitor the Colony from anywhere, and take control over individual robots should such a problem occur.

We will make the charging process more efficient, and develop methods for coordinating several charging stations in different locations. Furthermore, we will increase the complexity of tasks being performed by the robots, such as having them cooperatively carry or push large objects. We also hope to be able to utilize our rangefinders to perform cooperative mapping and exploration of unknown areas. Mapping, like many other tasks, requires large computational and storage capabilities not available on the current Colony robots. We will investigate ways to offload this computation to a local ColoNet server.

Acknowledgments

This work was undertaken by members of the Colony project, an undergraduate research project operating in the Carnegie Mellon Robotics Club. The authors would first like to thank past and present Colony members: Duncan Alexander, Chris Atwood, Austin Buchan, Ben Berkowitz, Brian Coltin, Felix Duvallet, Siyuan Feng, Ryan Kellogg, Aaron Johnson, Rich Juchniewicz, Jason Knichel, James Kong, Christopher Mar, Eugene Marinelli, Allison Naaktgeboren, Bradford Neuman, Suresh Nidhiry, Iain Proctor, Justine Rembisz, Justin Scheiner, Steven Shamlian, Gregory Tress, Pras Velagapudi, Kevin Woo, and Cornell Wright. We would especially like to thank our advisor, George Kantor for his guidance and advice throughout the year. We would also like to thank Howie Choset and Peggy Martin for their generous support and tremendous help, as well as Brian Kirby and Tom Lauwers for their contributions to this project. The Colony project is funded in part by Carnegie Mellon's Undergraduate Research Office and a Ford Undergraduate Research Grant. The results represent the views of the authors and not those of Carnegie Mellon University.

References

Atwood, C.; Duvallet, F.; Johnson, A.; Juchniewicz R.; Kellogg R,; Killfoile K.; Naaktgeboren A.; Nidhiry S.; Proctor I,; Rembisz J.; Shamlian S.; Velagapudi P. 2005. Relative Localization in Colony Robots. In Proceedings of the National Conference On Undergraduate Research.

Chaimowicz, L.; Sugar, T.; Kumar, V.; and Campos, M. 2006. An Architecture for Tightly Coupled Multi-Robot Cooperation. In Proceedings of the International Conference on Robotics and Automation.

Cortes, J.; Martínez, S.; Karatas, T.; and Bullo, F. 2002. Coverage Control for Mobile Sensing Networks. In

Proceedings of the IEEE Conference on Robotics and Automation, 1327-1332. Arlington, VA.

Fox, D.; Burgard, W.; Kruppa, H.; and Thrun, S. 2000. A probabilistic approach to collaborative multi-robot localization. Autonomous Robots 8(3).

Heger, F., and Singh, S. 2006. Sliding Autonomy for Complex Coordinated Multi-Robot Tasks: Analysis & Experiments. In Proceedings, Robotics: Systems and Science, Philadelphia.

Howard, A.; Parker, L.; and Sukhatme, G. 2006. Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection. The International Journal of Robotics Research 25: 431-447.

Hundwork, M.; Goradia, A.; Ning, X.; Haffner, C.; Klochko, C.; and Mutka, M. 2006. Pervasive surveillance using a cooperative mobile sensor network. In Proceedings of the IEEE International Conference on Robotics and Automation.

McLurkin, J. 2004. Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots. M.S. diss., Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass.

Sugar, T. G., and Kumar, V. C. 2002. Control of Cooperating Mobile Manipulators. In IEEE Transactions on Robotics and Automation, Vol.18, No.1, 94-103.

Trebi-Ollennu, A.; Nayar, H.D.; Aghazarian, H.; Ganino, A.; Pirjanian, P.; Kennedy, B.; Huntsberger, T.; and Schenker, P. 2002. Mars Rover Pair Cooperatively Transporting a Long Payload. In Proceedings of the IEEE International Conference on Robotics and Automation.