
Multi-Robot Systems Lab Research Projects

The Multi-Robot Systems lab at Rice university is looking for talented student researchers to work on a variety of projects. This document only contains the projects the I know I need to do. If you have a project idea that you find irresistible, please let me know. If I also find it irresistible, then all we need to do is to figure out how to do it.

I can be reached at jmclurkin@rice.edu. Thank you for your interest in our research group.

1 Basic Projects

These are general-purpose projects. They might involve some interaction with robots, but that is not their main focus. They require a basic understanding of programming and robotic systems.

1.1 Robot Setup and Player/Stage Testing

Setup and test the software that comes with the ERA-MOBI robot. This includes the player/stage simulation and control environment.

Skills needed: Linux (Ubuntu), maybe C or C++

1.2 Robot Orientation Measurement

Design and implement an measurement system to measure the ground-truth global heading of the robots in the system. This will be compared to the data collection system's estimates from the combination of camera positioning and inter-robot localization. This project will involve significant mechanical fabrication of a test rig to determine the ground-truth headings of the robot.

Skills needed: Mechanical design and fabrication

1.3 "FlatBot" Documentation

Write a careful document that clearly explains how to build the "FlatBot" robot classroom lab. Visit: <http://people.csail.mit.edu/jamesm/project-FlatBot.php> for more information.

Skills needed: A little C programming background, writing skills, basic knowledge of electronics

2 Advanced Projects

These are more advanced general-purpose projects. They might involve some interaction with robots, but that is not their main focus. They require a basic understanding of programming and robotic systems.

2.1 Boundary Detection Analysis

Design and implement an algorithm that constructs the C-Shape boundary *shape* [?].

Skills needed: Strong Java programming background, interest in computational geometry

2.2 Open-Source IR Localization System

Design and implement an infra-red inter-robot localization system that captures the *local network geometry* of a robot in a multi-robot system. This is a fancy way of saying a system that can determine the positions of neighboring robots. This system should be easy to build and use, and

inexpensive to implement.

Skills needed: Circuit design, microcontroller programming, PCB fabrication

2.3 Outdoor + 2.5-D Localization System

Design and implement an inter-robot localization system that captures the *local network geometry* of a robot in a multi-robot system. This is a fancy way of saying a system that can determine the positions of neighboring robots. This system should work on non-level floors, over long distances, and outdoors. This system should be a high-fidelity research-grade design.

Skills needed: Advanced circuit design, microcontroller programming, PCB fabrication

2.4 SwarmBot Camera Demonstrator

Design and implement some basic vision algorithms on the SwarmBots. These robots have limited processing power, so the vision processing will need to be carefully designed to work in these constraints. The first step is to build a simple JPEG compressor so that we can easily transmit images from the robots via their radio. Bonus points are given for an algorithm that uses the vision information from multiple robots.

Skills needed: Strong C programming background, interest in computer vision

2.5 Simple SwarmBot Simulator

Design and implement a simple simulator for the SwarmBots. I insist that it is written in Java, and is clean, well-documented, and very simple. It should present a simplified programming model of the SwarmBots and allow for repeated experiments and data collection. Did I mention that it should also be simple?

Skills needed: Strong Java programming background

2.6 Global Coordinate Estimation

Design and implement a distributed algorithm to estimate the global pose, $p = x, y, \theta$ of each robot. The robots should compute their pose based on the local measurements of their neighbors and the propagation of information throughout the network. It should be written in MATLAB and then C to run on the robots.

Skills needed: MATLAB, C, probability and other assorted math

3 Multi-Robot Distributed Algorithm Projects

These projects will be implemented on the preexisting SwarmBot robots []. A strong knowledge of the C programming language is required for these tasks. Each project is a conference-paper sized amount of work. In particular, the 2011 IEEE International Conference on Robotics and Automation (ICRA) to be held on May 9-13 in Shanghai, China. Any student that gets a paper accepted from my lab travels (and eats and sleeps) on my dime. If the prize is this good, what's the catch? Each of these projects will take a substantial amount of time to complete, test, and document properly. Depending on your background, you can plan on two semesters of work, maybe even part of the summer. But you can also expect to learn a great deal, and have a lot of fun in the basement of Ryon Lab.

3.1 Network Tethering

Design and test a distributed algorithm to produce a k -connected communications graph. There already exist algorithms for $k = 1$ [], so the fun starts at $k = 2$ and extends for larger k .

Skills needed: Strong C programming background, interest in distributed algorithms

3.2 Boundary Detection Algorithms: Network Containment and Void Remediation

Design and test a distributed algorithm that uses the C-Shape boundary detection algorithm [?] to contain a configuration of robots. The same (or similar) algorithm can also be used to move robots into a network void, and to reinforce local articulation points in the graph.

Skills needed: Strong C programming background, interest in distributed algorithms and computational geometry

3.3 Physical Bubble Sort

Design and test a distributed algorithm to perform a physical bubble sort while maintaining network connectivity. The current algorithm moves robots in between their neighbors, with no regard for the network links that are broken in the process. This can disconnect the swarm, and cause the entire algorithm to fail. A better implementation is to use a “do-si-do” motion behavior to sort the robots while guaranteeing connectivity. This will be a challenge to implement and prove robust, but how often do you get to use the word “do-si-do” in a technical publication?

Skills needed: Strong C programming background, interest in distributed algorithms

3.4 Optimal Uniform Honeycomb Coverage

Design and test a distributed algorithm to arrange the robots into a honeycomb pattern. This will provide the minimum number of robots required to completely and uniformly cover an area, while providing a provable upper bound on the maximum spanning ratio of the resulting network. The hard part will be implementing this algorithm on the physical hardware, and getting it to converge in to a stable honeycomb lattice. See work by Spears and Spears on multi-robot lattice formation[?, ?].

Skills needed: Strong C programming background, interest in distributed algorithms, graph theory, and computational geometry

3.5 Swarm Pipe Organ

Build and test a MIDI interface to the swarm. Each robot has a 1-watt audio system, imagine what 60 of them playing in unison would sound like!

Skills needed: C programming background, microcontroller operation, some hardware fabrication

4 “Master” Projects

New to robotics? Need time to build up the programming or mathematics background to do some of the other projects? Then look through these “master” projects to find something that might fit you better. These projects will give you the time to work on some smaller Java or C programming projects to get you ready to tackle some of the more rigorous projects listed above.

4.1 Datamaster

Setup and use the SwarmBot data collection system. Main task is the recalibration of the vision-based system that tracks the positions of the robots. Currently, it is designed to view the robots from the ceiling, it has to be adapted to view them from an angle. This part of the project will require some knowledge of the geometry of optics, and some Java programming skills to implement the changes. The next step is to build a robot playpen for data collection. Finally, data for an upcoming paper on stochastic multi-robot systems needs to be collected. But this means you get third author in a

prestigious conference, which makes all the other work worth it, right?

Skills needed: Java, linear algebra for camera calibration, patience for data collection, skills with hand tools

4.2 Webmaster

Build a first-rate Wiki-based lab website for the Multi-Robot Systems Lab. Use <http://groups.csail.mit.edu/drl> as a reference design. Will also be responsible for image and video editing (or finding someone who can do image and video editing).

Skills needed: HTML, Wiki setup and administration, image and video editing

4.3 Partsmaster

Build a first-rate research lab. Responsibilities include ordering about \$500 worth of parts per week, labeling little plastic drawers, ordering tools, optimizing desk placement for optimal workflow. The ideal candidate would have a strong desire to click on lots of purchasing websites, coupled with a compulsion to organize and label the parts once they arrive.

Skills needed: Advanced shopping skills, a touch of OCD

4.4 Batterymaster

The batteries in the SwarmBots are 6 years old, and they are all well beyond their useful lifetimes. This project entails ordering new batteries with custom wiring harnesses, writing up a details assembly/disassembly document for the robots, and organizing a SwarmBot battery party, which is really just a fancy name for a pizza-fueled assembly line for disassembling, swapping batteries, and reassembling all 100 SwarmBots. Did I mention free food?

Skills needed: Organization skills, group management, and hand tool use

4.5 LEGOmaster

Evaluate the various development environments available for the LEGO Mindstorms robot system. Develop a good example robot and program in each language, and carefully document the results.

Skills needed: Basic programming skills, love of little plastic bricks.