# Multi-Robot Search and Rescue: An Open-Ended Educational Bridge Between Theory and Practice

Philip Twu
Air and Missile Defense Department
Johns Hopkins University Applied Physics Lab
Laurel, MD 20723–6005
Email: philip.twu@jhuapl.edu

Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332–0001
Email: magnus@gatech.edu

***Abstract*—This paper reports on a multi-robot search and rescue final project that has been used at the Georgia Institute of Technology to educate students on how to methodically apply networked control theory concepts towards solving complex cyber-physical system (CPS) engineering problems. For the project, students design control laws to coordinate a team of simulated robots in completing a set of mission objectives within a custom-developed virtual environment. The virtual environment lets students script high-level algorithms and experience how their computational solutions perform when coupled with both physical constraints and environmental factors, as is often the case in real robotics applications. By allowing certain physical domain effects to be toggled on or off, students learn to iteratively adapt theoretical solutions based on simplified mathematical models to obtain engineering solutions for complex CPS problems.**

## I. Introduction

Educators in the systems and controls engineering field are oftentimes faced with a dilemma. On one hand, the need to control increasingly complex systems creates the need to teach advanced controls techniques that rely heavily on mathematical theory. On the other hand, technical assumptions used to arrive at computational solutions tend to oversimplify complex coupling behaviors with the physical domain, thereby creating the so-called gap between theory and practice. The need to close this gap and equip students with a solid understanding of how to design for cyber-physical systems (CPS) grows ever more urgent as engineered systems advance to become further integrated with one another and with the physical environment.

Many theoretically-oriented courses attempt to prepare students for solving "real-world" engineering problems by including numerical simulation problems in homework assignments. However, since most of these simulation problems amount to just a straightforward implementation of algorithms learned in class, students usually only learn about the computational challenges involved in implementing their solution, but not the cyber-physical challenge. Lab experiments with physical equipment can potentially provide the most exposure to CPS issues, but run the risk of overwhelming students with engineering problems that are significantly more complex than what other theoretically-oriented courses had prepared them for. Moreover, lab equipment can be expensive to maintain and the time commitment needed to work with them may require the addition of new courses into the curriculum.

One attempt to expose students to CPS engineering within a lecture-based class has been through the use of smaller-scaled "take-home" labs (e.g., [1], [2], [3], [4]) that supplement traditional lectures with portable and inexpensive experiments which students can work on at home. However, not all application domains are suitable for take-home labs. For example, in a multi-robot coordination course, the use of take-home labs may not be appropriate since both the cost for equipment and probability of hardware failure increases with the number of robots involved [5]. Furthermore, the performance of many multi-robot algorithms rely on the interaction of robots within a specific environment (e.g., search and rescue, sensor coverage, obstacle avoidance, convoy protection), which cannot be "taken home" by the student. A more appealing approach for teaching CPS engineering in application domains not suitable for take-home labs is through the use of "virtual environments" (e.g., [6], [7], [8], [9]), where students conduct experiments in large-scale modeling and simulation platforms that operate at a level of abstraction appropriate for an educational tool.

At the Georgia Institute of Technology, it was our ambition to teach students in the graduate-level networked control systems course (ECE8823) how to apply the computational algorithms learned in lecture towards engineering solutions for challenging multi-robot coordination problems with both cyber and physical aspects. In both Fall 2010 and Fall 2011 semesters, ECE8823 students were given a final project where they had to design high-level control laws to coordinate a team of simulated robots in completing a multi-robot search and rescue mission. Using a virtual environment created in MATLAB specifically for this project, students experienced first-hand the coupling of their computational solutions to physical constraints and environmental effects. Upon completing the project, students learned to iteratively adapt theoretical solutions based on simplified mathematical models to methodically obtain engineering solutions for complex CPS problems.

## II. Multi-Robot Search and Rescue

During the last month of the semester, students in ECE8823 each received the following set of instructions:

*The year is 2030 and NASA has identified an asteroid that is on a collision course with Earth! In order to deflect the asteroid, the scientists require samples from its surface to determine its physical composition. They have asked the robotics faculty at Georgia Tech to plan a multi-robot expedition to collect samples from the asteroid's surface and bring them back to Earth for analysis. The robots managed to land on the asteroid* *successfully and were able to gather the samples. However, an unexpected pulse of electromagnetic radiation temporarily disabled the electronics on-board the robots, stranding them on the asteroid.*

*Based on your experience in networked controls from having taken ECE8823, members of the robotics faculty at Georgia Tech have selected you to lead a rescue mission. Using the beacons placed on the surface of the asteroid from the first expedition for navigation, your mission is to design decentralized controllers for the multi-robot rescue team to:*

1) *Navigate a team of 6 robots through the rough terrain of the asteroid*
2) *Locate and re-activate the 6 disabled robots from the first expedition*
3) *Bring both robot teams back to the platform (leave no robot behind) and get into a specific formation to wait to be picked up by an orbiting spacecraft.*

Along with the instructions, students in ECE8823 were given MATLAB code to simulate a team of robots navigating through a virtual environment with 6 waypoints that must be cleared in order, as shown in Figure 1. For simplicity, the terrain was assumed to be planar and so each robot $i$ had position coordinates $x_i \in \mathbb{R}^2$. Moreover, each robot was given single-integrator dynamics so that students could focus on designing high-level coordination strategies instead of spending time on low-level control. Each of the robots were equipped with omnidirectional sensors, allowing them to detect neighboring robots and obstacles that were within a distance $\Delta$. The network topology at time $t$ could therefore be represented by an undirected graph $G(t) = (V(t), E(t))$ where $V(t) \subseteq \{1, \ldots, N\}$ is the vertex set, with each vertex corresponding to the agent in the network with the same index. The edge set $E(t) \subset V(t) \times V(t)$ is such that an unordered pair $(i, j) \in E(t)$ if and only if $||x_i(t) - x_j(t)|| \leq \Delta$. Actuator saturation and random sensor noise, both of which could be toggled on or off during testing, were used to represent physical limitations of the robot's sensing and actuation capabilities. Moreover, physical limitations on the robots' computational capabilities were modeled by allowing each robot to sense and compute a new control signal only once every $T$ seconds.

For each waypoint, students were tasked to write a single decentralized controller that when executed on each of the robots, would maneuver the multi-robot team to complete a system-level objective. MATLAB function templates were provided to the students for writing the decentralized controllers used to clear each waypoint. To ensure the coordination algorithms that students designed were indeed decentralized, each function was restricted to computing the robot's control signal while only taking as input the robot's unique ID, state, list of neighboring robots given by the current network topology, relative displacement measurements within the robot's local coordinate frame to nearby neighbors and obstacles, a flag indicating whether it was the first time the robot was executing the controller, and locally stored information within the robot's limited internal memory. A single "leader" robot was given the ability to sense its relative displacement to the current waypoint and could use that information to help the team navigate. The need for computed control laws to take into account the physical structure of the robots was emphasized by giving each robot a physical radius $D$. Although obstacle and inter-robot collisions must be avoided in the final solution, both physical effects could be toggled on or off during testing.

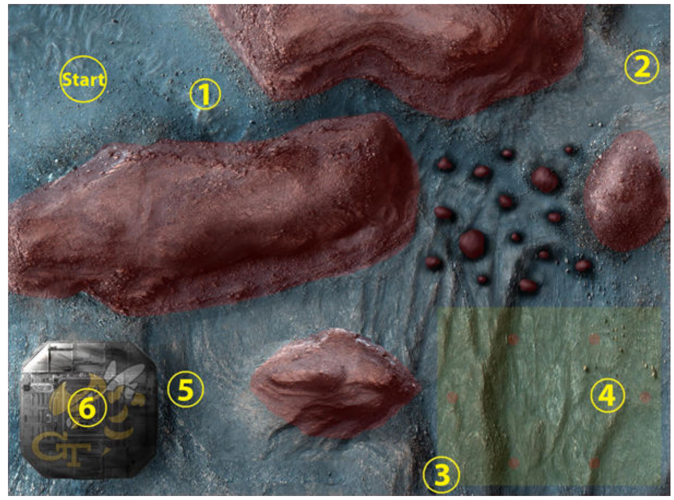


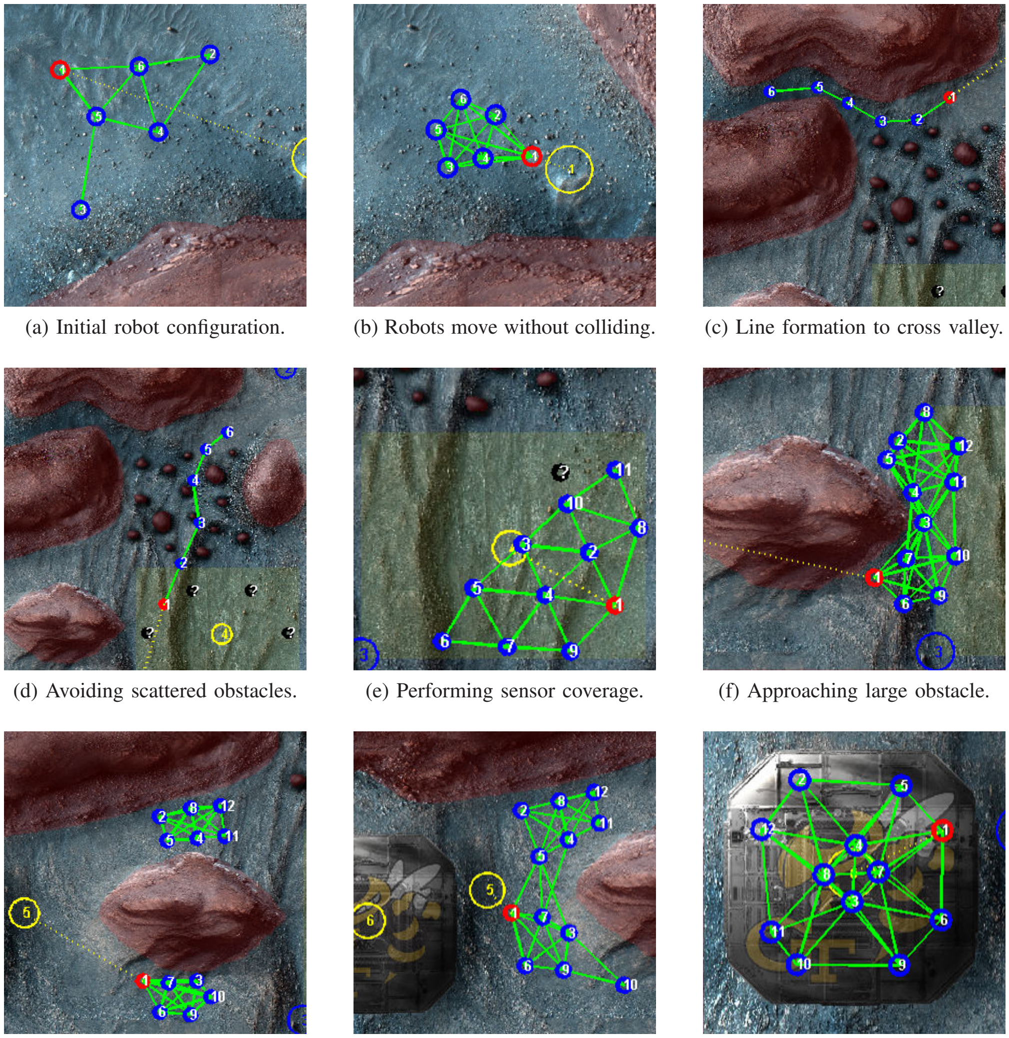
Fig. 1: Screenshot of the virtual environment used for the multi-robot search and rescue final project. Students design decentralized controllers to navigate a team of robots through all 6 waypoints in order, where each waypoint challenges the student to apply different concepts learned throughout class.

A waypoint is cleared when all previous waypoints have been cleared, a robot is at the current waypoint, and the current network topology is connected, i.e., a path of edges exists between each pair of robots. Each waypoint represents a different type of system-level objective for the robots, and thus challenged students to apply different concepts learned throughout the class to complete the overall mission. Referring to Figure 2a, the first waypoint requires the robots to simply move from one point to another as a team but without colliding with one another or with the environment. Looking at the map in Figure 1, the robots then must travel through a narrow valley to reach Waypoint 2, and then navigate through a field littered with small obstacles to reach Waypoint 3. To clear Waypoint 4, the robots must search a bounded area to recover the 6 stranded robots from the previous mission. All 12 robots must then perform a "splitting and merging" maneuver around a large obstacle to reach Waypoint 5. Finally, the robots must all move onto the platform on Waypoint 6 and get into a particular formation, as shown in Figure 2i, to await rescue.

## III. Project Outcomes

### A. Students' Solutions

The multi-robot search and rescue final project consists of two coupled challenges that must be solved. The networked controls challenge is to devise decentralized control laws for robots to coordinate and perform mission-level objectives such as getting into formation, maintaining network connectivity, or searching an enclosed area. The CPS challenge, on the other hand, requires students to take the theoretical solutions obtained from the first challenge and adapt them to take into account physical effects such as actuator saturation, sensor noise, and interactions with the environment. In order to



(a) Initial robot configuration. (b) Robots move without colliding. (c) Line formation to cross valley.

(d) Avoiding scattered obstacles. (e) Performing sensor coverage. (f) Approaching large obstacle.

(g) Network splits into two teams. (h) Teams merge after crossing. (i) Final formation to await rescue.

Fig. 2: Screenshots showing the students' solutions for solving all 6 waypoints in the multi-robot search and rescue mission.

successfully complete the search and rescue mission, students must design control laws which take both aspects of design into consideration while balancing multiple objectives at once.

As an example, notice how the ability to move cohesively as a team while maintaining network connectivity and avoiding collisions is needed to clear any of the 6 waypoints. Waypoint 1 was designed to give the students a chance to solidify their solution to this problem before moving on to more difficult tasks. The topic of weighted consensus protocols was discussed in class, where each robot $i$ moves with a velocity vector that is a weighted sum of the relative displacement vectors between itself and each of its neighbors:

$$\dot{x}_i(t) = -\sum_{j \in N_i(t)} w_{ij}(t)\left(x_i(t) - x_j(t)\right), \tag{1}$$

where $N_i(t) = \{j \mid (i,j) \in E(t)\}$ is the set of robot $i$'s current neighbors. By choosing the weight function $w_{ij}(t)$ carefully, it was shown in class, using Lyapunov-based arguments, that robots can be made to preserve network connectivity and maintain fixed distances from one another.

Despite mathematical solutions such as (1) being provably correct, the technical assumptions used to arrive at such conclusions may not always hold in the physical domain. For example, physical robots do not have unbounded velocities or infinite sampling frequencies. Nevertheless, to methodically arrive at a CPS solution, students had to use solutions based on simplified mathematical models as a starting point. A popular approach was to modify network connectivity preservation edge weights to account for actuator saturation and combine them with other edge weights to balance oftentimes competing objectives. Using these and similar methods, students were able to successfully clear Waypoint 1, as shown in Figure 2b.

To clear Waypoint 2, students must realize that the valley was too narrow to fit all the robots through at once. Methods for network topology control [10], as discussed in class, could be used to make the robots squeeze through the valley in a line configuration as shown in Figure 2c. Clearing Waypoint 3 also required obstacle avoidance, but students had to solve the problem of maintaining network connectivity even if different robots chose to take different paths through the field as seen in Figure 2d. Figure 2e shows students clearing Waypoint 4 by using Voronoi-based sensor coverage algorithms [11] to search the enclosed area for the stranded robots. Since a stranded robot reactivates and joins the network if another robot gets close enough to it, the sensor coverage algorithms used had to be scalable as robots dynamically join the network.

Waypoint 5 was the most difficult for students to clear since the "splitting and merging" maneuver, shown in Figures 2f, 2g, and 2h, caused the network to become disconnected as robots go around opposite sides of the obstacle. Most of the solutions that the students came up with fell into one of two categories. The first approach was to have the leader robot move towards the next waypoint and for all other robots to maintain a sense of "momentum". Therefore, even if the network became disconnected, robots would still have an idea of the overall direction that they should be moving in. Another approach was to have the leader robot move towards the next waypoint at a speed proportional to its distance from it. The neighboring robots could then estimate the position of the next waypoint through the leader's actions and become virtual leaders themselves. Finally, to clear Waypoint 6, students implemented many different heuristic-based methods to solve the distributed assignment problem of assigning a robot to each target point, without using any inter-robot communication.

### B. Student Discussions and Feedback

The multi-robot search and rescue final project has been successfully deployed in both Fall 2010 and Fall 2011 semesters of ECE8823 at the Georgia Institute of Technology. 28 of 41 students in the Fall 2010 semester were able to complete all 6 waypoints successfully, while 18 of the 19 students did so in Fall 2011. On the last day of class, a half hour is always dedicated to having students share and discuss their solutions with the entire class. The five fastest solutions in the class are unveiled one at a time, as students cheer and applaud for their fellow classmates.

An optional survey was given to the class for feedback on the project. 30 of 41 students from Fall 2010 responded, while 14 of 19 students in Fall 2011 responded. The feedback was overwhelmingly positive, with all 44 student responses stating that the material presented in lecture was helpful for completing the project, and recommending that the project be continued in future semesters. Students reported spending an average of 4 days (32 work hours) to complete the project.

When asked what they had learned from the experience, students commented on learning "how to combine different concepts in class for a working system," and felt that the project allowed them to "understand the ideas/subjects learned in the lecture more rigidly." Many responses included statements on how the project taught them to apply course material towards complex CPS engineering scenarios, as well as realizing the need to continue learning outside of the classroom. Examples include:

- "How to actually design for real situations."
- "Implementation is much harder than what I believed it would be based just on the material from lecture"
- "Theory and practice differ quite a bit, the encoding of weights in MATLAB had to be done in a different form than in theory."
- "There's a lot of ways to solve each problem, more research needs to be done to find optimal solutions."
- "I also had to research and learn into concepts that weren't exhaustively covered in class."

## IV. Future Outlook and Distribution

The multi-robot final project was shown to be a successful educational tool for teaching students in the networked controls application domain to utilize theoretical solutions based on simplified mathematical models as a starting point for methodically obtaining engineering solutions to complex CPS problems. The project will be continued in future semesters that the ECE8823 course is taught, with improvements made based on student feedback. Detailed instructions for assigning the multi-robot search and rescue final project, along with MATLAB source code for the virtual environment, are publicly available for download from the Internet at [12] for anyone interested in adopting a similar project in their own classes.


## Acknowledgment

The authors would like to thank all of the students from the Fall 2010 and Fall 2011 semesters of ECE8823 at the Georgia Institute of Technology, as well as Dr. Bonnie Heck Ferri for her helpful comments when preparing this paper.


## References


[1] T. Balch, J. Summet, D. Blank, D. Kumar, M. Guzdial, K. O'Hara, D. Walker, M. Sweat, G. Gupta, S. Tansley *et al.*, "Designing personal robots for education: Hardware, software, and curriculum," *IEEE Pervasive Computing*, pp. 5–9, 2008.

[2] W. Durfee, P. Li, and D. Waletzko, "At-home system and controls laboratories," in *Proceedings of the American Society of Engineering Education Annual Conference & Exposition*, 2005.

[3] B. Ferri, S. Ahmed, J. Michaels, E. Dean, C. Garyet, and S. Shearman, "Signal processing experiments with the LEGO MINDSTORMS NXT kit for use in signals and systems courses," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 3787–3792.

[4] B. Heck, N. Clements, and A. Ferri, "A LEGO experiment for embedded control system design," *Control Systems Magazine, IEEE*, vol. 24, no. 5, pp. 61–64, 2004.

[5] C. Kitts and M. Egerstedt, "Design, control, and applications of real-world multirobot systems [from the guest editors]," *Robotics Automation Magazine, IEEE*, vol. 15, no. 1, p. 8, 2008.

[6] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "USARSim: a robot simulator for research and education," in *Robotics and Automation, 2007 IEEE International Conference on.* IEEE, 2007, pp. 1400–1405.

[7] O. Michel, "Cyberbotics Ltd. WebotsTM: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[8] T. Murphey, "Teaching rigid body mechanics using student-created virtual environments," *Education, IEEE Transactions on*, vol. 51, no. 1, pp. 45–52, 2008.

[9] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intelligence*, vol. 2, no. 2, pp. 189–208, 2008.

[10] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks.* Princeton University Press, 2010.

[11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243–255, 2004.

[12] P. Twu and M. Egerstedt, "Ece8823 final project: Multi-robot search and rescue," http://users.ece.gatech.edu/∼magnus/ece8823finalproject.html, 2010, [Online; accessed 18-Jan.-2013].