An Integrated System for Command and Control of Cooperative Robotic Systems

Christopher M. Clark, Eric W. Frew, Henry L. Jones, & Stephen M. Rock

Aerospace Robotics Lab
Department of Aeronautics & Astronautics
Stanford University
{chrisc, ewf, hlj, rock}@sun-valley.stanford.edu

Abstract

Presented is an experimental investigation into three issues that enable increased autonomous functionality when using mobile robots. These issues are (1) interfacing a single user with multiple robots, (2) motion planning for multiple robots, and (3) robot trajectory generation for target tracking. For this research, the Micro Autonomous RoverS (MARS) test platform was developed that provides a means for implementing this technology on laboratory robots. Experimental results are presented in which a single user is able to command a group of robots to carry out tasks including collision-free motion and target tracking.

1. Introduction

Currently, remote robotic systems require many humans to operate a single robot. The goal for future systems is to require only one operator for many robots. For example, future space structure construction would benefit from the availability of a large group of robots that can be operated by a small group of humans.

While there has been a significant amount of research towards the operation of single remote robots, more work is still required towards the operation of groups of robots. In particular, an increased degree of autonomy must be given to the robots. To realize this autonomy, a variety of fundamental capabilities must be enabled that include:

> *(1) Providing an interface from which adequate information for decision-making is available to the human, and commands to one or several of the robots can be executed.*
> *(2) Providing autonomous, real-time construction of collision-free trajectories for all robots in the group*
> *(3) Providing robot trajectory generation that enables tracking of moving objects.*

In previous work, we addressed each of these issues: Jones [6] developed an interface that allows a single human to operate many robots; Clark [2] designed a multi-robot motion planner; and Frew [4] designed a trajectory generator that provides near-optimal solutions for object motion estimation.

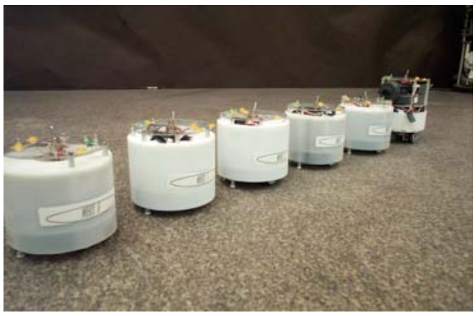


Figure 1: Rovers from the MARS test platform.

In this paper, we present an integrated system demonstration of these technologies.

The paper is organized as follows. Section 2 provides a brief description of each of the three research projects. In Section 3, the Micro Autonomous RoverS (MARS) test platform and its application to this research is described. Section 4 details a final demonstration that summarizes the previous research as an integrated system. Conclusions are presented in Section 5.

2. Technological Issues

2.1. Human Interface

For a single human to operate a multi-robot system, the human must have access to all relevant information about the remote environment so that appropriate commands can be executed. Also, the human must be provided with a means of executing these commands to one or several of the robots.

In [6], Jones developed an interface based on dialogues between the human and the robots as an effective method for operating multiple robots. In particular, Jones addressed the following issues:

- *Establishing the structure and scope of the dialogue*
- *Creating a robot infrastructure capable of conducting an effective dialogue*
- *Determining methods for dealing with the social conventions of dialogues*
- *Developing an interface that allows the operator to carry out the dialogue with the robotic system.*

His result is an implementation of a dialogue interaction patterned after the task-oriented dialogues common in human teams. The hypothesis is that similar dialogues can play a useful role within human-robot teams.

The interface was implemented using OpenGL to provide a three-dimensional view of the robot environment. An example screen-shot is shown in Figure 2. Dialogue through the interface takes place electronically rather than through voice. The dialogue begins when objects are selected by clicking on them on the screen. The interface then resolves the identity of the object and the robot that sensed the object. The interface waits until a response from the correct robot has been returned in the form of a list of tasks that the robot can accomplish on that object. This list is then displayed in a dialog that pops up next to the object. The user can select from this list of tasks, and the complete command of robot/task/object is sent to the robot for execution.

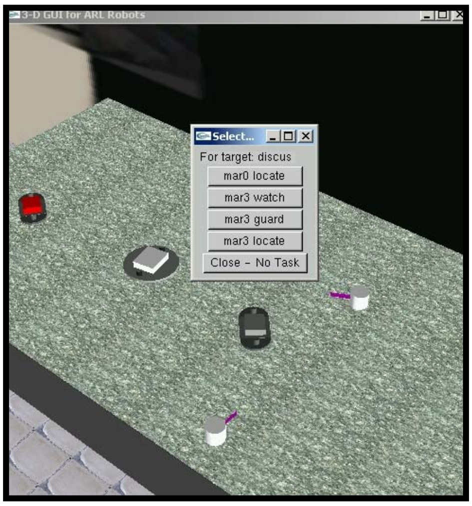


Figure 2: Screen shot of the human-robot interface.

The screen-shot in Figure 2 provides an example of the interface in use. Two robots (denoted by white cylinders) and 3 objects are located on the test bed workspace. The operator has queried the robot agents to determine what tasks can be performed on the object at the left side of the screen. A pop-up menu has appeared with a list of the query results.

Jones showed that it is possible to build a dialogue-based interaction that enables the control of multiple robots. This interaction, as implemented in a virtual three-dimensional world, provided an intuitive point-and-click method for determining the capabilities of the robot in the appropriate context, and enabled the operator to participate in the resource management and task planning for the robots.

2.2 Motion Planning

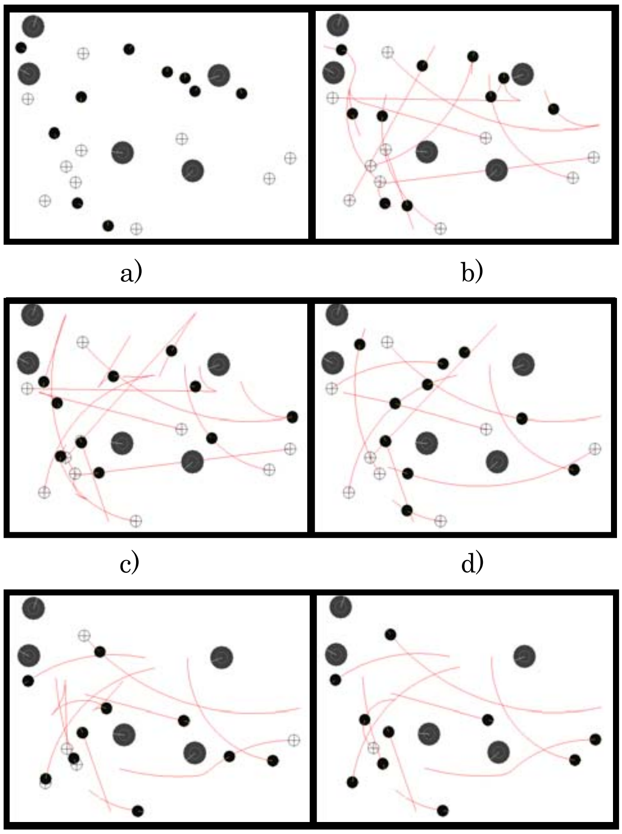
When large groups of robots and moving obstacles are working together within a designated area, high-level motion planning is required to avoid collisions. Continuous communication between all robots may not be feasible, and no system of sensors can provide global knowledge. Also, to function in a dynamic environment with moving obstacles, the system must be able to react quickly. For this type of multi-robot system, a motion planner that does not need global knowledge or high bandwidth communication, but that can still plan in real-time, is required.

A motion planning system that meets this requirement was developed by Clark [2]. The algorithm presented was based on the planner developed by Hsu and Kindel [5]. Their work demonstrates the use of a Kinodynamic Randomized Motion Planner for a single robot maneuvering around stationary and moving obstacles.

To handle more than one robot, the Kinodynamic Randomized Motion Planner was extended. In the extended planner, when robots detect one another using local sensors, they communicate with each other. Using a priority system, the robots coordinate their motion plans to avoid collisions. Each robot creates a plan with knowledge of only the few obstacles surrounding it. By planning around only those objects within the robot's local area, the motion planning problem is greatly simplified leading to decreased planning times. When new objects enter the robot's field of view, a re-plan is called for to ensure that the robot's trajectory is collision-free.

An example of a simulation involving 10 rovers, and 5 stationary obstacles is provided if Figure 3. Smaller circles represent the micro-rovers as viewed from above, while crosses represent goal locations and larger circles represent obstacles. Trajectories constructed by each robot's motion planner are indicated with lines that lead to goal locations.

The motion planner demonstrated its effectiveness in planning for a large number of robots within a bounded workspace. It planned with a high probability of success, even in "cluttered" environments involving 5 to 15 robots, stationary obstacles and moving obstacles. Planning times on the order of 0.1 s allowed the robots to re-plan in real-time and react quickly to changes in the environment. Although the motion planner was applied to a 2D workspace, it should be noted that the planner is extendible to 3D workspaces.



a) b)

c) d)

e) f)

Figure 3: Motion planning simulation example involving 10 robots and 5 obstacles. Figure a) illustrates rovers, their goals, and obstacles before the simulation. In b), the simulation has just begun and all rovers have constructed their first plan. Examples of real-time planning on the fly are shown in c) and d). (Note that some initial trajectories pass through obstacles, but as the robots come close enough to sense them, they replan to avoid them.) Finally, in e) the rovers have constructed their last trajctory and are headed towards their respective goal location. Figure f) shows all but one rover having reached their goal location.

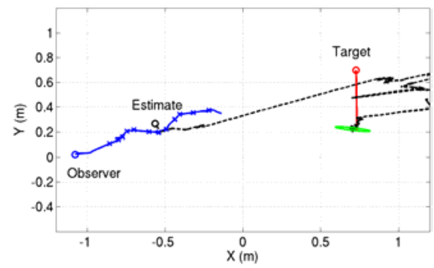
2.3 Trajectory Design for Object State Estimation

Object motion estimation is a core capability of autonomous robots. One solution to this problem can be achieved with a single camera by fusing image track data from a single feature with camera motion measurements. Such a system takes advantage of sensors already expected on a human-guided robot (motion sensors that enable navigation and cameras that provide situational awareness) and therefore requires little additional payload. Furthermore, a single camera solution adds redundancy and fault tolerance to current systems that use multiple cameras, combinations of cameras and scanning lasers, or stored environment models [3] [7].

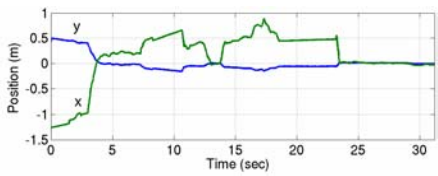
Object motion estimation using bearing sensors such as monocular vision has been well studied [1] [8]. The key features of this problem are that the object position and velocity are unobservable at any instant in time and that performance of the estimator is a strong function of the camera trajectory. Exploiting this dependence, Frew [4] developed a new trajectory design method that maximizes the information content provided to the estimation filter. The three main issues addressed by this new method are inclusion of the monocular vision field of view constraints, the quick generation of near-optimal trajectories, and the dependence of the optimal solution on the uncertain object state – the very parameter being estimated.

The new trajectory design method uses a pyramid, breadth-first search to generate paths in real-time that achieve a minimum estimate uncertainty bound in fixed time or a desired uncertainty bound in minimum time. The robot trajectories are parameterized as a series of constant-heading, fixed-speed maneuvers. By balancing the trade offs between several design parameters, including the number of maneuvers, size of the discretized heading space, and number of iterations, the new method creates trajectories that achieve near-optimal performance at greatly reduced planning cost and time. Additionally, the design method continually updates the trajectory as new data becomes available and the object estimate converges.

Provided is an example of a simulation that uses the new design method to estimate an object moving with a constant velocity of (0.0, -0.015) m/s. The method is called using the fixed-time minimum-uncertainty cost. The robot is allowed five, 6.0 second maneuvers, the heading space is discretized into five intervals, and the pyramid iteration is performed four times. The resulting observer path is shown in Figure 4a) along with the true target path and the evolution of the target estimate over the course of the run. Figure 4b) shows the position errors as a function of time. As expected the error in the y-dimension stays small throughout the run while the error in the x-dimension, which corresponds closely to the object range, takes more time to converge.



a)



b)

Figure 4: Example of a simulation that implements the new trajectory design method. Figure a) illustrates the resultant robot trajectory while b) displays the target position error as a function of time.

3. MARS Test Platform

The Micro-Autonomous Rovers test platform provides a two-dimensional workspace for researching autonomous rovers. The platform consists of a 12' x 9' granite table with six autonomous robots that move about the table's surface. Each robot has its own planner located off-board. Control signal processing is also done off-board, and the control signals are sent to the individual robots via a wireless RC signal. An overhead vision system provides position sensing using three cameras to detect LED's mounted on the top surface of robots.

All communication within the MARS platform is accomplished with Real Time Innovation's Network Data Delivery Service software, which works on a publish/subscribe architecture. Hence every node on the network can send and receive different data types. The flow of this data is illustrated in Figure 5.

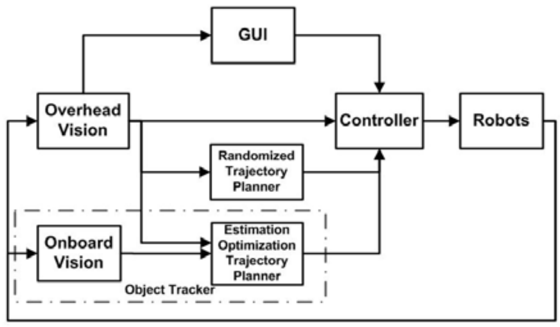


Figure 5: Data Flow in the MARS test platform.

To achieve on-board vision, a wireless camera is mounted on the top of one MARS rover looking downward towards the ground. The unit is commercially available and is integrated with a 2.4 GHz wireless video transmitter. Objects are outfitted with an LED to facilitate tracking.

4. Integrated System

To integrate the three projects into one system, several technical issues were resolved by way of the following system modifications:

-New dialogue capabilities were added to the GUI to allow a user to access the new functionalities.

-New display symbology was added to the GUI to provide users with the necessary information for the increased functionality.

-The tracking robot mounted with the camera was given highest priority for motion planning. This forced all other robots to construct their trajectories around it.

-A common communication protocol between all components was established.

-A central clock enabling time synchronization across the robots was established.

The following experiment shows the integration of the three research efforts. The GUI is used to operate multiple robots that use their planners to avoid collisions and locate targets.

At the start of the experiment, the robot at the bottom of the picture is first commanded to follow the robot with the camera on the left, (see Figure 5a). It will try to maintain a distance of thirty centimeters from the camera robot. Next, the camera robot detects the target, in this case the robot at the top. In Figure 5b), it is commanded to reduce the uncertainty of the location estimate. The last robot is tasked to move across the table as shown in Figure 5c). The trajectory constructed by the planner is denoted by the curved red line. This motion will bring it in front of and in conflict with the camera robot. In Figure 5d), this last robot senses a conflict with the camera robot and re-plans its path. This new path is created in real time and there is no noticeable discontinuity in the robot's motion.

Ultimately, the graphical user interface was used to command three robots to achieve useful tasks. The target in the environment was located correctly, this task was successfully monitored by the first robot, and the third robot was able to achieve its position goal while maneuvering around the other objects in the environment.

5. Conclusion

Three tasks fundamental to multi-robot systems were enabled. Specifically, the tasks allow the interfacing of a single user with multiple robots, motion planning for multiple robots, and robot trajectory generation for target tracking. It was shown that the tasks could be merged into an experimental demonstration that highlights their usefulness as components in a larger, more complicated system. The result is a cooperative multi-robot system demonstration where a single user can control robots in a group to carry out individual tasks including collision-free motion and target tracking.

6. References

[1] V. J. Aidala, "Kalman Filter Behavior in Bearings-Only Tracking Applications", IEEE Transactions on Aerospace and Electronic Systems AES 15, No. 1, 29-39, Jan 1979.

[2] C. M. Clark, T. Bretl & S. M. Rock, ``Applying Kinodynamic Randomized Motion Planning with a Dynamic Priority System to Multi-Robot Systems," IEEE Aerospace Conference, 2002.

[3] M. W. M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, & M. Csorba, "A Solution to the Simultanewous Localization and Map Building (SLAM) Problem", IEEE Transactions on Robotics and Automation, June 2001.

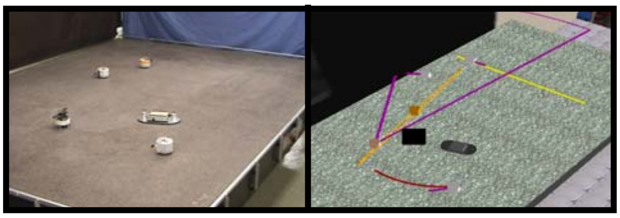
[4] E. W. Frew & S. M. Rock, ``Trajectory Generation for Monocular Vision-Based Tracking of a Constant Velocity Target," To be presented at the IEEE International Conference on Robotics and Automation, 2003.

[5] D. Hsu, R. Kindel, J. C. Latombe, & S. Rock, "Randomized Kinodynamic Motion Planning with Moving Obstacles," Workshop on the Algorithmic Foundations of Robotics, 2000.

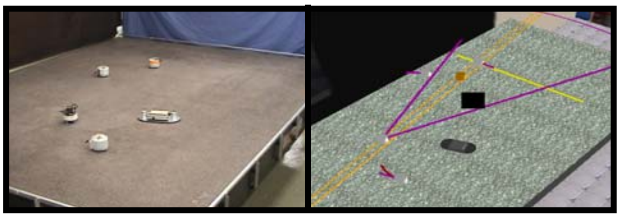
[6] H. L. Jones & S. M. Rock, ``Dialogue-Based Human-Robot Interaction for Space Construction Teams," IEEE Aerospace Conference, 2002.

[7] T. Kanade, R. T. Collins, A. J. Liption, P. Burn, & L. Wixson, "Advances in Cooperative Multi-Sensor Video Surveillance,", DARPA Image Understanding Workshop (IUW), 115-122, Nov. 1998.

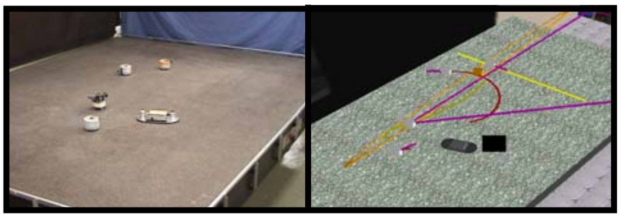
[8] A. Logothetis, A. Isaksson, and R. J. Evans, "Comparison of Suboptimal Strategies for Optimal Own-Ship Maneuvers in Bearings-Only Tracking", Proceedings of the American Control Conference, 3334-3338, June 1998.



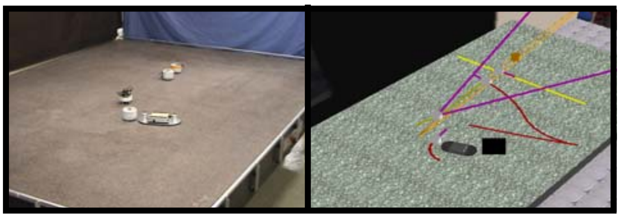
a)



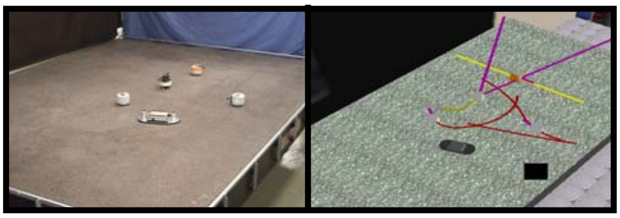
b)



c)



d)



e)

Figure 5: Hardware is displayed on the left, while the GUI is displayed on the right. Within the GUI, robots are denoted by white cylinders. The target trajectory is denoted by the yellow line in the upper right corner of the screen shot. Red curves denote robot trajectories constructed by the motion planner. The scope of the camera onboard the tracking robot is outlined by two purple lines.