Decentralized Autonomous Rovers

*Freedom Rover Units

1st Jordy A. Larrea Rodriguez
Department of Electrical and Computer Engineering
University of Utah
Salt Lake City, USA
Jordy.larrearodriguez@gmail.com

2nd Brittney L. Morales
Department of Electrical and Computer Engineering
University of Utah
Salt Lake City, USA
brittneymrls@gmail.com

3rd Misael Nava
Department of Electrical and Computer Engineering
University of Utah
Salt Lake City, USA
misaelnava812@gmail.com

Abstract—The state of the art in autonomous swarms employs a decentralized model consisting of multi-agent networks. These multi-agent systems hold the potential to adapt to new environments and optimize individual performance to specific tasks without having to deal with global systems prone to single points of failure. Our team's focus lies therein in developing a decentralized multi-agent system. The decentralized multi-agent system will incorporate three to five two-wheel drive rovers interfaced through the Robot Operating System (ROS) by raspberry pi 4 companion computers for edge-computing specifications. A central system is still beneficial for the assignment of global objectives; thus, our proposed decentralized system will employ a central bay station to communicate objectives for the agents to complete (carefully designed demos). LiDAR, simple infrared sensors, and low-resolution cameras will be procured to facilitate real-time (RT) decision-making by the agent(s) in response to the environment. Our development stack will leverage ROS for project management, simulation capabilities, navigation libraries, and native server-client model in robotics applications. The rovers AI will incorporate simultaneous localization and mapping (SLAM) techniques for RT positioning based on a priori grid or map; thus, facilitating navigation through improved state space mapping. Decentralized swarms allow for a set of agents to act as more than distributed actuators: i.e., more like your white blood cells rather than your limb as observed by central systems. Furthermore, drone Swarms hold the potential to gather large quantities of data for monitoring and area mapping that would otherwise prove too costly to collect, and can greatly simplify and reduce manpower in search-and-rescue, disaster recovery, or security scenarios. Swarms essentially hold the capability to replace humans in potentially dangerous and normally costly tasks.

Index Terms—Decentralized Communication, Swarm Communication, Multi-agent, ROS, Gazebo

I. Introduction

This document is a model and instructions for LaTeX. Please observe the conference page limits.

II. Background

A. Scope of Project and Design

The scope/goal of the project is to create a small group(our swarm) of semi-decentralized rovers that achieve a common goal. Our swarms will consist of an aim of 3+ rovers which although does not constitute an ideal amount for a swarm, however this will be enough for a proof of concept for the future that will use a larger group of individuals. As for the common goal, the team of robots should be able to work together to put on a little synchronized dance performance or to push a block from point A to another point in space. While the concept of these rovers being decentralized means that they are able to act independently with assistance from a central leader. This now leads into the question of how these autonomous rovers are going to be made and how they will be aware of their environment. There are multiple ways we can get the rovers to perceive their surroundings, in the case of the project there are three options. One option is to use a 2D lidar sensor. The lidar will give the robot an image of the objects currently around it and the distance from the unit itself. With the implementation of lidar it will open up the avenue to use SLAM(Simultaneous Localization and Mapping). For the actual build of the rovers the plan is to use a premade chassis for the outer shell of the robot. It will also use two stepper motors and a caster wheel to enable movement. As for powering/controlling the system, it will use a custom PCB power module for powering the different components with another custom PCB motor driver to control the stepper motors. As mentioned the chassis are premade, meaning the idea of making a rover swarm is not original.

III. Methods and Planning

A. Initial Timeline and Project Tasks

Here is the initial timeline for the project broken down into individual tasks:

- May: Assemble Rover/PCB

- Main goal is to make the custom PCB for power module and motor drivers
- Buy parts: Motor, Chassis, LiPo Battery, development boards, environment sensing parts
- Test PCB and parts to make sure they are functional and understand how to interface with them
- June: Setup ROS/Code Framework
  - Use ROS to create initial code for the multi-agent server to assist Rovers
  - Also use ROS to manipulate and control motors to allow Rover to move
  - Begin interfacing with sensors and other components
- July: Setup ability to see environment
  - Setup Lidar/Computer Vision
  - Get Rovers to build map of their environment SLAM
  - Ensure rovers can move through environment without colliding
- August: Get Rover to move in consistent manner
  - Finish up main part of Rover framework
  - Get the Rovers to move in a set pattern
- September: Setup central server/Test Communication
  - Finish creating ROS multi-agen server
  - Get the server to communicate with Rovers
  - Test reaction time between server and Rovers
- October: Rover will work together on a task
  - Get Rover to push box
  - Get Rovers to dance around
  - Or get Rovers to complete another demo that is list later
- November/December: Debug and finish up Project
  - These are free months to use to buffer if the project falls behind or need extra time to debug

B. Project Resources

In order to finish the project on time we will need to deploy the use of external tools to assist in its completion. Luckily there are many tools at disposable to get the rovers on their wheels and working together. The first and biggest resource the project will be using is the Robot Operating System, ROS for short. ROS is a tool that is made to interface well with robotics, offering a multitude of tools to help simplify the process. These tools range from helping enable the rovers to use SLAM to even PID control for controlling the motors. Raspberry Pi and ESP32 can be flashed with versions of ROS. With the assistance of ROS implementing control/deploying the rovers will be simplified to focus on other aspects of the project. Although the use of ROS enables an easier deployment of the tools needed for the project, simulations are an integral part of ensuring that the rovers operate as intended. To create simulations of the swarm, Gazebo will address this issue. Gazebo is 3D simulation software that integrates well with ROS. However, simulations of how the rover moves is only one part of the testing process.

IV. DISCUSSION

Testing will be done regularly from the first-day our components arrive to assure quality and functionality. As we assemble each component together, testing will help in debugging and localizing the issue in the system. Creating tests for each component helps with creating a good understanding on how that device individually works as well as how it interacts with other devices and software. On the topic of software, testing is of critical importance, without testing the software integration could lead to unforeseen behavior within the system as a whole. If our tests were done correctly, on demo day it will present our flawless creation, the freedom rover units to the world.

V. DEMO DESCRIPTION

Our goal is to demonstrate one of these three demos on demo day. The First demo shows each drone/rover coordinating with one another to align themself to make a letter. An example of this demo: from the server we give the task of creating the letter 'L', with 5 drones scattered on the map, each rover will represent a pixel and will move to that spot. If there were two rovers that are the same distance away from a spot, one rover will have higher priority over the other and the rover with the higher priority will get that spot. As for the other rover it will go to the next closest open spot. Another option for the First demo, is that each rover will have multiple different LED lights to make a light show on the ground.

The Second demo is to have the rovers coordinate with one another to move an object from one location to another. An example: 3 rovers scattered around the map. The map is 5' by 5', where (0,0) is located at the bottom left corner and (5,5) is located at the top right corner. Each rover get the task to move a cardboard box from (1,1) to (4,3). All 3 rovers will communicate and align themselves around the box, to move it to the said location. Once done, they will wait for their next instruction which may be to move it to a different location.

As for our last demo, the Third demo shows drones moving in a line around the map. For example: there are 2 tasks, the first task is to follow behind another drone and the second task is to move around the map. The second task will be given to one rover only and the first task will be given to all the other drones on the map. In this demo, there are 5 drones scattered throughout map and drone named #2 will get the second task. As for the other Drones: #1,#3,#4,#5, they will get the first task. Let's say the order of drone conga line is : #2,#3,#1,#5,#4. If we were to separate #1 from the conga line and place it elsewhere on the map, it will find its way behind the last rover in the congo line which in this case is #4. Thus conga line will look like: #2,#3,#5,#4,#1.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]"

or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.