Heterogenous Autonomous Robotic Exploration (HARE)

Jackson Parker, Allen Spain, Caleb Adams

***Abstract*—The purpose of this research is to prove that subtask division can be accomplished based on knowledge of robot capabilities. To prove this, we are developing an exploration system in which robots with different hardware and capabilities will work together to accurately explore and map a complex region of space. Through the definition of individual attributes and the sharing of those attributes, each robot will search and map regions of space that it is capable of maneuvering. Instead of simple binary mapping of the region, 1 = free space and 0 = obstacle, each robot will mark the region that it has explored and can maneuver through this a specific identifier. This gives the region itself a bit of information as well as allows other robots to help determine if they are capable of maneuvering as well. Once a robot comes to an obstacle or region that it cannot maneuver around, it will evaluate which team members are capable of doing so and mark that space for the capable robots to explore. Due to time limitations on this project, instead of relying on complex perceptive capabilities to evaluate which robots could explore that region, the utilization of object id which can be surveyed by the robots sensors to compare its own attributes with the attributes necessary for exploration.**

Figure 1: The wild hare our team mascot.

***Index Terms*—IEEE, IEEEtran, journal, $\LaTeX$, paper, template, Multi-Robot Systems, exploration, autonmous mobile robots, heterogenous, BFS, behavior trees**

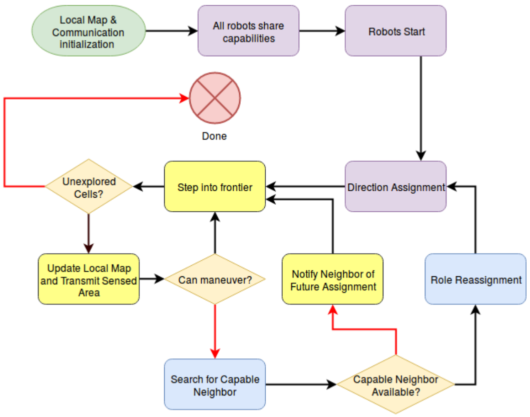

Local Map & Communication initialization
All robots share capabilities
Robots Start
Done
Unexplored Cells?
Step into frontier
Direction Assignment
Update Local Map and Transmit Sensed Area
Can maneuver?
Notify Neighbor of Future Assignment
Role Reassignment
Search for Capable Neighbor
Capable Neighbor Available?

Figure 2: High level block diagram for HARE

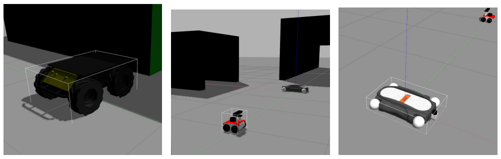
I. Introduction

A. Overview

As an individual working in a cooperative operation, knowing the capabilities of you and your teammates is necessary to the division of tasks. This idea provides the basis for this proposed research. As homogeneous multi-robot systems are inherently limited based on the capabilities of the specific robot, it makes sense to venture down a heterogeneous avenue. A lot of research has gone into heterogeneous multi-robot systems in the past few years, but a common trend amongst most of them is that the model is built for the specific platforms that are to be used. This proposed research is meant to provide the groundwork for a heterogeneous model that allows for the insertion of a robots with different skill sets with the purpose of enhancing the overall capability of the system. This framework would allow for a variety of models to be built on top of it, with a variety of hardware. Researchers wanting to build a specific system could allocate funds and resources to build each member to accomplish a portion of the overall task, possibly allowing for more cost efficiency and remove unnecessary redundancy. The main challenge in this research will be finding the optimal way to reassign tasks based on defined capabilities of each robot in the system. This brings up a secondary, more basic, challenge of defining robot's characteristics in a simple yet comprehensive manor. All-in-all, this research acts as a proof of concept for task division based on member attributes, or individual robots capabilities, and will allow for a high degree of heterogeneous utilization and multi-robot model expansion.

B. Background

The basis of this research is in the definition of attributes which is detailed in the methodology section. This section is meant to detail previous research that will be used in proving that the fundamental idea of task division based on individual capabilities. As this research is meant to strictly provide a basis for heterogeneous task division and capability utilization, the model built on top of it could take many forms. In this case, a multi-robot behavior tree will be constructed to guide operations and define command sequences at a high level like proposed in [3] which is an expansion of research in [2]. To implement such a system the DSL buzz [6] will be used which provides a level of abstraction that allows for focus on high level cooperation, information sharing, and simple behavioral definitions. The researchers that developed this language have created a ROS package that will be utilized, called ROS-buzz, in this proof-of-concept system. Control mechanisms and communication in multirobot systems have been a center of research for a while and updates/innovation in this corner of the field are practically constant. The Robot Operating System (ROS) [1] provides a large variety of packages that can be put together to form fully functioning and unique control mechanisms as well as providing easy to use communication packages. Exploration and path finding are widely researched and implemented using ROS packages, as components are pretty broad and can help illuminate edge cases in multi-robot systems as agents have to operate without global knowledge of their environment.



(a) Husky (b) Rosbot (c) Youbot

Figure 3: Three commonly used robots with very different shapes and sizes.

II. Methodology

A. Role Assignment

As aforementioned role assignment was predefined for the robot set. In the project three robots were given static characteristics, these characteristics were defined based on robot attributes. The simplified set of attributes taken into consideration were maneuverability and robot height. These attributes corresponded to terrain traversability, the terrain was indexed as follows: 0 - open terrain, 1 - ramp, 2 - short tunnels, and 3 - tall tunnels. Given the set of robots and attributes capabilities were defined. Robot 1 (Youbot) could traverse terrain 0, 2, and 3. Robot 2 (Rosbot) could traverse terrain 0, 1, and 3. Robot 3 (Husky) could traverse terrain all terrains in the terrain set, the robots and terrain is illustrated in figure 4.

As an individual working in a cooperative operation, knowing the capabilities of you and your teammates is necessary to the division of tasks. This idea provides the basis for this proposed research. As homogeneous multi-robot systems are inherently limited based on the capabilities of the specific robot, it makes sense to venture down a heterogeneous avenue. A lot of research has gone into heterogeneous multi-robot systems in the past few years, but a common trend amongst most of them is that the model is built for the specific platforms that are to be used. This proposed research is meant to provide the groundwork for a heterogeneous model that allows for the insertion of a robots with different skill sets with the purpose of enhancing the overall capability of the system. This framework would allow for a variety of models to be built on top of it, with a variety of hardware. Researchers wanting to build a specific system could allocate funds and resources to build each member to accomplish a portion of the overall task, possibly allowing for more cost efficiency and remove unnecessary redundancy. The main challenge in this research will be finding the optimal way to reassign tasks based on defined capabilities of each robot in the system. This brings up a secondary, more basic, challenge of defining robot's characteristics in a simple yet comprehensive manor. All-in-all, this research acts as a proof of concept for task division based on member attributes, or individual robots capabilities, and will allow for a high degree of heterogeneous utilization and multi-robot model expansion.

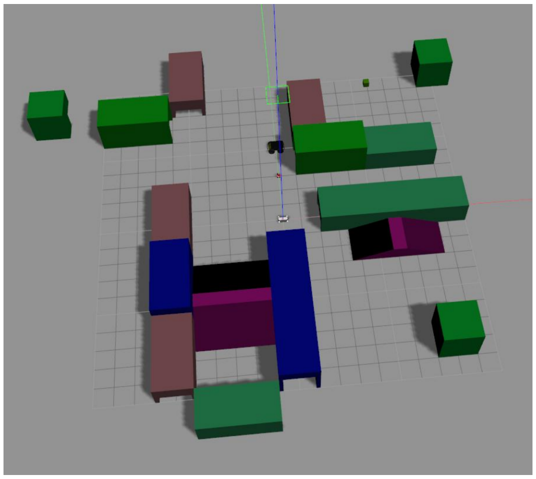
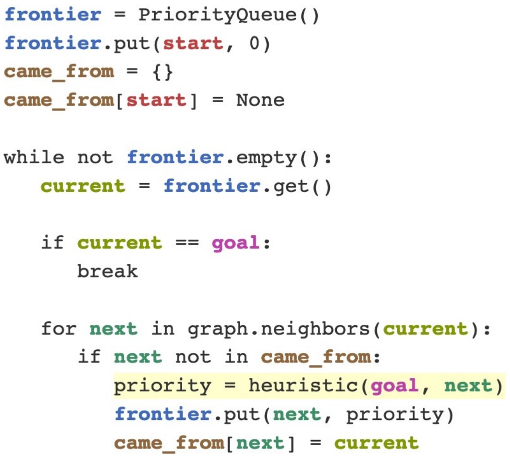


Figure 4: Gazebo map, with different terrain. Open terrain in gray, short tunnels in red and green, tall tunnels in blue.

B. Algorithms

For navigation we used two methods. The first is a Breadth First Search (BFS) which explores the cells adjacent to the robots current positions, before exploring the second level of adjacent cells. After determinining explorable cells method getPath as described in Algorithm 1 will compute the a hueristic based on the manhattan distance to determine a set of cells that make up an optimal path. For robot exploration getPath can be called for exploration by simpling passing in a goal represented as the nearest unexplored node a summary of the path planning algorithm can be called with the goal being the nearest global unexplored node as shown in the algorithm in figure 5. The use cases for this method are as follows: a) robot needs to swap locations, b) robot finds an obstacle it cannot explore and needs another robot to explore it 3) the robot's tree-state is such that exploration needs to take place.



```
frontier = PriorityQueue()
frontier.put(start, 0)
came_from = {}
came_from[start] = None

while not frontier.empty():
   current = frontier.get()

   if current == goal:
      break

   for next in graph.neighbors(current):
      if next not in came_from:
         priority = heuristic(goal, next)
         frontier.put(next, priority)
         came_from[next] = current
```

Figure 5: An overview of the A* Algorithm [2]

C. Multi-Agent Testing Framework

After exploring a set of options to test this idea, the best place to go was to the drawing board. Taking a step back it was decided that pushing to get straight into hardware testing was a poor decision and the underlying concept of heterogeneous task reallocation based on capabilities needed to be tested at the lowest of levels. To do this, the creation of a testing framework in roscpp was necessary. This framework allows for easy addition of robots by simply adding the details for the new robot in the xml robots launch file. To add any additional information yaml files can be used to set rosparams, just as was done here for robot capabilties. By configuring the launch file with the proper robots, one can edit the one robot launch file to have each robot launch any set of nodes that would exist in each robot's namespace. This allows for testing of a completely decentralized algorithm in a centralized system. The reason this was done was because of the difficulty of getting a multimaster architecture working within gazebo.

III. Results

A. Exploration

Our results were limited to test run outside of gazebo. We were able to test the breath first search algorithm which provided an obtacle free path to a defined goal. The results of this test are shown in figure 6. We also were able to test the robots motion model, the results of these tests revealed that the youbot was an ideal candidate since it was orientation agnostics and was fully holonomic. The one limitation to the youbot platform was its ability to traverse ramps, this was factored in when designing static characteristics as described in the methodology section. The motion model for the husky and the rosbot were far from ideal due to path deflection and turn radius.

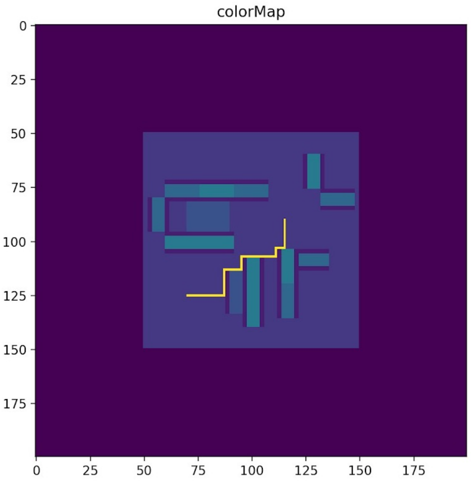


Figure 6: A* python path planning simulation

IV. Conclusion

A. Discussion

This project served as a foundation to enable future research and testing. This framework is going to be used more as a testing environment for future work such as multiagent micromouse simulation in a realistic 3d environment allowing the user to easily test their Algorithms, with sensing and realistic robot maneuvering. Before this system is launched we will provide a more robust steering mechanissdm which will take into account the robot model, this will abstract out the difficulty associated with navigating to a goal or following a line. In addition to abstracting out the motion models, this framework is designed to support complex multiagent decision making like those represented in behavioral trees.

robots to share their information states including the enountered terrain in order for the robot team to best explore and traverse a given heterogenous environment. The resultant decision that is derived from an agents neighbor's states namely TREE STATE, this variable represents the robots decision, which could entail further investigation (PROD) or stopping all motion and entering into a low power mode (IDLE).

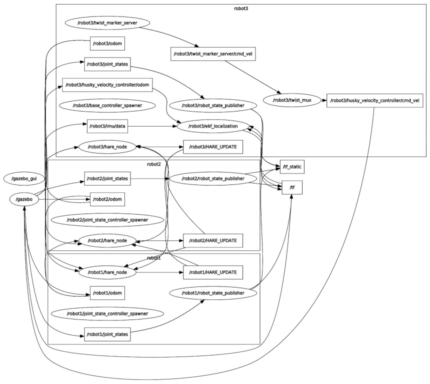

robot3
/robot3/twist_marker_server
/robot3/odom
/robot3/twist_marker_server/cmd_vel
/robot3/joint_states
/robot3/husky_velocity_controller/odom
/robot3/twist_mux
/robot3/husky_velocity_controller/cmd_vel
/robot3/base_controller_spawner
/robot3/robot_state_publisher
/robot3/imu/data
/robot3/ekf_localization
/robot3/hare_node
/robot3/HARE_UPDATE
robot2
/tf_static
/gazebo_gui
/robot2/joint_states
/robot2/robot_state_publisher
/tf
/gazebo
/robot2/odom
/robot2/joint_state_controller_spawner
/robot2/hare_node
/robot2/HARE_UPDATE
robot1
/robot1/hare_node
/robot1/HARE_UPDATE
/robot1/odom
/robot1/robot_state_publisher
/robot1/joint_state_controller_spawner
/robot1/joint_states

Figure 7: ROS multi-master communication architecture

B. Future Work

Even though the testing framework was not the overall goal of this research, it turned out to be one of the strong suits of the project. Future work would certainly include iteration on this framework to ensure that it is as easy as possible to randomly generate maps as well as add any number of any kind of robot. The benefit of doing this on Gazebo is the ability to emulate robotic agents much closer to the real world representation than Unity, ARGoS, or any other open source platform capable of multi-robot simulation. Another step would ofcourse be iteration of the tested cooperation model on hardware. This would require much more extraneous simulation to ensure that all edge cases are covered. To properly do this, the testing framework would have to be iterated as described above. Due to Gazebo not inherently supporting multimaster network architectures, there is the possibility that a plugin or new simulation package would need to be developed [3]. The communication architecture with all nodes encapsulated within the robot namsepace this communication archietecture is shown in figure 7.

References

[1] J. M. O'Kane, "A gentle introduction to ros," 2018.

[2] "Introduction to the a* algorithm," *Red Blob Games*, 2014.

[3] A. Tiderko, "multimaster fkie," *ROS wiki*, 2015.