Robot Collision Avoidance

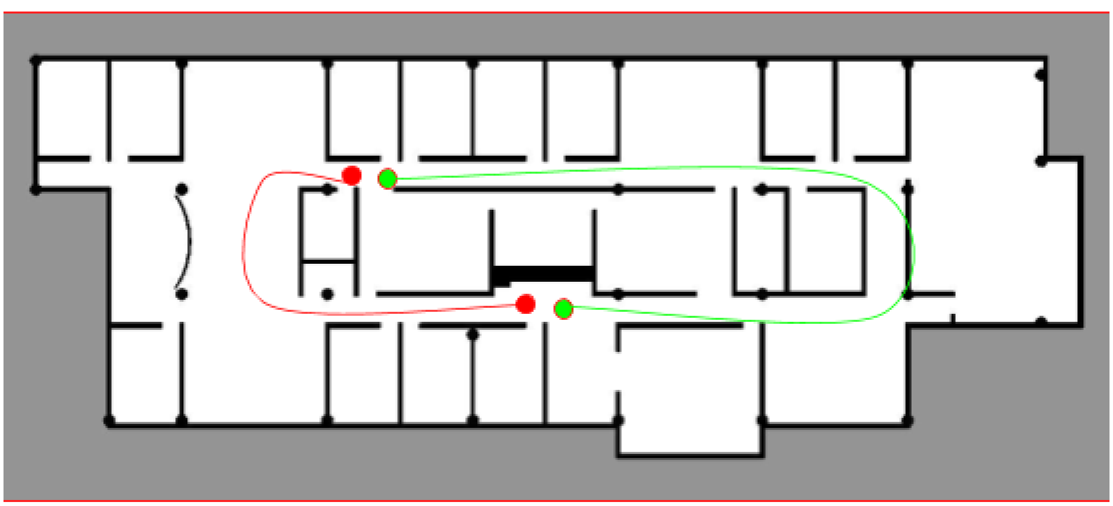
Brigette Krause and Madeleine Williams

I. Abstract

The mission of the Building Wide Intelligence Project is to allow fully autonomous robots to eventually be integrated into the Gates Dell Complex's everyday environment. Currently, with three operational robots working under the project, there are instances where two robots cross paths, and identify each other as obstacles. This leads to the two robots replanning their course, resulting in lost time and, in certain locations within the Gates Dell Complex, repeated occurrences where the robots are forced to plan a new path when it is unnecessary. The goal of our project was to design a ROS node that would allow for one robot to move out of the way of the other. This research paper outlines our approach to the task, and discusses the technical aspects of our solution.

II. Introduction

Currently the lab BWI segbot machines are unaware of how to detect other currently active robots as machines and continuously mistake them for obstacles. By mistaking robots for obstacles, all bots that come across each other remain in a deadlock state; this deadlock state is recognized by robots replanning based on the "obstacles" that reside in their current path to the goal and, depending on the building's structure such as the square structured hallways in the GDC, coming across each other every time one attempts to find another route to its goal. By being stuck in this loop, the robot runs the risk of wasting time and never accomplishing its desired goal unless an outside factor impacts one of the colliding robots.



In order to avoid the situation of continuously encountering the other segbots, we attempt to maneuver a robot from its current path and into an open area in the map so as to avoid colliding with other active segbots utilizing the same pathway. To do this, each robot must have access to other active BWI machines' coordinates as well as update their own coordinates constantly to create an active layout of where each bot resides in the map. Currently, through CMASS, information about the segway bots, including x and y coordinates, are displayed on a website and available for users to incorporate in their own projects [1] . By having access to this multi-robot server, it will be easier for the segbots to keep track of others on the same floor and process how close they are to having a collision. Even though each robot only knows of the environment they're currently exploring, it is vital to this project that, though the robot does not need to know every detail of the environment surrounding other bots, it must know the location of a bot in the same environment.

From our project, we hope to be able to incorporate the information displayed on the nixons-head website in order to instruct a bot to maneuver out of its current path if any of its coordinates align with another active bot's coordinates. This project will allow robots to autonomously operate themselves and complete tasks without having to worry about being in a state of deadlock. Through this we hope that users will no longer encounter a situation where they have to interfere with a robot in order to stop it from colliding with others and will embrace working with multiple bots on a small layout, such as the GDC third floor lab.

III. Background and/or related work

Concerning the concept of our project, we turned towards previous works that contained similar elements and ideas that furthered our approach. Though we aren't focused on the planning of the bots and the instance where one machine needs the assistance of another, other papers that have worked on this methodology have described algorithms to smaller problems in our overall project. Researches have focused on the "ant algorithm" where tasks are chosen by other robots based on the needed assistance of others. However, this project demonstrates robots in a closed container and has them rotate until the path in front of them is clear if they happen to interact with other machines [2] .The approach of having the segbots turn until their pathway was clear of any obstacles introduced the idea of having the bots turn when they come across each other. Though this approach aids the robot if working in a small, closed room, the robots would still be perceiving each other as obstacles and merely avoid them rather than detect each other as bots. This would cause the robots to take part in an action that is similar to the current bot's replanning system and would need to be altered in order to work in the desired form.

In order to solve the situation of robots detecting each other, robots must take into consideration the environment of other bots when calculating movement. Thus, in this project we considered the various aspects and algorithms that researchers took into consideration in a program that implemented a "M+ algorithm" [3] . Our project took a similar approach in the aspect that the M+ algorithm focuses on updating a multi-robot server that each active bot has access too and reads from. Though we only focus on the coordinates of the bots, the robot will continuously update its coordinates as well as keep track of every other segway bots' position. Similar to the multi-server used, the implementation of the website updated by CMASS allows for each machine to depict when it is necessary to relocate in order to improve efficiency of completing a set task. This technique allows for more efficiency in the sense that robots will only need to keep track of location in the overall environment rather than details and situations in other bots' individual environment.

When facing the situation of how a robot should navigate and cooperate with other robots on the same path or avoiding collision, many researchers have taken approaches of the robots giving right away to those with higher task priority and are based on local sections of the maps [4] while other plans rely on both local and global maps in order to check neighboring areas to avoid more probable collisions [5] . Both global and local planning are essential in accomplishing goals given to multiple AIs residing on the same floor, however, global planning is not utilized in this project due to not attempting to resolve path collisions from the beginning of each task assigned because of the potential of coming across undocumented obstacles or outside factors, such as speed and human interaction, delaying the robot's position. Overall, each paper focuses on including system-wide connectivity for the robots through the form of a server.

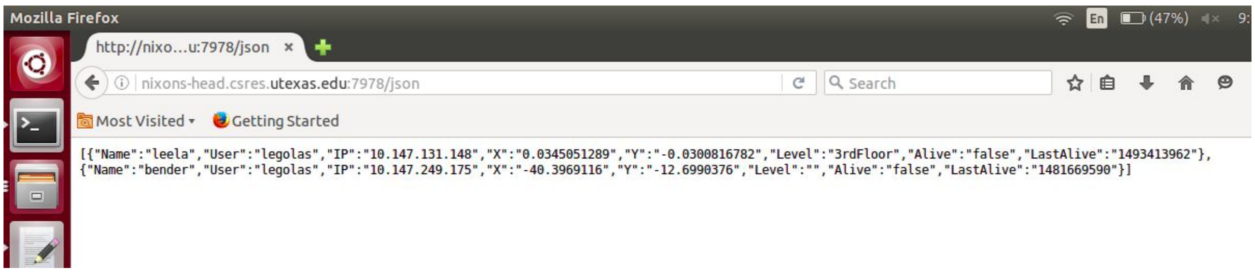
Another approach to the problem of robot cooperation and robot collision avoidance is an incremental plan-merging technique. Instead of having a centralized server to keep track of information, including coordinates, goals, and plans from every operational robot, each robot undergoes a "Plan-Merging Operation". Through this technique, every robot coordinates its plan around the plans that have already been prepared by other robots. If no possible path can be coordinated to ensure that it does not interfere with others, then a robot will wait at a stationary position until another has completed its task or has changed its route [6]. This method was tested in the MARTHA Project, which employed multiple autonomous robots in transshipment tasks. The endeavor was successful, with each robot able to execute different tasks that were given with little influence from a central planning station [7]. Although we do not coordinate plans between the segbots in the BWI lab, plan-merging could prove to be inspiration for future projects. Another technique that also could possibly be incorporated to one day achieve this is having one robot assigned to the role as a leader and all other robots assigned to the role of followers, as proposed in previous research. Similar to the design of our project and other research projects, there is little need for an external central station to plan out every aspect of robot coordination [8]. Despite using a multi-robot server to access the locations of each robot on the map, the robot currently running our node decides on its own the location to move to once it detects another robot in the vicinity.

Overall, many other researchers have considered the problem of multi-robot cooperation and have created algorithms in order to deal with these projects. From these various sources, we have broken the project into sets of smaller problems in which we focus on local planning and checking neighboring areas rather than searching the entire grid for other active bots. Furthermore, we also attempt at utilizing a multi-robot server in order to process the necessary information that would aid us in locating all BWI bots and relocating a robot if it has the potential of colliding with another machine.

IV. Technical Approach

In order to achieve the goal of having one robot move out of the way of another, we wrote a ROS node that would change the location of the robot that was running the node. A stationary robot was used to test the node, and when the coordinates of a mobile robot entered within a certain range of the coordinates of the stationary one, the stationary robot would then move forward until it is able to move to the side and into a location that is no longer in the path of the mobile robot.

In order to obtain each robot's location, we used the JSON (JavaScript Object Notification) file from the Centralized Multi-Agent Status Server (CMASS) as shown in the image below. Once the JSON file is accessed through curling from a URL (Uniform Resource Locator) that contains the information that relates to the active and inactive robots that operate



under the BWI Project, it is then parsed in order to access the x-coordinates and the y-coordinates of each robot. The coordinates are updated from the JSON file every few seconds to ensure that the two active robot's locations are accurate. The most difficult part of this section was converting the x-coordinates and y-coordinates stored within the JSON to usable values that the robot could interpret as positions.

Once the coordinates of the two active robots have been received, the stationary robot checks if the mobile robot is in the surrounding area. If the mobile robot is within a certain distance, then our node prompts the stationary robot to continually change its x and y positions until it is no longer in the same general location as the mobile one. This ensures the robot is not in the path of the other one that is currently moving, thus preventing unnecessary replanning by the robots. Once the stationary robot that is running our node is safely out of the way, the mobile robot is able to continue with its assigned task unimpeded.

V. Evaluation

When running the code and various implementations on the robot, the machine succeeded in achieving the initial goal. Due to curling from the website and parsing the JSON file continuously, each bot is able to keep track of the coordinates of every active robot on the same floor. Though the code is able to update coordinates, the site itself was not updating positions during the given testing phase and presented problems with the experimental phase. However, when working with the given set of coordinates, the current BWI bot was able to perceive another robot on a similar axis and was able to change x and/or y coordinates until its path no longer aligned with another active BWI bot. The image displayed below demonstrates the robot's movements when attempting to relocate to a safer location.



Though the code allowed the BWI bots to succeed in accomplishing their goals, there were instances where the robot would travel closer to the robot in an attempt to find a safe place to relocate. With a stationary robot this would be fine, but as an oncoming robot would approach the robot ran into a situation where it would have a higher probability of colliding before finding a location outside of the merging paths.

VI. Limitations

When considering limitation factors in the project, there were a few that caused a delay in testing and recording progress as well as determining whether the code contained bugs or outside factors were affecting results.

- *Unable to update positions correctly:* During a few trial runs towards the end of the experiments, the robot was unable to receive updated coordinates. From this the robot would not be able to detect that there was an incoming segbot in its path nor would it be able to come to the realization that it had successfully relocated and was no longer disrupting the path of another machine. Though it only happened during the last few phases of testing, this type of problem could occur in the future and prevent robots from determining robot's positions relative to their own.
- *Algorithm to how the robot found a proper location:* From testing various routes for the robot to calculate and figure out how to relocate to a given position, we came to a solution that, in few instances, created more of a problem when it concerned avoiding other bots.The map of the 3rd floor GDC provides a view of various spaces in a pathway that the robot could venture to if located in the current area, but being able to detect this are within a given time was not efficient. Fortunately, the only other active robot was stationary due to being tested on and allowed our robot enough time to locate an area and then venture towards its new goal. However, if the robot is tested with moving bots, it might run into the problem of not being able to relocate and approach an area before the other bot reaches a close enough distance.
- *Only seems to work with two robots and must be properly localized:* With the current way that the robot is accessing and keeping track of its own coordinates as well as active segbots, the code is successful if the information displayed on the website includes the current robot's position and one other segbot. This is not a fault of the website, but in the way in which the robot does not have a method of comparing to see if it's in the correct position as the coordinates its maintaining until it is properly localized. Since the robot

can only currently be tested with one other active segway bot, this limitation has not been fully tested and cannot be tested until a third bwi robot is placed in the environment. Furthermore, the robot relies on the generated map of the floor it is residing on and must be properly localized in order to determine its own location and proper hiding areas. Without proper placement, the robot could possibly avoid potential open spaces because of believing it is beside obstacles rather than a potential pathway.

VII. Conclusion and future work

In conclusion, our initial goal was met. A stationary robot was able to successfully move to a location that was out of the way of a mobile one. However, improvements can be made to the project. Accessing the static map of the current floor the two robots are on, as well using the occupancy grid associated with that current map, would be a possible refinement of our project that would result in more efficient results. Our stationary robot would then be able to randomly sample points to move to within a certain range, check if a location is occupied, and then move to that location if it able to be reached. This approach to the task of ensuring two robots do not interfere with each other's path would possibly have a higher success rate. A robot running our node would be able to ensure that a location it randomly selects is out of the way of another robot before moving there, thus reducing the amount of time that is wasted from having a robot move and check its coordinates against the other robot's coordinates each time to verify that is no longer in an area that could possibly interfere with the other robot's path.

In addition, we would like to be able to have a mobile robot recognize that their given path could impede the path of another mobile robot, and then act accordingly in a way that would allow the other robot to proceed unimpeded along its path before the mobile robot running our node could continue on its given path. In order to achieve this goal, we could program a node that would estimate another robot's future coordinates based on the locations of past movements, and then confirm that the robot with our node running is not in the approximate path of the other. Another node could also be written for the instances when a robot is in fact in the way of the estimated path of another robot. This program would prompt the robot to momentarily halt its preplanned course to its target location, then determine a position that is out of the way of the other robot. Once the other robot has safely passed and neither robot poses as an obstacle in each others' paths, our robot with the node running would be allowed to resume its original course to complete it previously given task.

VIII. References

[1] Website with all of the robot's given information
http://nixons-head.csres.utexas.edu:7978/json

[2] Yingying, Ding, He Yan, and Jiang Jingping. "Multi-robot cooperation method based on the ant algorithm." In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pp. 14-18. IEEE, 2003.
http://ieeexplore.ieee.org/document/1202241/

[3] Botelho, Sylvia C., and Rachid Alami. "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement." In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, pp. 1234-1239. IEEE, 1999.
http://ieeexplore.ieee.org/document/772530/

[4] Regele, Ralf, and Paul Levi. "Cooperative multi-robot path planning by heuristic priority adjustment." In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5954-5959. IEEE, 2006.
http://ieeexplore.ieee.org/document/4058417/

[5] Shucker, Brian, Todd Murphey, and John K. Bennett. "A method of cooperative control using occasional non-local interactions." In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1324-1329. IEEE, 2006.
http://ieeexplore.ieee.org/document/1641892/

[6]Alami, Rachid, Frédéric Robert, Félix Ingrand, and Sho'ji Suzuki. "Multi-robot cooperation through incremental plan-merging." In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3, pp. 2573-2579. IEEE, 1995.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=525645

[7]Alami, Rachid, Sara Fleury, Matthieu Herrb, Félix Ingrand, and Frédéric Robert. "Multi-robot cooperation in the MARTHA project." *IEEE Robotics & Automation Magazine* 5, no. 1 (1998): 36-47.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=667325

[8]Chaimowicz, Luiz, Thomas Sugar, Vijay Kumar, and Mario Fernando Montenegro Campos. "An architecture for tightly coupled multi-robot cooperation." In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3, pp. 2992-2997. IEEE, 2001.
http://ieeexplore.ieee.org/document/933076/?part=1

Link to code on Github: https://github.com/BRKRAU0310/robot_avoid
Demos:https://drive.google.com/open?id=0B8nR9JX1ZNYBOE4xNzNJUVhjbXl6YXJ2c1p5Qjh1aWNlM2xN