ENPM808X FINAL-TERM PROJECT PROPOSAL

Manfred - The Automated Guided Vehicle

Shelvin Pauly
M.Eng. Robotics
University of Maryland
College Park, MD - 20740
spauly@umd.edu

Venkata Sairam Polina
M.Eng. Robotics
University of Maryland
College Park, MD - 20740
sairamp@umd.edu

Component description:

This project involves the development of a AGV to transport a package around the warehouse and store it at desired location. It also involves simulating a robotic system using ROS and Gazebo for an ACME Robotics product.

Brief explanation of importance:

Collection robot

Automated Guided Vehicles (AGVs) have become essential in today's technologically advanced warehouses and the need for Automation has fueled this. Assume an industrial environment/warehouse setup in constant need of a huge workforce for sorting, pick and place or transporting objects. The industry conventionally uses conveyor belts for the transportation of mechanical/electrical components in an industry setup and products in a warehouse setup. Often, these belts are found to be obsolete with minimal advantages as they are static and, in most cases, cemented to a specific spot. This is where the AGVs score as they are accounted to have less overhead and flexible loading options. AGVSs can also adapt to the size and makeup of the payloads, thus making them highly customizable. They can be modified in terms of weight, size, and configuration. Another prominent feature is that once programmed, they can work independently, even during off-hours. We will be developing an AGV robotics system that takes inputs from the user before every run about the location of the object/item to be picked up. The AGV will then autonomously navigate to the location and try to find the package. Finally, it will pick it up and deliver to the location specified by the user. This ACME Robotics Product will also use vision data along with the sensor feedback, to avoid collision and choose a different path to traverse through the environment. Furthermore, the vision data will be used to identify the objects. Once identified, the operation of manipulation is done to pick-up the object and then we navigate through the environment again to reach the destination and manipulation again to drop it off.

Proposed Methodology:

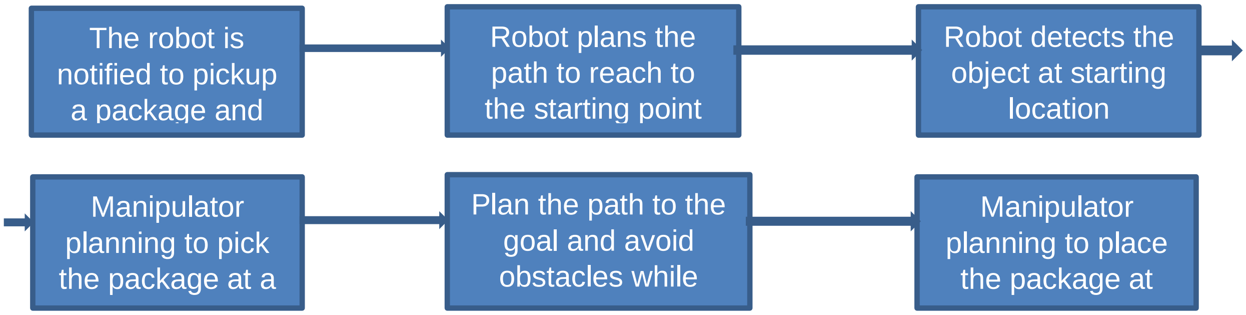
As the robot intended to design works in a closed environment, with the ability to pick a package and store and it desired location. It also can get a package from the storing racks of the warehouse to delivery location. As the AGV moves in a closed environment, a predefined path can be provided to robot on where it can move and where it cannot. It should have the ability to avoid different obstacles and make way for other robots. We plan to use the ubiquitous turtlebot3 with manipulator attached on top of it. The objective of the robot is to carry small packages as well as search and retrieve packages and also store packages in narrow spaces. So, the robot should have the ability to do following abilities:

Store a package:

As soon as package is sorted and needs to be stored for a later delivery this robot gets notified of a location where to collect the package through a **Ros service**. Empty locations can be retrieved from the **parameter server** and this empty location can be set as a goal to robot. But, before that the robots need to pick the package from the sorting station. The robot's camera detects the objects and manipulator planning should be done to pick the package which can be done using Movietonews, the robot should plan a path to an empty slot to put the package. **Ros Nav** packages can be used to navigate to the goal and manipulator planning can be done to successfully store a package.

Retrieve a package:

All the functionalities described above will be used by the robot but in a opposite direction to get the package for the delivery.


The robot is notified to pickup a package and
Robot plans the path to reach to the starting point
Robot detects the object at starting location
Manipulator planning to pick the package at a
Plan the path to the goal and avoid obstacles while
Manipulator planning to place the package at

Functionalities involved to pick up and drop a package

Team Members, Roles and Responsibilities:

• Venkata Sairam Polina (Driver)	• Shelvin Pauly (Navigator)

The roles are subject to rotation among the team members to encourage equal contribution and overall development experience for everyone. This switching will be done after a thorough discussion and analysis of skill and conceptual competency for the development phase.

Deliverables:

- A robotics system simulation that can identify an item, collect, and drop it from locations specified by the user using C++ and ROS packages.
- A demonstration of the collection and delivering of the package using random locations.
- Software Version Control with Git and detailed commit history.
- UML Diagrams, Log Sheets with timestamps, Software Development Timeline.
- Continuous Integration and Code coverage using Travis CI and Coveralls
- Profiling and Memory Leak checking using Valgrind.
- Developer level Documentation using Doxygen.

Software Development Strategies: TDD, AIP, Pair Programming.

Software Technologies:

Programming Language: C++

Development, Debugging and Testing Tools:

- ROS2 Humble
- Colcon build
- Navigation2
- Moveit
- cpplint
- cppcheck
- clangd
- Git
- Valgrind
- GTest
- Gmock
- pytest
- VScode

Operating System: Ubuntu 20.04

Documentation: Doxygen

Continuous Integration and Code Coverage: Travis CI and Coveralls

Libraries: OpenCV 4.6.0

Potential Risks and their mitigation:

The robot might not be able to comprehend the fast-moving obstacles in the map. In those cases, we aim to provide a constraint that the robot will wait for the obstacle to clear off from the range of the robot.
We are also accounting for the case where the local cost map fails and provides false solutions, in those cases, the global planner will supersede the local planner path.
This project will be implemented as pair programming where there will be timely switching between driver and navigator roles, we mutually plan to share the design-keeping duty. Git version control will be used with consistent and clear commit history to track the updates. More than one branch will be created in the repository for the team to work and each pull request will be merged after thorough inspection by the significant other. Software development processes like TDD and AIP will be used.

Project Code Name: Project Manfred (Mammoth from Ice Age)

References:

- Macenski, S. *et al.* (2020) “The marathon 2: A navigation system,” *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Alexy M, Steve M, “A Comparison of Modern General-Purpose Visual SLAM Approaches,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)