Decentralized multi-robot logistics

Remco Aarts, Jeroen van den Akker, Robert Delmaar, Bas Janssen, Addie Perenboom, Dimitri Waard

***Abstract*—This paper details the design and implementation of a decentralized multi-robot logistics system. Both the hardware design and the software design and implementation are expended upon below.**

***Index Terms*—**

I. Introduction

The decentralized multi-robot logistics project is one of the main projects of the Adaptive Robotics Minor at Fontys Hogeschool Engineering, in the 2016-2017 academic year. The goal of this project is to develop a system, both hardware and software, that is capable of transporting product between a warehouse and production locations in an Industry 4.0 setting. The system can be a replacement for a traditional conveyor belt, or can be used as a flexible addition to an existing conveyor system.

II. Hardware

A. Stackup

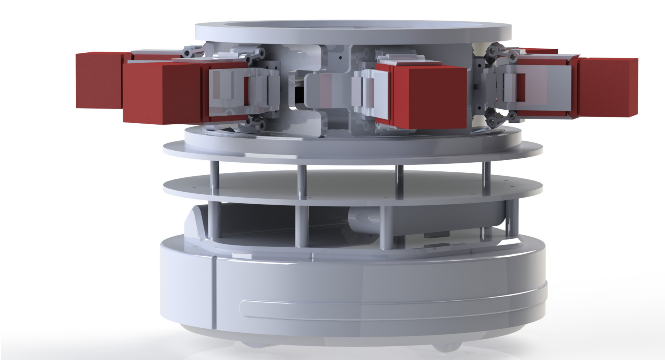

Fig. 1. Mechanical Design

B. Gripper

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

III. Software

During this project a team of two students has developed the software making the decisions on the robots. This software is designed to receive commands from one or more workstations, and have to robots decide which robot should execute a command. The robots have the ability to request help from one another when more product have to be transported than a single robot can carry, or it is beneficial to transfer one or more products between two robots to get the products to their destination faster.

A. Distance calulation

The robots make decisions based on their available inventory space, and the distance between the robot and the goal. The distance of the planned path is calculated using equation 1. This means that for each point in the path, the distance to the next point is calculated. The length is the sum of these distances.

$$x = \sum_{i=0}^{j} \sqrt{(y_i - y_{i+1})^2 + (x_i - x_{i+1})^2} \tag{1}$$

B. Multimaster-fkie

All the robots in the group communicate with each other. As each robot is its own ROS master, the normal [1] way of using multiple machines in the ROS ecosystem is not viable. However, a package capable of communicating between multiple independent ROS masters has been developed at the Fraunhofer-Institut fr Kommunikation, Informationsverarbeitung und Ergonomie FKIE, called multimaster-fkie [2]. Using this package it is possible to synchronize one or more topics between one or more ROS masters. This project uses said functionality to synchronize the command and control topics for the robots between the robots and the workstations.

The multimaster package provides two nodes, namely master-discovery and master-sync. The master-discovery node uses a multicast address to find the other masters on the network, and this list of masters is passed to the master-sync node to synchronize the specified topics.

C. Simulation

To test the software during development a simulation in Gazebo has been configured [3]. The simulation has been configured for four robots, because the computer systems available could not simulate more robots at the same time with reasonable performance. Four robots also allow for testing assistive operation and provide the capability to have multiple robots running multiple different commands.

Within this simulation, the complete robots are simulated in a fully simulated world. This means that the robots are subject to simulated physics, collisions are possible and the entire software stack for each robot is running. To complete the simulation, a virtual world has been build for the robots to move around in. This would consists of eight one by one by one meter cubes that represent obstacles. The cubes are spaced through out this virtual world.

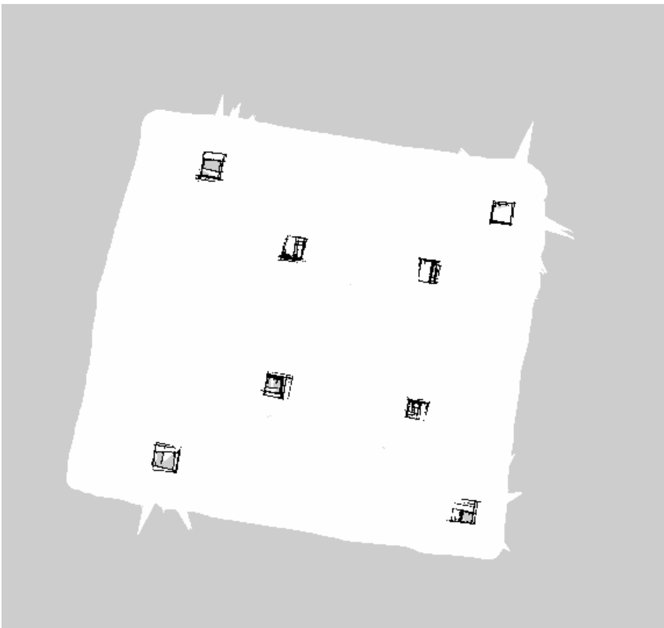

Fig. 2. Map of the simulated world

Figure 2 shows the map of this virtual world generated with gmapping SLAM package [4], and edited to expand the known area. The white area represents the know area, grey represents unknown and the black markings are the detected obstacles.

D. Software scales to more robots

The software has been designed to scale to an unlimited number of robots. The only requirement is that the systems running the robots have enough RAM to store all the information for the robots. The software has been tested in the simulation with four robots.

E. Software performs better when using a big map

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

F. Replace turtlebots

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

REFERENCES

[1] JanDebiec. Running ros across multiple machines. [Online]. Available: http://wiki.ros.org/ROS/Tutorials/MultipleMachines

[2] A. Tiderko. multimaster_fkie. [Online]. Available: https://github.com/fkie/multimaster_fkie

[3] Jakub. (2012, August) Multiple robots simulation and navigation. [Online]. Available: http://answers.ros.org/question/41433/multiple-robots-simulation-and-navigation/

[4] B. Gerkey. slam_gmapping. [Online]. Available: https://github.com/ros-perception/slam_gmapping