Autonomous control of a warehouse forklift

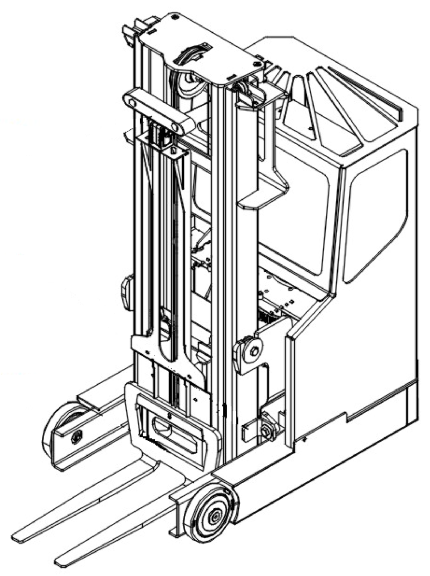
Henrik Andersson, Johan Almgren, Gustav Elingsbo, Mikael Hartman, Petter Lannerhed, Andreas Norén, Jenny Stenström

Abstract

The interest for autonomous control and the possibilities to provide a higher efficiency for the customers and to create competitive advantage towards other competitors has risen dramatically lately. Of special interest is the warehouse automation, where forklifts manage their tasks with minimal or no interaction from humans. Using sensor fusion, digital signal processing and control theory the forklifts are able to execute tasks by themselves within the boundaries of the warehouse.

System overview

This project has been a collaboration with Toyota Material Handling, where the goal was to improve the existing system of a autonomous warehouse forklift.
To reduce the risks of testing developed code on a physical forklift there were a forklift at scale 1:3 available instead of a full size forklift. A simulation environment were mainly used during the project where the included forklift was also at scale 1:3.



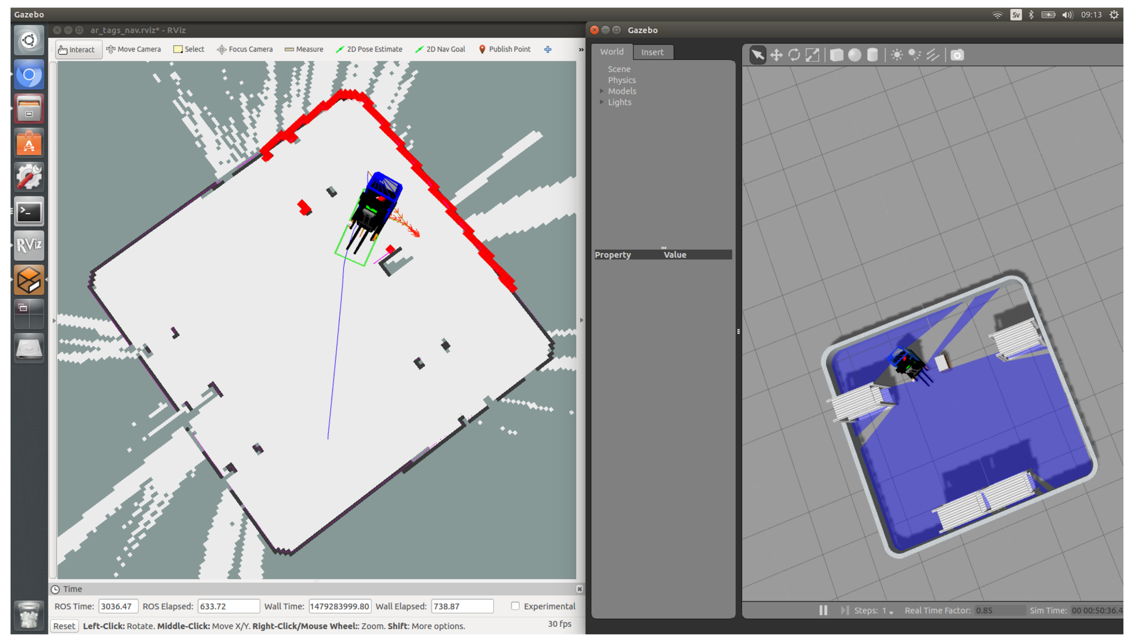
The forklift Minireach used in the project.

The software used is based on the open source platform called ROS (Robotic Operating System). It consists of a set of software libraries and tools used to build robot applications. The applications was altered and imporved by custom made Python and C++ code.

Simulation environment

The simulation environment enables the possibility to develop and test the system without having access to the actual forklift.

The environment consisted of two programs. The first program is called Gazebo, which is the actual simulation world where the forklift operate. It also show what is visible to the forklift by LUA sensors.



Right: The environment Gazebo, representing the actual simulation world.
Left: The program RViz presenting sensor data on a saved map.

The second program is called RViz and presents how the forklift interprets its surroundings, such as sensor data. RViz can be used both for simulation and the actual forklift.

Flexible, modular system architecture

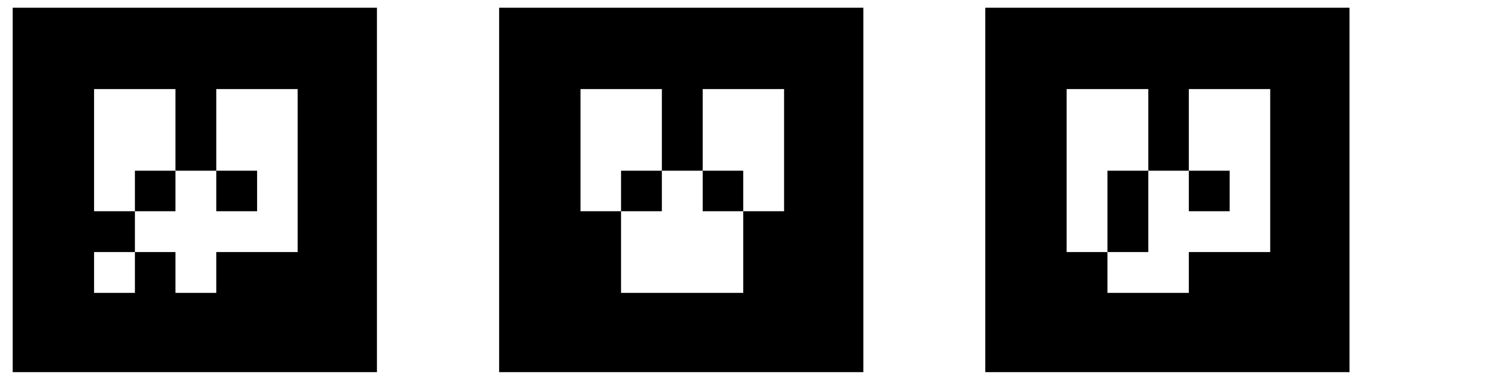
One of the main advantage of the development system is the possibility to add new features, thanks to the module based design. The different functionalities are joined by a structure called SMACH (State MACHine). SMACH is a functionality provided by the Python library and by combining them with features from ROS it is easy to execute and follow up tasks.

Planning and following a route

Using the current map, including stationary obstacles, the forklift is able to plan a route to a desired destination. To locally follow the route the planner TEB is used, which is an abbreviation of Timed Elastic Band. The planner locally optimizes the robot's trajectory with respect to the trajectory execution time and distance to obstacles. It also takes into account kinodynamic constraints at run-time. It is important to notice that the forklift strive towards driving with the forks pointing backwards.

Detection and interpretation of AR-markers

To be able to find and distinguish the pallets and pallet stations each one of them is identified by a so called AR-marker, shown below.



Three examples of AR-markers, which were used to identify pallets and pallet stations.

The system is able to detect these AR-markers and store them onto a file, along with the world position of the marker. Each time a new marker is detected it is added to the file and if a previous added marker is seen again, its position is updated on the file.

Mapping

With the newly released Cartographer,developed by Google, the forklift is able to map the world, save the map and then use the map to navigate. Cartographer focuses on LIDAR SLAM, even though we only use the mapping functionality of Cartographer. The localization is taken care of by the system AMCL, a probabilistic localization system.

Autonomous repositioning of pallet

The main focus of the project was to move a pallet from its current position to a marked pallet station. With help from the implemented functionalities above, we designed a state machine to execute the scenario and possibility to manage potential faults. More over there was a necessity to develop a method to position the forks in the holes of the pallet. This was solved by using the laser scanners and positioning the forks in line with the front of the pallet.

LINKÖPING UNIVERSITY
TOYOTA MATERIAL HANDLING

LINKÖPING UNIVERSITY
Division of Automatic Control