TECHNISCHE HOCHSCHULE NÜRNBERG
GEORG SIMON OHM

Technische Hochschule Nürnberg Georg Simon Ohm
Kesslerplatz 12
90489 Nuernberg
Germany

autonohm@work Team Description Paper for RoboCup@Work 2016

Dominik Heigl
Florian Gramß,
Alexander Gsell,
Tobias Fink,
Jon Martin

Advisors:
Prof. Dr. Stefan May

Nuernberg, January 30, 2016

1 Introduction

The Autonohm@work team at the Nuernberg Georg Simon Ohm University of Applied Sciences was founded in September 2014. The team consists of Bachelor and Master students leaded by a research assistant who guides and supervises them. To develop a functional mobile-robot-manipulator the different groups in the team had put much effort and knowledge into research to develop the packages to manage the robot. Our main focus is attended to mobile manipulation, object perception and navigation in a unconstrained environment.
Only two members of the team that took part last year in Magdeburg continue. To compensate that, new members have joined us and the cooperation between the rescue and the atwork teams has been intensified.

2 Robot Platform

2.1 Hardware

We use the KUKA youBot omni-directional mobile platform, which is equipped with a 5 DOF manipulator. At the end effector of the manipulator an Intel RealSense camera with a motion sensor has been mounted. A Hokuyo URG-04LX-UG01 laser scanner at the front of the youBot platform is used for localization and navigation. Last year we use the internal computer, together with an external ASUS Mini PC (4 GB RAM, Intel Core i3). To avoid the communication problems and latency between them both, we decided to replace them for a unique more powerful CPU Intel Core i7-4790K, 4x 4.00GHz. We also modified the original grasper with a new one to be able to grasp bigger object more precisely.

2.2 Software

The software architecture is based on the Robot Operating System ROS. The actuators of the robot are controlled by the basic youBot-ROS-drivers. The system runs with Ubuntu 12.04 and ROS Hydro ROS communication infrastructure is used to communicate and pass information between the nodes, as for example camera date or execution orders for the 5 DOF manipulator.

Several software tools are needed for image processing and controlling the system.
openCV
ohm_tsd_SLAM
amcl
Stack_Navigation
PCL

3 Object Manipulation

To grasp objects reliably an exact position from the object perception is needed. The position of objects will be calculated based on information, received from optical/infrared sensors (2D and 3D). After the calculation is finished the robot will navigate to a pregrasp position. Once the base has reached the final position, kinematics will lead the arm near the object. For precise gripping a 2D/3D optical/infrared sensor has been attached to the end effector. In gripping stance the arm-

camera will be activated to measure the final gripping pose. Because manipulation is an upcoming issue in our robotic institute, we decided to build our own inverse kinematic.

4 Image processing

This task force deals with recognizing of components to grasp and QR-codes. With the help of an Intel RealSense camera the orientations and positions should be determined by given objects. Then this information is made available for the robot.

5 Localization and Mapping

The Navigation is based on the ROS Navigation stack.

5.1 Global Navigation

In order to create the map, we use the SLAM developed by our colleges in Rescue team at the Georg-Simon-Ohm University of applied science. The SLAM itself is based on the iterative closest point (ICP) algorithm and laser data provided by the front-mounted Hokuyo URG04-LX laser scanner. Then we base the localization on the amcl package provided by ROS.

5.2 Local Navigation

The data provided by the Hokuyo is additionally used for collision avoidance. If an obstacle is detected, a report goes to the path planning state which adjusts the path.

5.3 Mission and path planning

For planning issues, a State Machine with singleton pattern design is used. Every state is designed to be as small as possible. The origin and destination is handed over to an A* path planning algorithm. For debugging issues, the nominal value of the joints can be controlled by a virtual master.

6 Conclusion

In this paper we gave a brief description about our robots modification and functions. We use and develop existing software to make it even better but we also have to invent new methods and software.