

INDUSTRIAL AND ROS-ENABLED MOBILE ROBOTS COOPERATION

W. Kaczmarek^{*}, K. Besseghieur^{**}, J. Panasiuk^{***}, P. Prusaczyk[†], M. Siwek[‡]

Abstract: *In this article, the authors focus on the issue of developing an application enabling the cooperation of mobile robots with industrial robots. The mobile robots (TURTLEBOT) are based on ROS, whilst industrial robots (IRB 120) on RobotWare. The result of this project is robots cooperation using TCP/IP protocol communication. The robots cooperate to complete the required task which is a pick-place process of an item by the industrial robot on the mobile robot, which must navigate autonomously to the required position. The developed applications have been tested in a virtual mode using RobotStudio and ROS and in real mode with the use of real robots: mobile and industrial robots.*

Keywords: industrial robots, mobile robots, TCP/IP communication, robot cooperation

1. Introduction

Production is a basic field activity which is necessary to meet the needs of individuals and for the development of society. Nowadays, due to the high competition and high demands of consumers, it is necessary to look for modern and flexible solutions of production lines. Currently existing production systems are limited especially by the efficiency of production development, the efficiency of the introduction of new techniques and limitation of the flexibility of making changes in production. The rise of new digital industrial technology is known as Industry 4.0 (Wruetz et al., 2017). In this new vision, sensors, machines, workpieces, and IT systems should be connected together and interact with one another using standard Internet-based protocols. The researchers agree that this will greatly improve the efficiency, production flexibility and adaptation of production to the needs of final recipients.

Devices and software available on the market allow for effective increase of production flexibility by combining many technologies. The use of mobile/industrial robots cooperation in industrial applications can significantly reduce the time to adapt automated production lines to new needs. An example here can be the increase of production flexibility by using mobile robots cooperating with industrial robots on production lines. Depending on the type of production, a group of mobile robots can perform transport tasks for the needs of industrial robots. In this brief, the cooperation experiment consists of running a picking-placing process of an item by the industrial robot on the mobile robot, which has to navigate autonomously towards the desired position.

^{*} Wojciech Kaczmarek, PhD.: Faculty of Mechatronics & Aerospace, Military University of Technology, 2 Urbanowicz Street, 00-908, Warsaw, PL, wojciech.kaczmarek@wat.edu.pl

^{**} Jarosław Panasiuk, PhD.: Faculty of Mechatronics & Aerospace, Military University of Technology, 2 Urbanowicz Street, 00-908, Warsaw, PL, jaroslaw.panasiuk@wat.edu.pl

^{***} Khadir Besseghieur, MS.: Faculty of Mechatronics & Aerospace, Military University of Technology, 2 Urbanowicz Street, 00-908, Warsaw, PL, besseghieurkh@hotmail.fr

[†] Piotr Prusaczyk, MS.: Faculty of Mechatronics & Aerospace, Military University of Technology, 2 Urbanowicz Street, 00-908, Warsaw, PL, piotr.prusaczyk@wat.edu.pl

[‡] Michał Siwek, MS.: Faculty of Mechatronics & Aerospace, Military University of Technology, 2 Urbanowicz Street, 00-908, Warsaw, PL, michal.siwek@wat.edu.pl

2. TURTLEBOT – mobile robot

TURTLEBOT (see Fig. 1) is an indoor mobile robot that is modeled as a two-wheeled differential drive robot. It runs ROS which is a Linux-based, an open source software package that provides a software framework to aid in the development of robotic application. For more details, the reader may refer to (Quigley et al., 2015).

In this project, the mobile robot must be able to carry on its own the two following tasks:

- Navigate autonomously in the working environment to reach the required point (See Fig. 1).
- Communicate with the industrial robot.

2.1. The robot navigation

The navigation stack is used for this purpose. It allows the robot to navigate safely within the environment as shown in (Bessegheir et al., 2017a; Bessegheir et al., 2017b) and (Hennes et al., 2012). It takes in information from odometry and sensor streams and outputs velocity commands to send to a mobile base (Żygadło et al., 2016). The navigation stack is open source and can be downloaded from the ROS community. Before starting navigating, the robot needs an offline-built 2D map of the environment necessary for the robot's localization. The map is generated using the open-source package G-Mapping where the robot is teleoperated around the environment; the 2D map is built and stored locally on the robot's laptop. To start the navigation, the two following instructions are launched from two separate terminals:

```
roslaunch turtlebot_bringup minimal.launch
```

```
roslaunch turtlebot_navigation amcl_demo.launch map_file:= path to the map file
```

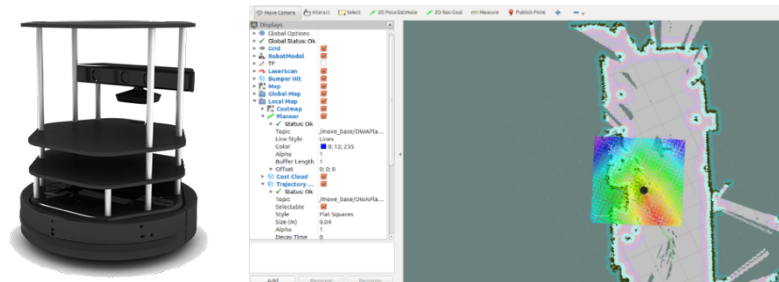


Fig. 1: General view of TURTLEBOT and autonomous navigation of a TURTLEBOT in RVIZ

2.2. The coordination node

First a package named “coordination” with a dependency on the move_base_msgs, actionlib, and roscpp is created. The coordination node is responsible for the communication with the industrial robot and for sending the goal point to the navigation stack and therefore, the code is divided into two parts. The first part consists of creating the client socket and establishing the communication with the server. Whereas in the second part, the goal point is created and sent to the navigation stack via a ‘ROS action protocol’. In our application, the goal is a move_base_msgs: MoveBaseGoal message type that contains information about where the robot should move to in the world. The node is executed from a terminal with the necessity to specify the IP address and the port number of the server as private ROS-parameters:

```
Rosrun coordination coordination_node _ip:= IP address of the server _port:= port number of the server
```

3. IRB 120 - industrial robot

The ABB IRB120 (see Fig. 2) is the smallest of the ABB industrial robot family. The manipulator has a weight of 25 kg and reach approximately 580 mm. The manipulator lifts objects up to 3 kg, the weight of objects increases to 4 kg in only vertical movement scenarios. The robot is controlled by the IRC5 controller.

The latter contains the electronics required to control the manipulator, additional axes and peripheral equipment. In the IRC5, the RobotWare-OS is installed. The IRB 120 is programmed in RAPID language. In this project, the Festo gripper is used for grasping and releasing objects. The main task of the robot is to pick and place items. RobotStudio's online environment was used to program the robot.

4. Industrial and mobile robots cooperation.

The two robots communicate through a common wireless network provided by a WIFI router. To exchange data between TURTLEBOT and ABB IRB120 via TCP/IP, the latter can be connected only by wired Ethernet cable whilst the mobile robot is connected via WIFI as depicted in Figure 2.

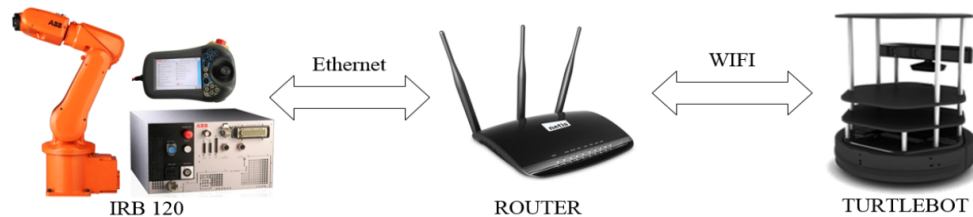


Fig. 2: Connection model

Protocol of communication is based on the client-server model, which is the basic interaction pattern between applications or processes communication. Server's role is to provide services, acquires demands and returns results. It is usually implemented as a software application that can be applied to any type of computer operation system, which supports TCP/IP communication model. The client sends a request for services provided by the server and wait for the reply (Comer et al., 1997). The developed application operates as shown in the diagram in Fig. 3.

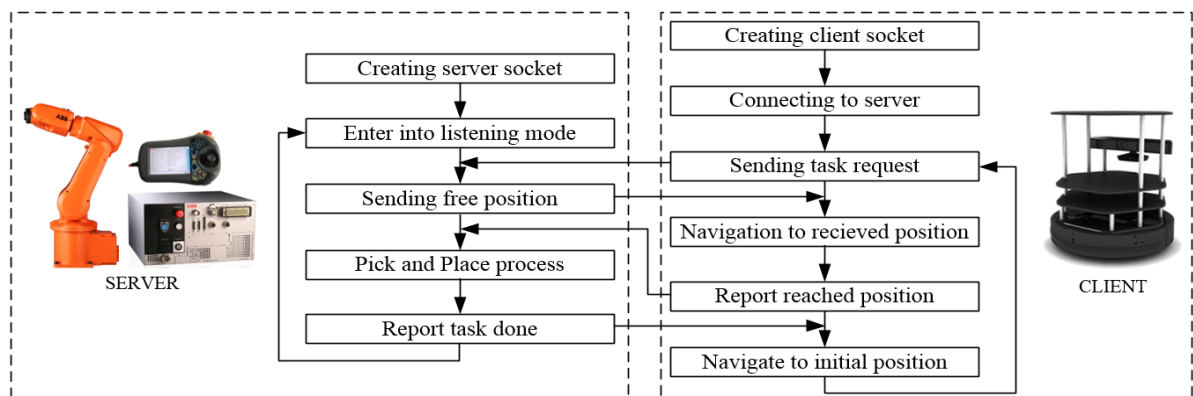


Fig. 3: Client-server work algorithm

Thanks to modern programming environments, it was possible to develop and test applications in off-line mode (without the use of real robots). The correct functioning of the programs has been tested in the RobotStudio environment (for IRB 120 - Fig. 2) and in RVIZ (for TURTLEBOT - Fig. 1). The applications were run on two PCs connected to the local Ethernet network.

In the next step, the applications were tested using real robots (see Fig. 5). Several scenarios have been analyzed, especially examining the correctness of the tasks performed by the robots and the communication with each other. Tests have proved the correct functioning of the application.

Experiments with real mobile and industrial robots were conducted as depicted in Figure 5. After the mobile robot receives the goal position, it makes its way to it based on the navigation stack implemented on the robot (see Fig 5a). As soon as the TURTLEBOT reaches the required position, it sends an acknowledgment to the industrial robot and the latter starts the process of picking-placing the item on the mobile base (see Fig 5b). The industrial robot sends an acknowledgment to the latter that the item has been placed then the mobile robot starts navigating again to its initial place as shown in Fig 5c.

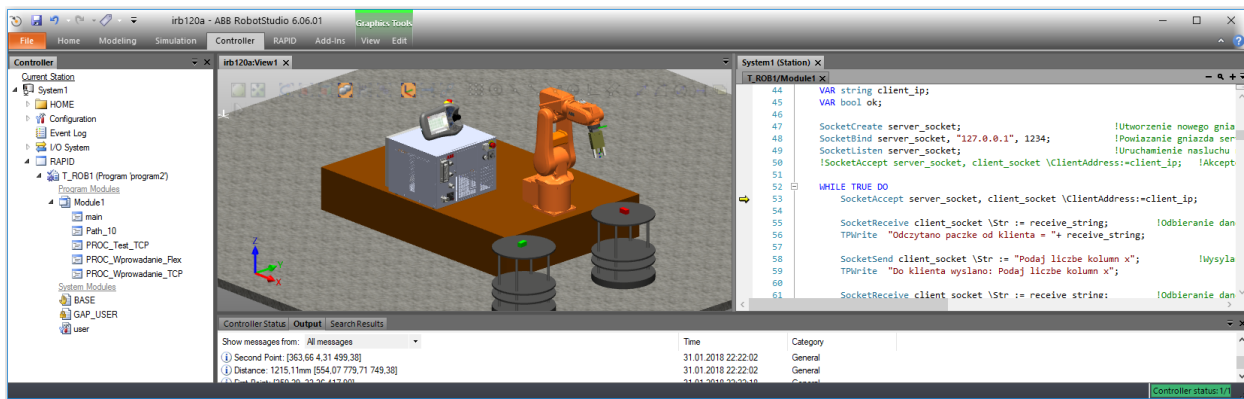


Fig. 4: General view of the cell in RobotStudio

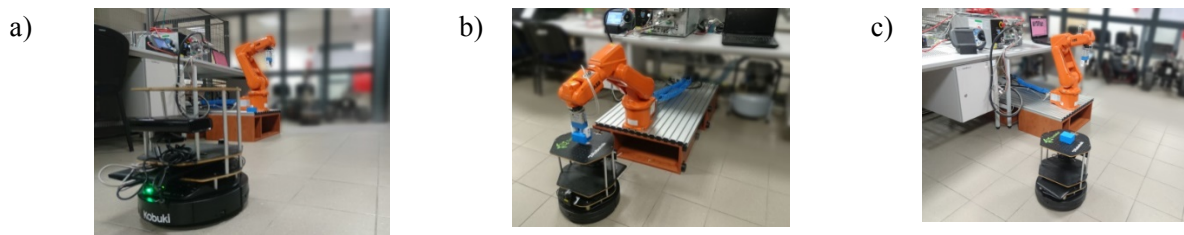


Fig. 5: General view of the cell in reality.

5. Conclusion

In this work, the cooperation between mobile robots based on ROS and industrial robots ABB is investigated. Robot navigation is implemented on the mobile robots whereas the pick-place process is programmed on the industrial robot. The two robots coordinate and communicate based on the IP/TCP protocol where client and server sockets were created on the mobile and industrial robots respectively. The communicated data between the two robots consists mainly of sending acknowledgements that trigger the other robot to start its next task until the whole procedure of picking-placing an item on a mobile robot, which is initially located far from the industrial robot, is done. Experiments were conducted successfully using TURTLEBOT and ABB IRB120 demonstrating the effectiveness of the mobile/industrial robots cooperation. Due to the autonomy of mobile robots, the discussed solution increases the production flexibility for synchronous and asynchronous work. Depending on the needs, materials with different characteristics and dimensions can be supplied to the cell of industrial robot, which is difficult to implement when materials are delivered on conveyors. Future applications will include multiple mobile robots and thus collision avoidance and accuracy of the system will be investigated.

References

- ABB Robotic Division, (2014) ABB IRB120 User Manual, ABB
- Bessegheir, K., Kaczmarek, W., Panasiuk, J., and Prusaczyk, P., Formation control of mobile robots under ROS, Engineering Mechanics 2017, Brno University of Technology, Brno, pp. 138-141.
- Bessegheir, K., Kaczmarek, W., Panasiuk, J., Multi-robot Control via Smart Phone and Navigation in Robot Operating System, in Problems of Mechatronics Armament Aviation Safety Engineering, 2017, volume 8, pp. 37-46.
- Comer, D. (1997) Internetworking with TCP/IP Volume I Principles, Protocols and Architecture Polish edition, WNT, Warszawa
- Hennes, D., Claes, D., Meeussen, W., and Tuyts, K. (2012). Multi-robot collision avoidance with localization uncertainty. In Proc. 11th Int. Conf. on Autonomous Agents and Multiagent Systems-Volume 1 (pp. 147-154).
- Quigley M., Gerkey B., and Smart W. D., (2015). Programming Robots with ROS, a Practical Introduction to the Robot Operating System, First. O'Reilly Media, Inc.
- Wrutetz T., Goltz J., Biesenbach R., Wireless Multi-robot Network Approach for Industry 4.0 using RoBo2L, International Multi-Conference on Systems Signals and Devices, 2017, pp. 661-664.
- Żygadło, S., Podciechowski, M., Rodzik, D., Functional solutions of radio altitude sensors, Elektronika: konsultacje, konstrukcje, zastosowania vol.57, 2016, pp. 22-26.