https://doi.org/10.26552/com.C.2016.1A.81-86

Alexander Rengevic – Darina Kumicakova *

# NEW POSSIBILITIES OF ROBOT ARM MOTION SIMULATION

*The importance of computer simulations of the industrial robot arm motions at the technological scene becomes more important especially in connection with industrial applications of the new types of robot arm kinematic structures or with solving the human robot safety collaboration in industry. The article is aimed at the simulation of the basic (primitive) tasks executed by industrial robot with utilization of software packages of the "open-source" platform ROS. ROS is the software of modular architecture that extends the capabilities of the different robot types programming and controlling.*

***Keywords:*** *Computer simulation, industrial robot, ROS, program.*

## 1. Introduction

Computer simulation belongs to group of important tools for aid of industrial robotic development. Process of the simulation enables engineers to solve many tasks related to designing of the industrial robots construction and configuration, and also to test them at the robotized workplace virtual model by simulation. These tasks type can be classified as "basic tasks" which are usually solved with support of the complex program packages known as CAD/CAE and CAPP systems [1 and 2]. Their tools allow user to create the detailed models of the robot individual constructional parts, to define kinematic joints between them, to create the robot arm assembly and simulate its motions, velocities, accelerations and acting forces and torques, to test the design of the robot. For solving the complex tasks related to the industrial robot programming and implementation of the various input/output devices, it is necessary to use the programs belonging to the group of computer aids of manufacturing engineering (CAPE). Such programs consist of the special functions applicable only to the certain robot type and to solving the specific tasks. These programs enable to simulate: process of the robot programming, changes of the robot control system settings, controlling of input/output devices, grasping of parts and manipulation with them and also to test the robot working cycle at the virtual model of a robotized workplace. The Department of Automation and Production Systems has the practical experiences with solving the tasks related to the simulation of robotized welding and deburring, and also with robots cooperation. The Department has been using a special program of computer aided robot control and programming [3].

Till this time, the industrial robotics was focused most of all on automation of welding and painting processes, and manipulation with objects. But at the present the new trends in industrial robot applications are related to the request to involve them to the collaboration with human [4] or to place them at a mobile platform to be able to move inside a factory [5]. The programming of such industrial robots is much more complex and it requires an implementation of various specific devices or control algorithms which are not standard in current robotics. The solution of these problems is the object of interest of different development groups in the world. The effort to engage these specialists into the unified platform resulted in the project ROS Industrial that brings new possibilities to solve such complex tasks in industrial robotics [6] implementing a unified open-source platform The Robotic Operating System (ROS). This platform is freely accessible for researchers from around the world and enables them to cooperate and exchange information with each other and determine the future direction of the ROS.

In the framework of currently "running" European research projects [7] the several research workplaces (IPA Fraunhofer, Danish Technological Institute, TU Delft, LMS-University of Patras, etc.) participate in the research of tasks connected with intelligent and safety interactive Human-Robot (H-R) cooperation for assembly plants of the future. Research in individual solved projects is most of all aimed at hardware and software aids of symbiotic assembly solution and application of new technologies in area of the shared workspace operations sensing and controlling. The open-source system ROS is used for tasks relating to the H-R cooperation. This system consists

* **Alexander Rengevic – Darina Kumicakova**
Department of Automation and Production Systems, Faculty of Mechanical Engineering, University of Zilina, Slovakia
E-mail: alexander.rengevic@fstroj.uniza.sk

of tools as 3D sensing, automatic generation of trajectories or solution of inverse kinematic tasks [8].

The research of possibilities to utilize the platform ROS for creation of an intelligent safety system for the laboratory robot Fanuc LR MATE 200iC is now starting at the Department of Automation and Production Systems. Such a safety system should allow the robot to obtain online information from a surrounding workplace, recognize the objects in a monitored space and adapt the robot arm planned motions to current state of the risk of unwanted collisions with an object in the workplace. The article presents the basic information about the structure, tools of open-source platform ROS and selected steps related to the solving of the robot workplace simulation in ROS.

## 2. ROS and ROS Industrial

The ROS platform represents developmental environment that is used for creation of reliable programs for mobile and service robots control with utilisation of various types of hardware, drivers, libraries and other devices [9]. The ROS platform consists of the group of packages that include files starting the required tasks. The goal is to create the control program composed of several independent programs that can work together. Such starting program is called "nod". Single nods communicate with each other in environment called "Master" that is a very important part of the ROS platform. Dependencies between single nods in the framework of Master environment is possible to view with help of tool "rqt graph" - see example shown in Fig. 1. The ellipse represents a single nod. Arrows represent relations between single nods which send or receive messages related to the topic. The nod/turtlesim represents a simulation environment and it receives information from nod/teleop_turtle that provides motion commands. Both nods send information to the nod/rosout that collects published data from all available nods [10].

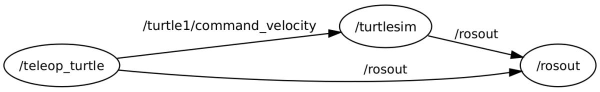


*Fig.1 Visualisation of dependence between individual nodes under the ROS*

When creating programs in ROS the build system catkin is used. The build system represents a tool that is able to translate the source code to files, libraries, binary files and scripts. When creating new projects in ROS it is necessary first to define the workspace. With regard to the directory the whole files essential for the right function of the created application must be placed. Structural arrangement of the working directory, packages, and codes which are used in the build system catkin is in Fig. 2.

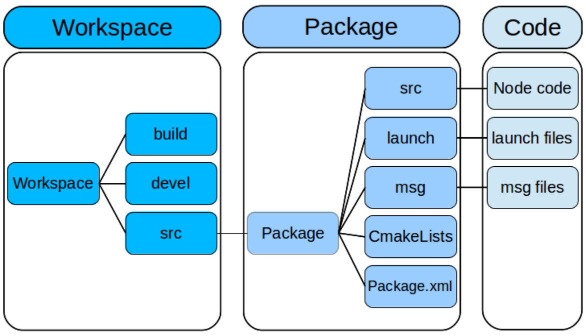


*Fig. 2 Structure of directories and files within the build system catkin*

ROS expands current possibilities of industrial robotics with new tools for spatial perception, recognition of objects, faces, gesture and motion tracking, stereovision and more. It is based on the support of the standard used hardware within robotics like industrial robots from known manufacturers (ABB, Adept, Yaskawa Motoman, Universal Robots or Fanuc), security elements and different types of end effectors and grippers. Besides all this, it has the ability to work with hardware which is not directly specific for usage in industrial robotics, for example, different types of image sensors and space depth sensors [9].

## 3. Simulation of robot working activity

The process of simulation is preceded by some basic steps. It is about choosing the right simulation tools, configuration of simulation tools and system linking. This section will describe each simulation capabilities and the environment in which the simulation will take place.

### 3.1. Simulation facilities

For the purposes of verifying and testing possibilities offered by the ROS platform – a notebook ASUS X550C computing device was used. The parameters of this device are stated in Table 1. The used operating system is preferentially intended for the ROS platform with codename Hydro. In GitHub repository which represents a worldwide stack of data for ROS community are available packages which are able to create programs for industrial robots Fanuc. For the use of an industrial robot Fanuc LRMate 200iC within platform ROS several initialization steps were executed. These initialization steps were able to recompile files created with ROS to formats compatible with the robot control system. This allowed the mutual communication between the robot and the ROS.

Table 1

| ASUS X550C | |
|---|---|
| Processor | Intel Core i5-3317U |
| System memory | 4016 MB DDR3 |
| Graphic accelerator | Nvidia GeForce 720M 2GB DDR3 |
| Operating system | Ubuntu 12.04.5 LTE Precise Pengoline |

## 3.2. Communication between the robot control system Fanuc R30iA Mate and ROS

For the compilation of these files the program Fanuc Roboguide V7 was used. This program contains an essential function for compilation of ROS files to format supported by the robot control system. After compilation this folders were uploaded to a robot control unit of the real robot. After this procedure configuration and initialization of the files was provided. For the exchange of information between the control system of the robot and ROS the ROS configuration had to be setup. This included a set of network address, communication port and start up of the supporting communication programs. For the complete recovery of communication it will be necessary to add the robot control system with the options R623 and R648. These options will be able to exchange data between both systems.

## 3.3. Communication between Kinect device and ROS

The next device which will be necessary to create an intelligent safety system is a sensor used for spatial vision. For confirmation of the capabilities of spatial vision implementation a Kinect device was chosen. The functionality and communication ability of this device was tested with the use of computer with operating system Microsoft Windows 7 and corresponding drivers. After confirmation of the hardware functionality the device was connected to the ASUS X550C notebook with corresponding packages. Subsequently, the application for verifying the basic setups for the Kinect device was launched.

As can be seen in Fig. 3 on the right, the quality of image from rgb camera is sufficient enough. In the figure on the left is an image from a depth camera which shows the distance between the person and object in the space with the assistance of the color scale. Based on these two figures it can be concluded that the sensor Primesense is working correctly. The implementation of the picture from the depth camera in simulation environment is described in the experimental part.

## 3.3. Simulation environment

For creation of the intelligent safety system the program MoveIt! was used. This program is able to define the kinematics of the robot, motion and trajectory planning, collision detection, spatial perception and more in the environment of ROS platform. For the purposes of simulation the MoveIt! a graphical interface Rviz was used.

Rviz is able to easily detect and adjust actual settings of the simulating environment, simulated model and also monitor and display output data from the sensors that the robot is equipped with.

The first step in the creation of simulation environment was the preparation of the mathematical model of the industrial robot. For the purpose of simulation of the model in fanuc_drivers a package from GitHub repository was used. This eliminated the need to create a model of a robot in CAD system and convert it with to URDF (Universal Robotic Description Format) supported by ROS.

The next step was to create a configuration package for the chosen robot model. In this step it was necessary to define a way how to solve the inverse kinematics with the help of one of the

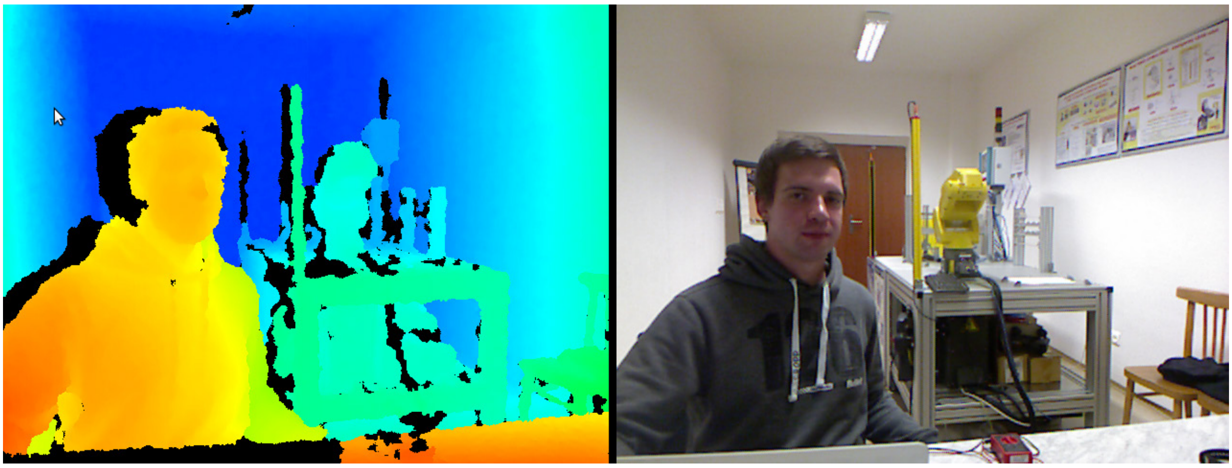


*Fig. 3 Verifying the functionality of Kinect device*

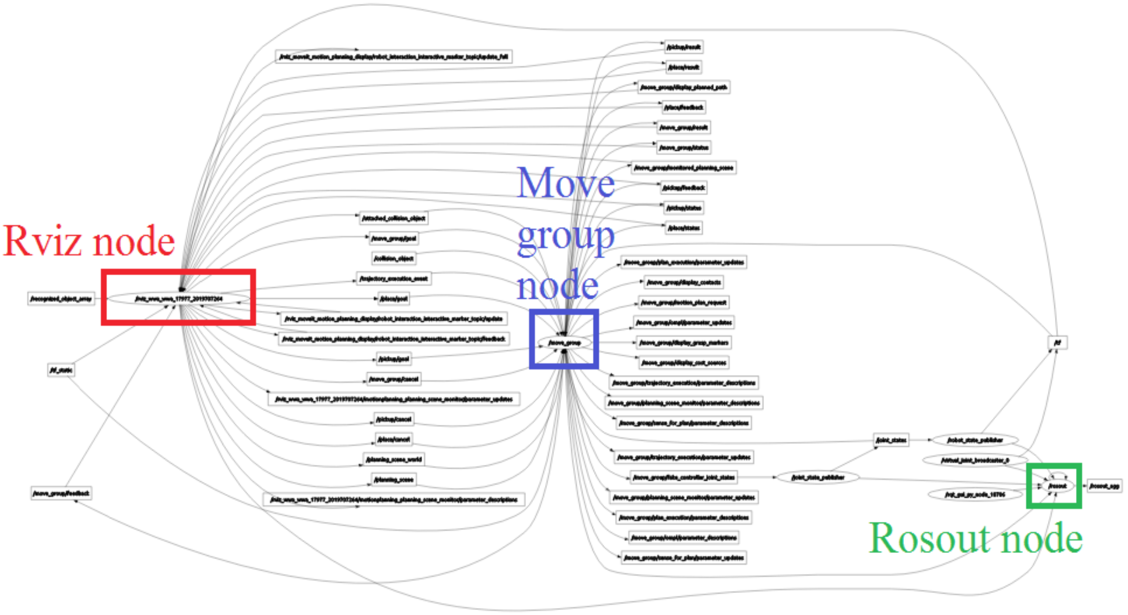


*Fig. 4 Dependence between nodes after simulating environment launch*

existing packages. The chosen library KDL (The Kinematics and Dynamics Library) is standardly used in ROS for industrial robots with six controlled axis. Except this, the collision states of joints, starting positions were setup and the range of the joints was verified. In the following step configuration files necessary for work in MoveIt! were generated. After launching the generated file in the Rviz environment we went in the definition of a move planner. This planner helps to specify the results of the solved kinematics. As a motion planner within ROS the OMPL (Open Motion Planning Library) was used. This library offers several ways of solutions depending on the chosen algorithm. For the purposes of experiment we chose a PRM (Probabilistic Roadmap Method) planner.

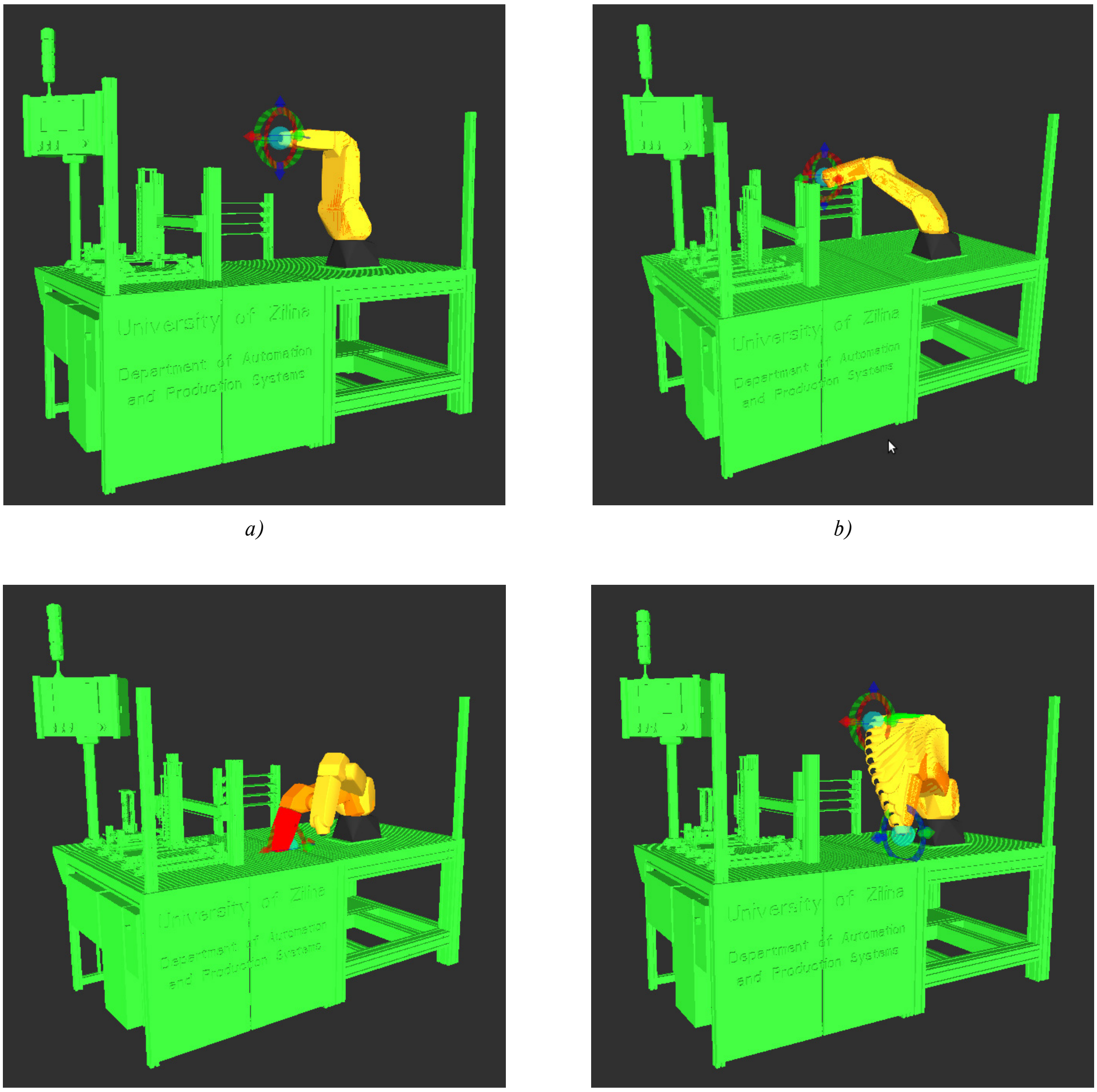
For the launch of the simulation environment it was necessary to use several nodes in which mutual dependences are represented in Fig. 4. As can be seen, the dependence net is quite complicated. The main nodes can be described as rviz node which represents the basics of simulation environments (marked in red color), move_group node which is responsible for the execution of the moving commands (marked in blue color) and node rosout (marked in green color) whose function is described in Chapter 2.

## 4. Simulation experiments

As first, the work of the industrial robot was simulated. The goal of the robot was to achieve required points within the operational space. The robot arm was, at the beginning of the simulation, in the starting position described in Fig. 5a. This position of the robotic arm was saved in the position database. From this database it is possible to load the position which was saved before. With the help of an orientation element located in the middle of the robot flange (in the tool center point, TCP) the whole positions were set. These positions are essential for the creating of the required robot working activity. One of the achieved positions is described in Fig. 5b.

The simulation served for the verification of functionality of the selected robot model and selected library for the purposes of solving inverse kinematics. The robot collision with the objects in the working space or self-collision was simulated. Recorded unwanted collision states were indicated in red color representing appropriate part of the robot arm. (Fig. 5c) After performing the execution for planned motion of the robotic arm the system evaluates the planned motion as faulty and does not allow the execution. The possibility of describing the trajectory of the robotic arm (Fig. 5d) and the motion in loop was available. Mathematical model of the robot which was selected for the simulation is not enough processed and does not contain the end effector by which the objects in the workspace are caught. Obtaining the basic information about behavior of the simulation is sufficient enough.

The display of output data obtained from the Kinect device was tested in the environment Rviz. Figure 6 represents the real workplace displayed as a point cloud. The point cloud represents a group of a bigger number of points where each point has a defined position in space. Each point has assigned information about color obtained from the rgb camera with which the Kinect device is equipped. In defining the settings in Rviz environment it was also possible to choose the resulting picture from different kinds of displays. Figure 6a displays a standard view of the



*a)* *b)*

*c)* *d)*

*Fig. 5 Examples of the robot simulation, a) initial state, b) achievement of defined point, c) collision between the robot arm and table, d) display of the robot trajectory*

scanned space by using the Kinect device in the Rviz. After turning the view in program (Figure 6b) it is obvious that points are distributed within the simulated space. From the figures it is obvious that using a single Kinect device is not sufficient for the purposes of detecting and recognizing objects in the monitored area. That is why for the solution of creating an intelligent safety system we will consider a multi sensor system.

## 5. Conclusion

The article presents individual steps of testing the ROS platform utilization for the purpose to propose an intelligent safety system for the robot Fanuc LR Mate 200iC control. We set and verified communication between the robot Fanuc control system and ROS and also between Kinect and ROS. The functionality of the robot selected model was verified in the ROS simulation environment through the simple simulation experiments. Achieved results showed that ROS platform can be

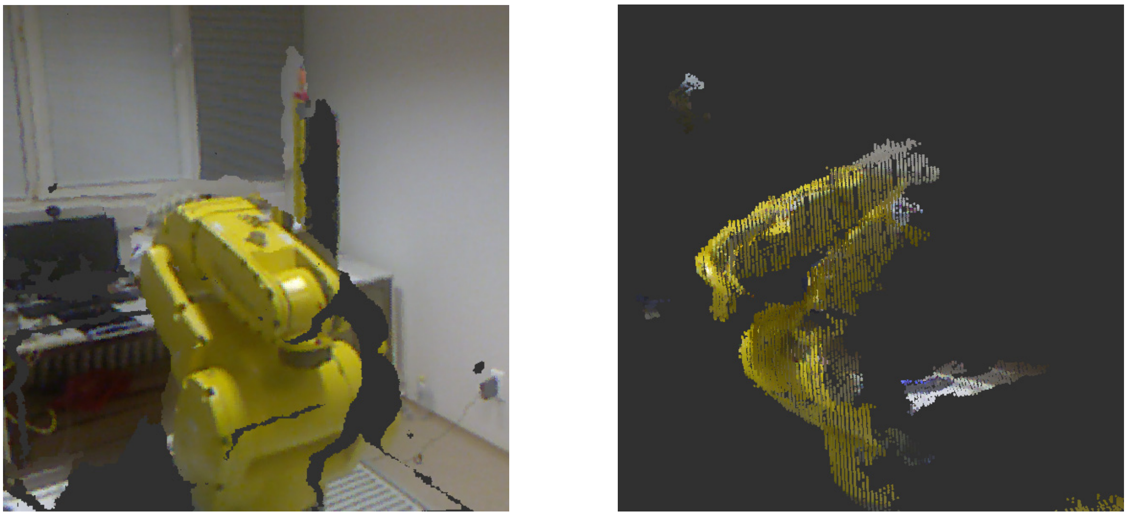


*Fig.6 View of workspace of robot in Rviz, a) direct view, b) at an angle*

used for research of possibilities of controlling the robot working activities in laboratory conditions of the department. On the other hand, the obtained experiences and knowledge led us to specification of other tasks which will be implemented in the coming period:

- To create the new mathematical model of industrial robot with an integrated model of robot´s end effector that allows us to perform a better simulation.
- To setup, activate and verify the communication between the robot FANUC LR Mate 200iC control system and the ROS applications.
- To elaborate methods of recognizing the particular objects in the monitored space and implement created function into the ROS simulation environment. This allows us to create the new and advanced applications for the robot arm motions controlling.

## Acknowledgement

*This paper was created within the framework of a preparation phase of solution of the national project VEGA 1/0632/16 "Research of possibilities of intelligent safety systems solution for hybrid assembly systems based on interactive human-robot cooperation with aid of application of new technologies in the field of sensing of activities on the shared workspace".*

## References

[1] CUBONOVA, N.: Postprocessing of CL Data in CAD/CAM System Edgecam using the Constructor of Postprocessors, *Manufacturing technology: J. for Science, Research, and Production,* vol. 13, No. 2, 2013, 158-164, ISSN 1213-2489.

[2] GROZAV, S. D., CECLAN, V. A., KURIC, I.: *Determining Deformation Force in Orbital Deformation and Comparing it with other Conventional Methods,* International conference on Quality and innovation in engineering and management, 227-230, ISBN 978-973-662-978.

[3] KUMICAKOVA, D., RENGEVIC, A.: Automation of Manufacturing Technologies with Utilization of Industrial Robots, Applied Computer Science (ACS), vol. 11, No. 3, 2015, 6-18, ISSN 1895-3735.

[4] RENGEVIC, A., KUMICAKOVA, D.: Safety Cooperation between Human and Robot, TRANSCOM 2015: 11th European conference of young researchers and scientists, 257-262, ISBN 978-80-554-1047-0.

[5] BRUMSON, B.: New Applications for Mobile Robots, Robotic Industries Association, http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/New-Applications-for-Mobile-Robots/content_id/3362, 2015.

[6] http://rosindustrial.org/

[7] www.eu-robotics.net,

[8] http://www.smartroboticsys.eu/?lang=en

[9] http://www.ros.org/

[10] O'KANE J. M.: *A Gentle Introduction to ROS,* University of South Carolina, Columbia, ISBN 978-14-92143-23-9, 2014.