International Journal of Advanced and Applied Sciences, 3(4) 2016, Pages: 88-92

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

A ROS-based human-robot interaction for indoor exploration and mapping

Hendriyawan Achmad *, Mohd Razali Daud, Saifudin Razali, Dwi Pebrianti

Robotics and Unmanned System (RUS) Research Group, Instrument and Control Engineering (ICE) Cluster, Faculty of Electrical and Electronics Engineering, University Malaysia Pahang, Pekan, Pahang, Malaysia

ARTICLE INFO

Article history:
Received 19 March 2016
Received in revised form
21 May 2016
Accepted 24 May 2016

Keywords:
Human-robot interaction
Robot operating system
Simultaneous localization and mapping
2D LIDAR sensor
Kinect sensor

ABSTRACT

This paper proposes the methods how humans work with tele-operated to perform interaction with mobile robots in order to explore and build an indoor map by utilizing robotic software framework known as Robot Operating System (ROS). However, robotics researchers have limited time during the experiments with definite targets and ultimately there is no choice but to use an integration tools as software framework to avoid re-inventing the wheel. A ROS infrastructure tools are involving together from the file system level to the community level, enables independent decisions about development and implementation. This experiments focus on two major area; the way human delivers the targets coordinate to the robot and the way robot conducts SLAM as a feed back to the human related to current location and occupied map within ROS platform. Hector SLAM plays an important role in 2D localization using 2D LIDAR sensor which is needed by Octomap in order to construct 3D mapping simultaneously using Kinect sensor. The results showed that human-robot interaction using ROS-based tele-operated system for mapping task has able to satisfy and met the quality of the desired results in term of generate 3D map in unexplored area and less time consumption in term of large-scale service robot development.

1. Introduction

The technology of robotics has grown rapidly and applied to many fields, such as industry, education, health, household, entertainment, and the military. An interaction between humans and robots in completing the task is one of the benefits of robots to make people's lives better. In some cases, robots will share the same work space and work closely with humans to accomplish collaboration tasks as part of their day-to-day work (Hinds et al., 2004). The aim of Human-Robot interaction research is to determine models of interaction in term of hardware and software development to obtain effective collaboration between humans as masters and robots as slaves (Salvini et al., 2011). Robots with many sensors used for exploration and mapping tasks without software framework will get a higher level of complexity compared to utilizing of software framework. ROS was developed to address a set of big challenges of complexity being faced when developing large-scale of service robots (Quigley et al., 2009). There are many studies discussing about ROS related to humans – robots interaction to accomplish collaboration tasks (Voisan et al., 2015).

The Simultaneous Localization and Mapping (SLAM) technique is a common issue in mobile robots discussion. The basic idea of SLAM is to use the robot for explorations mission at an unknown location and construct the map base on obstacles surrounding it. Researchers have been studying localization and mapping techniques by using many type of sensor devices, one of which is use laser depth scanners (Pinto et al., 2013).

This paper discusses about the implementation of ROS over human-robot interaction in order to carry out exploration and mapping tasks (SLAM). For localization function we used Hector SLAM algorithm, and for 3D mapping we used Octomap algorithm. Both of techniques are open-source and widely used together with ROS.

2. Materials and methods

2.1. The architecture of proposed systems

A mobile robot, remote station, and network provider are the main components of the system. Remote station is a computer that runs ROS-master to facilitate human during interaction with remote mobile robot. Since ROS is designed with distributed computing, single ROS-master is able to serve multiple ROS-slaves in the same network. Network

---

[*] Corresponding Author.
Email Address: hendriyawanachmad@gmail.com (H. Achmad)

provider take in charge to ensure network availability during runtime, fast network and low latency are indispensable. Mobile robot is equipped with computer which runs ROS-slave to accommodate sensors reading (laser and Kinect) and robot base controlling. PC ROS-slave contain three major ROS package, such as Arduino package for base controller interface handler, Hector package for 2D localization using laser sensor, and Octomap package based on Octree method for 3D mapping using Kinect sensor incorporates with Hector 2D localization which is obtained. The two-way interaction between human-robot with tele-operated mode are operator set the goal position based on obtained map from remote area and robot send the occupied map back to the human as an exploration result. Fig. 1 describes overall components configuration of the system as based on robot operating system (ROS) framework which was used in the experiment.

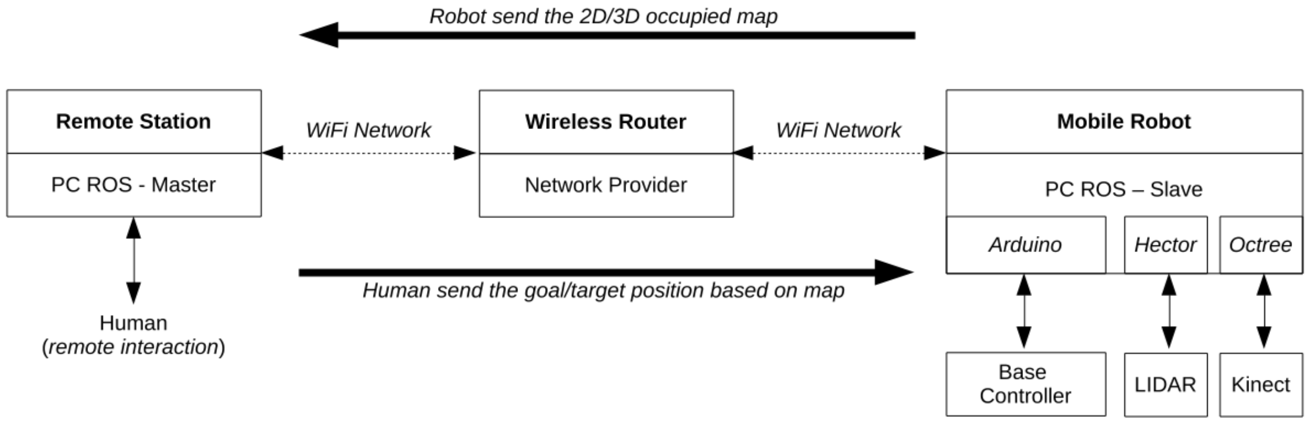


Fig. 1: The architecture of proposed system based on ROS framework

2.2. Non-holonomic differential mobile robot

The type of robot used in this experiment is non-holonomic differential mobile robot (DMR). Right wheel and left wheel has an independent driver of each. For each driver is handled by a PID controller to maintains speed of robots wheel which is desired by PC ROS-slave as seen on Fig. 3. An integrated mobile robot controller in this experiment is equipped by some type of sensors, namely laser sensor: Hokuyo URG-UTM30LX, depth image sensor: Kinect Xbox, odometry sensor: wheels encoder (Fig. 2).

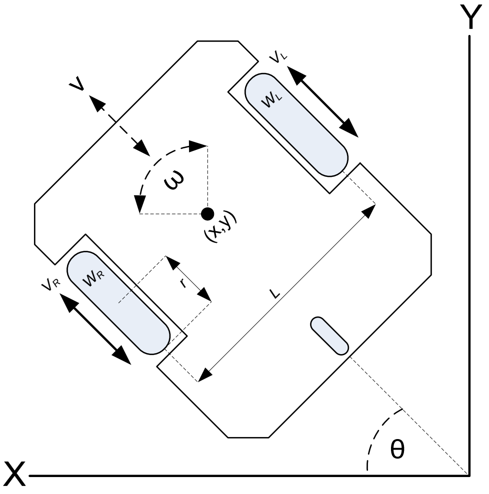


Fig. 2: The kinematics model of experimental DMR

Base on kinematics model of mobile robot as seen on Fig. 2, and then we can calculate desired velocity for both of right wheel (V_R) and left wheel (V_L), and estimate both of relative position $(\dot{x},\dot{y})$ and relative heading$(\dot{\theta})$ of the robot as well using Eqs. 1 to 5.

$$V_R = \frac{v-\frac{L\times\omega}{2}}{r} \tag{1}$$

$$V_L = \frac{v-\frac{L\times\omega}{2}}{r} \tag{2}$$

$$\dot{x} == \frac{r}{2}(V_R + V_L)\cos\theta \tag{3}$$

$$\dot{y} == \frac{r}{2}(V_R + V_L)\sin\theta \tag{4}$$

$$\dot{\theta} = \frac{r}{L}(V_R + V_L) \tag{5}$$

where **v** is targeted linear velocity and **ω** is targeted angular velocity of mobile robot, **L** is distance between left and right wheel (wheelbase), and **r** is radius of wheel.

PC ROS-slave receives messages from Kinect and LIDAR and transmits message to base controller. Kinect sensor produces RGB-D image data, laser sensor produces 2D depth scanning data, and base controller receives two types of velocity, linear and angular in 3-axis (v_x, v_y, v_z, ω_x, ω_y, ω_z). Velocity messages received by base controller are interpreted in two correspondence values of speed (left and right) based on robot kinematics. Fig. 3 shows the configuration of controller for differential mobile robot which is applied by model as seen on Fig. 2. Set points for two PID controllers are V_{left_wheel} (V_L) and V_{right_wheel} (V_R) which correspondence with targeted linear and angular velocity given by PC ROS-slave. Afterward, PID controllers maintain linear and angular velocity of the mobile robot toward the target position.

In this experiment, we have built the real DMR based on kinematics model as seen on Fig. 2 and configuration of controller as seen on Fig. 3 to obtain real-time information regard to our research goal. Fig.4 shows the real DMR which is used during our experiment.

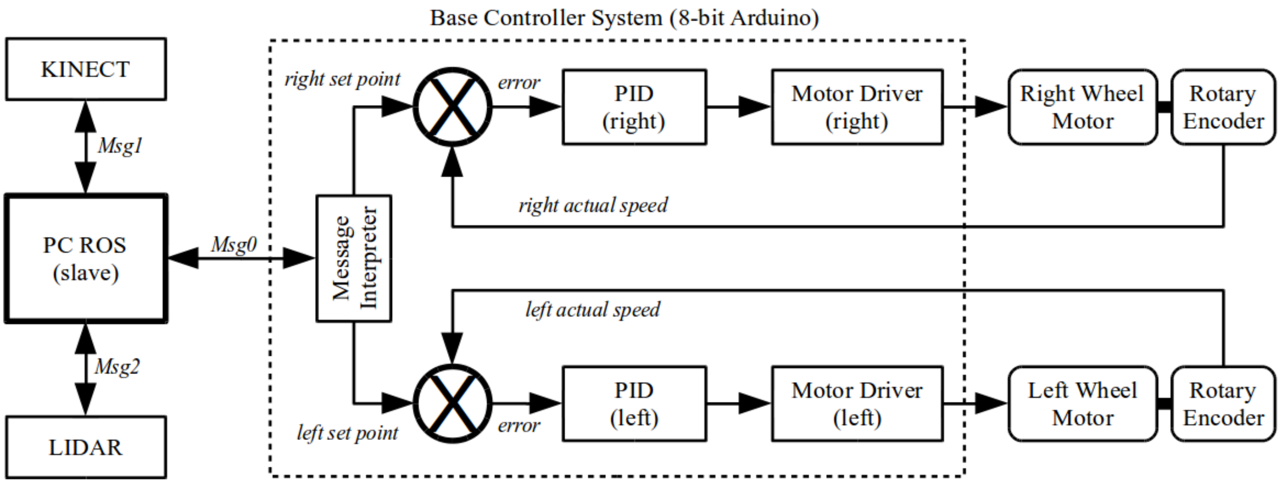


Fig. 3: The configuration of controller for DMR

Fig. 4: Experimental DMR

2.3. Robot operating system (ROS)

Robot Operating System (ROS) is an open source software framework primarily based on UNIX platform for operating robots. It is widely conducted for robotics research in last decade and an obvious overview of ROS has been presented by (Quigley et al., 2009). ROS provides the services user would expect from an operating system, including hardware layer abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple platforms. The primary goal of ROS is to support code reuse in robotics research and development. ROS is a distributed framework of processes (nodes) that enables executables to be individually designed and flexible at runtime. These processes can be grouped into Stacks and Packages, which can be easily shared and distributed. ROS also supports a federated system of code Repositories that enable collaboration to be distributed as well. Fig. 5 shows the basic concept of ROS.

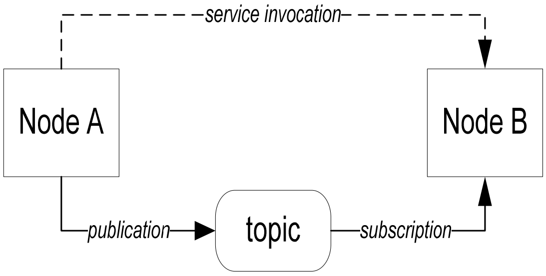


Fig. 5: ROS basic concept

2.4. Hector SLAM

Hector SLAM is an open source implementation tool of the 2D SLAM technique proposed in (Kohlbrecher et al., 2011). The technique is relies on laser 2D scanning data to detect landmarks and build a grid map of the surroundings. Since wheel odometer has a notoriously unreliable in wheeled mobile robot due to slip factor, Hector SLAM is designed to not utilize the odometry data, instead fully relies on fast LIDAR data scan-matching at full LIDAR update rate. Scan matching is playing important role to align current laser scans with other before or with an existing map. Modern LIDAR come up with high update rate and accuracy has significance contributions for the scan matching process that is fast and accurate pose estimation. This technique is based on a Gauss-Newton approach, which no require for a data association search between beam endpoints nor a complete pose search. The algorithm tries to find the rigid transformation $\xi = (p_x, p_y, \psi)^T$ and minimize its value as seen in Eq. 6.

$$\xi^* = arg_{\xi}min \sum_{i=1}^{n}[1 - M(S_i(\xi))]^2 \qquad (6)$$

That is minimizing ξ^* which gives the best alignment of the laser scan with the map. Here, $S_i(\xi)$ are become a function of ξ and represent the world coordinates of scan endpoint $S_i = (S_{i,x}, S_{i,y})^T$. Robot pose in world coordinates are given by Eq. 7.

$$S_i(\xi) = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} S_{i,x} \\ S_{i,y} \end{pmatrix} + \begin{pmatrix} P_x \\ P_y \end{pmatrix} \qquad (7)$$

The function $M(S_i(\xi))$ returns the map value at the coordinates given by world coordinates of scan endpoint (S_i). Once a starting estimate of ξ is given, then the step transformation Δξ can be estimated by optimizing the error calculation according to Equation (8).

$$\sum_{i=1}^{n}[1-M(S_i(\xi+\Delta\xi))]^2 \rightarrow 0 \qquad (8)$$

Implementing first order Taylor expansion of M $(S_i(\xi+\Delta\xi))$ then setting the partial derivative with respect to Δξ to zero delivers the Gauss-Newton equation for the minimization problem as shown by Eqs. 9 and 10.

$$\Delta\xi = H^{-1}\sum_{i=1}^{n}\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right]^T[1-M(S_i(\xi))] \qquad (9)$$

with

$$H = \left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right]^T\left[\nabla M(S_i(\xi))\frac{\partial S_i(\xi)}{\partial\xi}\right] \qquad (10)$$

2.5. Octomap 3D mapping

Octomap is an open-source implementation tool of the probabilistic 3D mapping technique proposed in (Hornung et al., 2013). This technique uses a tree-based representation (octree) to provide maximum flexibility related to the mapped area and resolution. It's performs a probabilistic occupancy estimation to guarantee updatability and to deal with sensor noise. Furthermore, compression methods are applied to ensure compactness of memory usage of resulting 3D map.

An octree technique is using a hierarchical data structure for spatial subdivision in 3D space where each node represents the occupied space in a cubic volume, commonly known as a voxel. Every single volume is recursively divided into eight sub-volumes until reach the minimum voxel size. However, the minimum voxel size determines the resolution of the octree. Sensor readings are performed using occupancy grid mapping technique as introduced by (Moravec and Elfes, 1985). Thus, the occupancy probability $P(n \mid z_{1:t})$ of a leaf node-n provided by the sensor measurements z1:t is estimated according to Eq. 11 (Fig. 6).

$$P(n|z_{1:t}) = \left[1+\frac{1-P(n|z_t)}{P(n|z_t)}\frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})}\frac{P(n)}{1-P(n)}\right] \qquad (11)$$

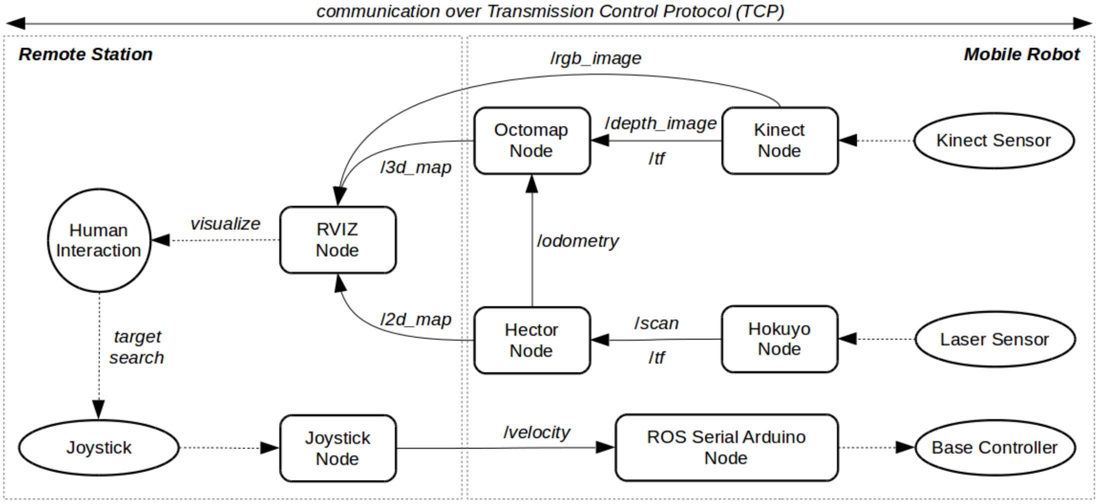


Fig. 6: Experimental ROS ecosystems configuration

3. Implementations and results

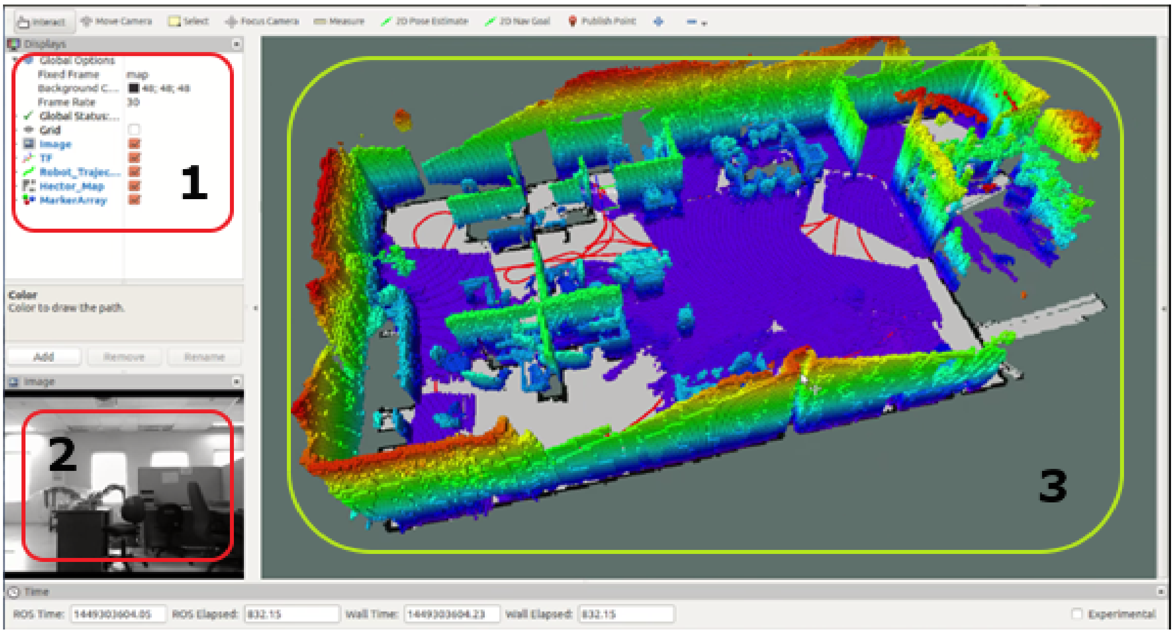
3.1. ROS ecosystems configuration

Based on the proposed architecture of the system as seen in Fig. 1 which is used in this experiment, there are two main sections; remote station section and mobile robot section. All of nodes in the entire system communicate over Transmission Control Protocol (TCP). Beside generates a 2D map, Hector node provides the laser odometry values as well for Octomap node localization in order to generate 3D map. Fig. 6 shows the configuration of ROS ecosystem node that is used in this experiment.

The operator sets the goal position to be explored by robot using Joystick device based on current visual information and location. Base controller on the mobile robot will keep maintain linear and angular velocity of the robot using PID controller time after time to achieve the goal as seen in Fig. 3 and current location (position and heading) is estimated by using Hector SLAM which is use laser scan match and wheel odometry. Beside generates the odometry function, Hector SLAM generates 2D map as well, and the odometry topic (/odometry) will be used by Octomap as localization input in order to generate 3D map instead of estimate localization using depth cloud data from Kinect which will increase the computation cost.

3.2. Constructed map

A 3D map which obtained from Kinect sensor using Octomap was exactly located on the top of 2D map, shows that localization conducted by Hector SLAM was well performed during exploration. RVIZ runs on ROS platform which provides attractive visual interaction facilities to accommodate human-robot interaction impressively. Fig. 7 shows the occupied map during exploration in our experiment and visualized using RVIZ (GUI). All of topics from messages which belongs to nodes are could be monitored by RVIZ and could be analyzed as main purpose in term of human-robot interaction that has been done. Section-1 shows information about topics which were published, section-2 shows RGB visualization from remote site, and section 3 shows 2D/3D constructed map during exploration.



Fixed Frame: map
Background C...: 48; 48; 48
Frame Rate: 30

Color
Color to draw the path.

ROS Time: 1449303604.05 ROS Elapsed: 832.15 Wall Time: 1449303604.23 Wall Elapsed: 832.15

Fig. 7: RVIZ node panel for human-robot visual interface on ROS ecosystem

4. Conclusion

In this experiment, localization methods are become a big issue in term of accuracy and precision. Hector SLAM which come up with laser scan match and wheel odometry fusing method provides best performance so far. 3D mapping using Octomap together with Hector SLAM to do localization instead of use depth cloud localization will provide low memory consumption and low computation cost. In this research we are trying to emphasize the application of ROS for human-robot remote interaction in mapping missions. The results indicate the ease and convenience during system development using ROS, and shorten the time in term of software integration to develop tele-operated mapping as human-robot interaction compared to the efforts when develop same system from scratch.

Acknowledgement

This project is supported by Malaysian Ministry of Higher Education under FRGS (RDU130142) and ERGS (RDU130604).

References

Hinds PJ, Roberts TL and Jones H (2004). Whose job is it anyway? A study of human-robot interaction in a collaborative task. Human-Computer Interaction, 19(1): 151-181.

Hornung A, Wurm K M, Bennewitz M, Stachniss C and Burgard W (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots, 34(3): 189-206.

Kohlbrecher S, Von Stryk O, Meyer J and Klingauf U (2011). A flexible and scalable slam system with full 3d motion estimation. In Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on IEEE: 155-160.

Moravec HP and Elfes A (1985). High resolution maps from wide angle sonar. In Robotics and Automation. Proceedings. 1985 IEEE International Conference on IEEE, 2: 116-121.

Pinto M, Paulo Moreira A, Matos A, Sobreira H and Santos F (2013). Fast 3D map matching localisation algorithm. J. Autom. Control Eng, 1(2): 110-114.

Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J and Ng A Y (2009). ROS: an open-source robot operating system. In ICRA Workshop on Open Source Software, 3(3.2): 5.

Salvine P, Nicolescu M and Ishiguro H (2011). Benefits of human-robot interaction [TC Spotlight]. IEEE Robotics & Automation Magazine, 18(4): 98-99.

Voisan EI, Paulis B, Precup RE and Dragan F (2015). ROS-based robot navigation and human interaction in indoor environment. In Applied Computational Intelligence and Informatics (SACI), 2015 IEEE 10th Jubilee International Symposium on. IEEE: 31-36).