ROS-based Robot Development Toward Fully Automated Network Management

Takayuki WARABINO
Network Operation Automation Laboratory
KDDI Research, Inc.
Saitama prefecture, Japan
warabino@kddi-research.jp

Yusuke SUZUKI
Network Operation Automation Laboratory
KDDI Research, Inc.
Saitama prefecture, Japan
uu-suzuki@kddi-research.jp

Masanori MIYAZAWA
Network Operation Automation Laboratory
KDDI Research, Inc.
Saitama prefecture, Japan
ma-miyazawa@kddi-research.jp

***Abstract*— While the introduction of the softwarelization technologies such as SDN and NFV transfers main focus of network management from hardware to software, the network operators still have to care for a lot of network and computing equipment located in the network center. Toward fully automated network management, we believe that robotic approach will be significant, meaning that robot will care for the physical equipment on behalf of human. This paper explains our experience and insight gained throughout development of a network management robot. We utilize ROS(Robot Operating System) which is a powerful platform for robot development and secures the ease of development and expandability. Our roadmap of the network management robot is also shown as well as three use cases such as *environmental monitoring*, *operator assistance* and *autonomous maintenance of the equipment*. Finally, the paper briefly explains experimental results conducted in a commercial network center.**

Keywords—Network management, automation, robot, ROS, cyber-physical world

I. INTRODUCTION

It is essential for network operators to reduce CAPEX and OPEX of maintained networks. Especially focusing on the reduction of OPEX, automation of network management has been widely discussed. For example, ETSI ENI ISG (Experiential Networked Intelligence Industry Specification Group) [1] tries to define architecture of automate network management and develops necessary standards. The ISG utilizes emerging AI (Artificial Intelligence) technologies. Network management system collects various types of data from network and analyzes it to solve some of the problems of network deployment and operation.

On the other hand, softwarelization technologies such as SDN and NFV are now becoming popular. While the introduction of the softwarelization technologies transfers main focus of network management from hardware to software, the network operators still have to care for a lot of network and computing equipment which are located in network centers. When the network operators want to expand the system, they have to install the equipment in server racks, connect network cables, tun on and configure the equipment etc. These operations are usually carried out by hand. Toward fully-automated network management, we believe that robotic approach will be significant, meaning that robot will care for the physical network and computing equipment on behalf of human.

As pioneers of such robotic approach, IBM researchers developed a robot for temperature monitoring in a data center in 2011 [2][3]. In their trial, thermometer and USB camera were attached at iRobot Create which was a programmable robot based on iRobot Roomba [4]. The robot created a map inside the data center by using the USB camera in the preparation stages. It then autonomously run around in the data center and monitored temperature. Since there was no common robot platform at that time, they had to develop necessary software from scratch. It required long development time and efforts while it lacked for expandability.

In addition, there are other related papers. In [5], thermometer was attached at their original robot and it then monitored temperature in small or medium sized server rooms. In [6], an optical camera was installed to detect LED status of network and computing equipment. In both work, they developed all necessary software from scratch.

This paper reports our work about development of network management robot based on ROS(Robot Operating System) [7][8], which is a powerful platform for robot development. By developing the robot on ROS, we ended up both hardware and software implementation in a few months. In the following sections, our roadmap about network management robot is described in Section II and ROS-based robot design and implementation are explained in Section III. In Section IV, we briefly explain experimental results in a commercial network center as well as insight gained throughout parameter tuning. Finally we conclude the paper in Section V.

II. ROADMAP OF NETWORK MANAGEMENT ROBOT

While robotic technologies are rapidly evolving, there are still many technical issues to realize versatile robots. Then, we think that utilization of the robot for network management will be promoted in an incremental manner. In this section, we describe three possible use cases in which target time ranges from short-term to long-term (Fig.1).

A. Environmental monitoring (short-term)

As the initial step, the robot can be utilized to monitor environmental data (e.g., temperature, humidity, noise or air flow etc.) in the network center. The robot, on which some sensors are attached, periodically goes around in the network center and collects the environmental data through the sensors. If the robot finds out abnormal value, it sends an alert to the network operators.

Temperature is a typical example of monitoring data because it is widely known that high temperature leads to breakdown of network and computing equipment. In addition,

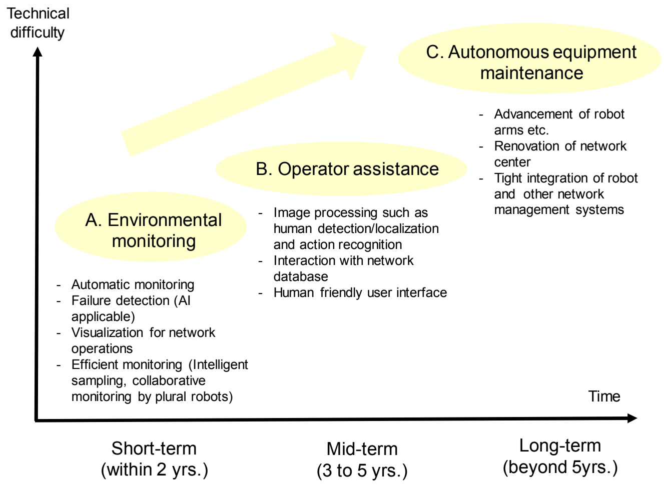

Fig 1. Roadmap of network management robot

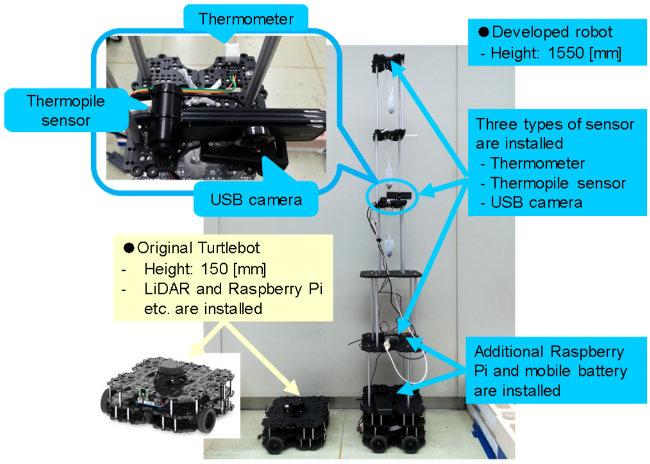

Fig 2. Developed robot(right) and original Turtlebot(left)

the temperature will be more important than ever since density of network and computing resources will be high due to recent virtualization technologies etc. The work of IBM researchers, described in Introduction, corresponds to this use case.

B. Operator assistance (mid-term)

We think that human operators and robots will coexist during the next decade since it will take considerable time to actualize fully-automated network management robot. Then, a possible use case arises in which the robot assists the operator's work.

For example, when the robot finds out the operator in the network center, it firstly identifies the operator by using facial recognition technologies etc. Then, it acquires operation information of the operator from network database and conveys some notice throughout display and/or speaker. Moreover, in an advanced example, the robot standing near the operator recognizes the ongoing operation by using image recognition technologies and then checks whether the operation is performed in an orderly manner and sends a report to network database.

C. Autonomous equipment maintenance (long-term)

In the final stage of the robotic approach, when the robot receives a message indicating that network or computing equipment is broken, it autonomously replaces the broken equipment with new one. To actualize such a scenario, some technical challenges should be tackled. The robot has to remove the old equipment from a server rack and install new one, connect a network cable and push a power button on the equipment etc.

In addition to advancement of robotic technologies, we think that two big changes will be necessary.

(1) Renovation of the network center facilities

Facilities of the network center including the server rack and network and computing equipment etc. should be renovated to be more robot friendly design. The current network center is basically designed for human and it is not necessarily suitable for the robot.

For a simple example, the current server rack can be lock by a physical key. Since the key and key hole are usually small, it is difficult for the robot to insert the key into the key hole correctly. Moreover, the size of the key or location of the key hole differs depending on product of the server rack and it seems to be a big challenge that robot handles all products. To deal with such a situation, the server rack itself should be changed so that the robot can open the key by means of tele-communication. The server rack authenticates the robot through Wi-Fi etc., and it then opens the door after the successful authentication.

(2) Tight integration of robot and network systems

The robot and other network management systems should be tightly integrated. To actualize the use case of equipment replacement, the robot has to get exact location, size and shape of the broken equipment. In addition, the robot has to know where the new equipment is stored and which equipment is used. Moreover, the robot has to know position of the power button and how to configure the new equipment. To go through such a series of steps, necessary information should be maintained in the network management systems and the robot can access to the systems if necessary.

III. ROS-BASED ROBOT DEVELOPMENT

Focusing on the use case of the environment monitoring, we designed and implemented the robot. We utilized the ROS(Robot Operating System) which is an open-source, meta-operating system for robot development. It provides the services including hardware 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 computers.

A. Why is ROS proper?

ROS is a powerful platform for robot development and has many features. Especially in this paper, we focus on the ease of development and expandability. According to ROS homepage [5], it officially supports over 150 robots and over 150 sensors. When a user wants to make a robot, he selects the most suitable robot and sensor, and assembles them. After setting up ROS software on a PC or an onboard computer, he can start a trial. Moreover, when he wishes to add other sensors to the robot, he selects the sensors from the supported list and he can setup the sensors by downloading package software available in the Internet.

Table 1. Major spec. of original Turtlebot waffle pi 3

Item	Value
Maximum translational velocity	0.26 [m/s]
Maximum payload	30 [kg]
Size (L x W)	281[mm] x 306[mm]
Expected operating time	2 [hour]
Single Board Computers	Raspberry Pi 3 Model B and B+
Actuator	Dynamixel XM430-W210
LiDAR	360 Laser Distance Sensor
IMU	Gyroscope 3 Axis Accelerometer 3 Axis Magnetometer 3 Axis

In facts, by developing the robot on ROS, we ended up the implementation of the environmental monitoring robot in a few months. In the followings, our implementation is explained in terms of both of hardware and software.

B. Hardware implementation

We developed the robot hardware based on Turtlebot which is a low-cost robot kit and is compatible with ROS [9]. The implemented hardware is shown in Fig. 2. Base robot used is Turtlebot waffle pi 3 whose major hardware specification is shown in Table 1. The Turtlebot carries omni-directional LiDAR and Raspberry pi. If there is an obstacle on a running route, it can detect the obstacle by the LiDAR and re-select the route using a ROS's route planner functionality.

We extended the original Turtlebot. To increase the height of the robot up to 1550 [mm], we added additional 6 shelves. And following three types of sensors were installed.

- Thermometer
- Thermopile sensor
- USB camera

While the thermometer is used for monitoring temperature in the network center, the thermopile sensor is used for monitoring surface temperature of the server rack. The USB camera is used for recording a still picture of the server rack. All three sensors are installed at a height of 1550 [mm], 950 [mm] and 350 [mm].

C. Software implementation

While utilizing basic ROS packages, we implemented additional functionalities. The configuration of developed software is shown in Fig. 3. Red dotted blocks were implemented by us.

In the followings, 4 basic ROS packages are explained.

- gmapping: It is a ROS wrapper for OpenSlam's Gmapping [10] and provides laser-based SLAM (Simultaneous Localization and Mapping). It uses a particle filer to learn grid maps from laser range data. By running the robot by manual operation using a remote controller, a 2-D grid map can be created.
- move_base: It is a main package for autonomous navigation. Given a goal in the world, local and global planners in the package compute an optimal path and then derive velocity and direction. If the robot detects any obstacle on a running path, the planners recompute an alternative one.

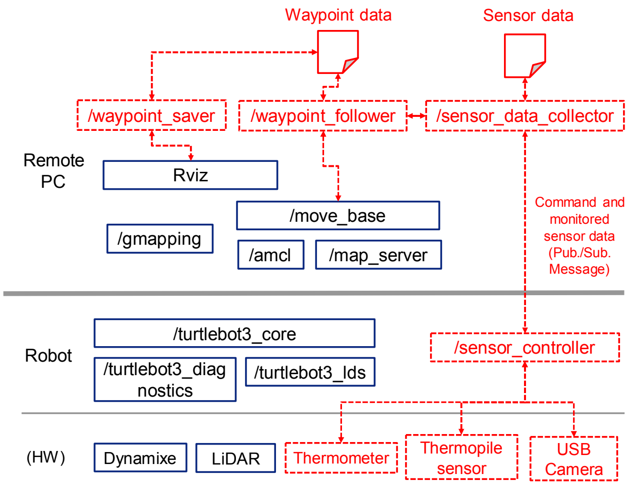

Fig.3. Configuration of developed software

- amcl: It is a SLAM function for the navigation. It is similar to the gmapping explained above, but is different implementation.
- rviz: It is a visualization tool in ROS. It displays a created map, current location of the robot and the detected obstacles etc. Users can check them over the network. It supports both 2D and 3D visualization.

While utilizing these basic functionaries of ROS, we developed additional functionalities which are necessary for the environmental monitoring use case.

- Waypoint saver and follower: For the environmental monitoring, the robot moves to a temporal goal (called waypoint) and collects sensor data. After collecting the data, the robot moves to a subsequent goal. The waypoint saver creates the waypoints by user operation in the preparation stages. The waypoint follower designates the subsequent waypoint to the move_base. In our implementation, operators can create the waypoints through rviz's GUI interface.
- Sensor data collector: When the robot reaches to a certain waypoint, this function collects the data from the embedded three sensors (i.e., thermometer, thermopile sensor and USB camera) and stores them. Format of the data is text or image. In our implementation, ROS's publish/subscribe message service was utilized for exchanging commands and data between the robot and a remote PC. This implementation contributes to reduce the development cost compared to traditional socket programming.

IV. EXPERIMENT AND TUNING IN COMMERCIAL NETWORK CENTER

A. ROS parameter tuning

When a user wants to move the robot on the certain environment, it is indispensable to tune some parameters. Tuning guide is available on web sites [11] and some papers discuss tuning methodologies [12].

While there are some difficulties to move the robot in the network center, this paper explain about two points.

Fig.4. Grating

(1) Ruining on gratings

Grating is a lattice shaped floor tile and is installed for ensuring air flow (Fig. 4). When the robot runs on the grating, it tends to make a rocking motion. This phenomenon often occurs when the robot is approaching a goal. Beside the goal, it moves back and forth in small motions to reach the goal at the designated exact location and direction.

To mitigate the effect, some parameters of local planner and move_base were tuned (Table 2 and Table 3). The *latch_xy_goal_tolerance* was set to be True, meaning if the robot reaches the goal location it will simply rotate in place. In addition, to detect the rocking motion quickly and stop the robot, related parameters such as *oscillation_reset_dist* and *oscillation_timeout* were tuned.

(2) Flatness of the network center

In the network center, a lot of similar shape racks align and the scene is less features. In such an environment, it becomes difficult to perform SLAM precisely.

To improve the accuracy of SLAM, parameter tuning of gmapping and amcl was conducted (Table 4 and Table 5). In the tuning, we increased update intervals of particle filter. Meaning *lstep* in a gmapping package and *update_min_d* and *update_min_a* in an amcl package were reduced. And some *odom-alpha* values, which are related to expected noise of odometry information, were increased.

B. Experiments of the environmental monitoring use case

Using the tuned parameters, experiments were conducted in a KDDI's commercial network center. In the experiments, the robot run autonomously once in every hour according to a created waypoint list and collected sensor data. While size of one room in the network center was approximately 1,000 [m^2], the experiments were conducted in a 1/4 size area of the room.

Experimental results are briefly summarized below. Details of that can not be shown due to the security reasons. Throughout the experiments, hotspot or coldspot (i.e., spot where temperature is remarkably high or cold) could not be observed. This indicates that cooling system of the network center worked properly. In addition, we confirmed that surface temperature of some server racks, which was observed using the thermopile sensor, was higher than that of other server racks. We fed back these results to the responsible network operators.

Moreover, since it took about 45 [minutes] for the environmental monitoring in a 1/4 size area of the room, it is

Table 2. Tuning parameter of local planner

name	Default value	Tuned value
latch_xy_goal_tolerance	False	True
oscillation_reset_dist	0.05	0.2

Table 3. Tuning parameter of move_base package

name	Default value	Tuned value
oscillation_timeout	10.0	2.0

Table 4. Tuning parameter of gmapping package

Name	Default value	Tuned value
lstep	0.05	0.005

Table 5. Tuning parameter of amcl package

name	Default value	Tuned value
update_min_d	0.2	0.05
update_min_a	0.2	0.05
odom-alpha-(1-4)	0.1	2.0

estimated that it takes 3 [hours] for that of the entire room. Then, we will try to investigate methods to reduce the monitoring time as future studies.

V. CONCLUSIONS

This paper reported our work about development of network management robot. We selected ROS as a development platform and this absolutely reduced our burden of the development. It was also confirmed that the developed robot could work properly even in a commercial network center. The paper shared some knowledge which was obtained throughout the development and experiments. On the other hand, we are still standing at the entrance of the robotic approach. We will continue the research and development activities aiming for further stages such as operator assistance and autonomous equipment maintenance.

REFERENCES

[1] ETSI ENI ISG (Experiential Networked Intelligence Industry Specification Group): https://www.etsi.org/technologies/experiential-networked-intelligence

[2] C. Mansley et al, "Robotic Mapping and Monitoring of Data Centers," Proceedings of 2011 IEEE International Conference on Robotices and Automation, May, 2011.

[3] J. Lenchner et al, "Towards data center self-diagnosis using a mobile robot," Proceedings of the 8th International Conference on Autonomic Computing, June, 2011.

[4] https://www.irobot.com/about-irobot/stem/create-2

[5] S. Yazaki et al, "S3R: Automated Temperature Measurement in Small and Medium Sized Server Rooms by Using A Tiny Computer and Self-driving," Journal for Academic Computing and Networking, No.19 pp.114-121, 2015

[6] W. Choi et al, "SCOUT: Data center monitoring system with multiple mobile robots," Proc. of The 7th International Conference on Networked Computing and Advanced Information Management, 2011.

[7] http://wiki.ros.org/

[8] https://github.com/ros

[9] https://www.turtlebot.com/

[10] https://openslam-org.github.io/

[11] http://wiki.ros.org/navigation/Tutorials/Navigation%20Tuning%20Guide

[12] http://kaiyuzheng.me/documents/navguide.pdf