Towards Developing a Teleoperation System for a Disaster Response Robot

Udaka A. Manawadu¹, Hiroaki Ogawa², Keito Shishiki², Yuki Funayama¹, Hiroto Sasaki², Takuma Suzuki², and Keitaro Naruse^{2*}

¹Graduate School of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Japan.

²Department of Computer Science and Engineering, University of Aizu, Aizu-Wakamatsu, Japan.

Abstract. Standard Disaster Robotics Challenge of World Robot Summit (WRS) aims to test the ability of robots that can be used as disaster response robots. Robot Engineering Lab of the University of Aizu is developing a robotic system to address challenges in the WRS. The competition has five stages, and the teleoperation robotic system had to be developed to satisfy the requirements of each challenge. REL uses a disaster-response robot called Giraffe, which has the capability of traveling in hard terrain. Open Robot Technology Middleware uses to integrate all of the subsystems inside the robot. Each subsystem has different tasks that process video data, RGB depth data, Point Cloud Data, sensor data and, feedback data. The Robotic system includes 6 cameras and NDI Software Developer Kit used to transmit and view video streams remotely. The video stream from each camera can be viewed separately, and it gives wider control over the robot for the operator. The teleoperation robotic system was tested during a robot demonstration held in Fukushima Robot Test Field, and results were analyzed according to the WRS 2018 competition guidelines. The results were at an acceptable level.

1 Introduction

In this paper, we are going to describe how the Robot Engineering Lab of the University of Aizu (REL) developed a Teleoperation (Tele-Op) System for a disaster-response robot to face the Standard Disaster Robotics Challenge of World Robot Summit (WRS). The aims of the WRS are accelerating the social implementation, and research and development of robots in real life, social and industrial fields [1]. The categories of WRS were created based on the incidents that happened after the 2011 Tōhoku earthquake and tsunami. After the earthquake of 2011, a large-scale nuclear accident occurred in Fukushima Daiichi Nuclear Power Plant. At the nuclear power plant, most of the restoration work was being carried out manually at that time. Therefore the nonexistence of robotic systems that could address such a situation became an issue [2]. Developing robotics systems that can be used for such systems has become a timely manner.

The teams participating in WRS should prepare their system for five stages: Negotiate, Catwalk, Meter/Valve, L-Shaped Obstacle on Uneven Ground, and Large-area Inspection. Figure 1 shows each of the tasks given in the challenge [1].

Participating in such competition is difficult, especially managing a large team and developing software in a distributed environment. This paper focuses on describing how we manage our team for the WRS, hardware and software selection for developing the robotics system,

and how this development helps to collaborate between academia and industry.

2 Methodology

Since the competition tasks varied in a wide range, team management and software development throughout a large team was a challenging procedure. Usually, about 10 students in REL were involved in software development and carrying out the robot's operations. When selecting the hardware for the robotic system REL, consider the selected hardware's accuracy and availability. For software development, REL considered reliability and high responsiveness when building software for the robotic system.

2.1 Managing Teams

Firstly, the students in REL were divided into two main teams, a team that was dealing with actual machines (STM) and a team that was dealing with simulation environment (SIM). Each team was working towards different tasks. Since the tournament is held at least once a year, students worked together to improve robots and software for the challenge. Each of the teams was led by a final year undergraduate (B4) or a student studying for the master's (M1), and the rest of the students (1st, 2nd and 3rd year undergraduate students) were in charge of robot operators for each task.

If B4 and M1 students found research themes and topics related to the tasks in WRS, they continuously updated the robot software by adding new features according to

*e-mail: naruse@u-aizu.ac.jp

Figure 1. Each of the tasks given in the WRS. In the brackets, the task code is given. (a) Negotiate (Task MAN1): stage to test whether the robot can pass through a narrow space well. (b) Meter / Valve (Task DEX1): A stage that reads the meter’s value by turning the valve or lever. (c) Catwalk (Task MOB1): A stage that climbs stairs and is tested for stability on unstable scaffolding. (d) L-shaped Debris on Uneven Ground (Task DEX2): a comprehensive stage, such as traversing the rattling scaffolding and grasping and operating the pipe with the arm. (e) Large Area Inspection (Task EXP1): A stage that reads the QR code affixed to the wall, photographs the wall, and creates a three-dimensional map of the wall [1]

their research. While operating the robot, the junior students would gradually learn the robotic system and deepen their understanding of the disaster response robotic field. Figure 2 shows the learning cycle of the student who participated in the challenge.

The team usually practiced the robot in an environment that imitated the previous versions of WRS challenges. This environment allows students to find improvements to the robot and practice with higher accuracy. By practicing the operation daily, it is possible to understand the performance lacking areas of the robot and shape-up the controlling accurately. The issues that come when practicing were divided into long-term, medium-term, and short-term issues, and they were solved according to the difficulties. There are two ways to get a good score in the competition.

One is to improve the ability of the robot, and the other is to improve the skill of operation of the operator. In the future, the goal is to improve the capabilities of robots so that anyone can use them easily, but considering the competition, it is also important to get good results by covering them with operations. The team also considered the safety

Figure 2. Usually, students in REL go through the learning process from practicing the robot to adding improvements to the robotic software.

of themselves when handling the robot since the Giraffe (figure 3) weighs about 80 kilograms, and it is required to wear helmet and safety shoes when operating the robot. The team also held seminars every year to practice safely.

2.2 The Robot

In REL, we are using a disaster-response robot called ‘Giraffe’ as the crawler. Giraffe was made by Aizuk Pvt. Ltd, with collaboration with REL. Giraffe is a crawler-type robot that has two main crawlers and four sub crawlers. Four sub crawlers work independently, and it enables to run through rough terrain, and the flexibility of the crawlers made it easy to face challenges of WRS (Figure 3) [3]. Apart from that, it consists of an inbuilt PC that runs in Linux Operating System.

Figure 3. Giraffe robot doing “Catwalk” task in the 4th University of Aizu Robot Symposium in 2019 [4].

The Giraffe was modified to connect an arm externally and connect another PC externally. KINOVA JACO arm with 6 degrees of freedom Curved Wrist is used as the external arm of the robot. The JACO arm’s flexibility is heavily useful for tasks like DEX1 and DEX2 in the competition (In figure 1). Jaco arms are usually used in Service Robotics, and Assistive Robotics due to their flexibility and finger capability [5]. As in figure 4, a rubber was put on the robot finger to give an extra grip for holding or turning an object.

Apart from the arm, 6 cameras were installed in different directions enabling total control of the robot remotely. There are two front and back cameras and two fish-eye 360° cameras to inspect the surroundings more widely.

Figure 4. The Robotic system created by REL consisting of all the hardware arrangements, and the Tele-Op side consists of all the controlling components.

The robot was mainly controlled by using those four cameras. The other two cameras were placed on the Jaco arm (figure 4). One is a web camera for the meter reading task, and the other is Intel RealSense Depth Camera D435 to measure and map depth in 3-D space in the Large Area Inspection task. Ease of use, accuracy, small size, and availability of the Software Development Kit (SDK), were considered when choosing RealSense D435 [6].

The robotic system is integrated with three Personal Computers (PC) in a single network. All PCs were run in a Linux environment.

- Crawler PC: The PC inside the Giraffe that controls crawlers (A less powerful PC).
- Arm PC: The PC that was put on the Giraffe (figure 4) to handle the Jaco arm, six cameras, and then transfer the control signal to the Tele-op PC.
- Tele-Op PC: The 3rd PC (laptop) for the operator that connects the controller and shows the video stream of the camera.

On the Tele-op side, the operator operates the robot remotely using the controller connected to the laptop. We need to move the robot remotely in the competition. Therefore, the operator needs to grasp the situation of the robot while watching the images of the cameras attached to the robot and perform various tasks. Either WiFi or network wires can be used as the communication method between the robot and the Tele-op laptop. In the competition, we are using network wires as the communication method due to the high stability of the system. We called the whole robotic system in figure 4 as "Spider2020".

2.3 Software Development

The control and software integration of the robot is managed by the Open Robot Technology Middleware (OpenRTM-aist), and the camera viewer in Tele-op side

is developed by an SDK called NDI. OpenRTM-aist is a distributed component middleware for robot technologies developed and distributed by the National Institute of Advanced Industrial Science and Technology, Japan [7]. The team used OpenRTM-aist instead of Robot Operating System (ROS) due to the high responsiveness of the OpenRTM-aist than ROS systems. In OpenRTM-aist, the function elements in the system are called RT components (RTC). By connecting components, it is easy to configure a system that applies robotic technology. For example, we can issue commands from a controller to move the robot or check the robot's status at a glance on a PC. This can be accomplished by RTC communicating with each other. One can explicitly check the connectivity of RTC. By connecting the desired RT components, one can reflect the behavior of the robot. Figure 5 shows the OpenRTM-aist system diagram of our system.

Figure 5. The OpenRTM-aist system diagram that used for the robotic system.

According to the diagram, it is easy to distinguish how the signal flows through the system. RTC_DS4_Controller0 is a component that gives the controlling signals to RTC_Spider2020_Motion0 component

controls arm and motion controls then signals pass to RTC_Spider2020_Crawler0 and RTC_Spider2020_Jaco0, which are the components of Spider2020's motions and Arm motions in order. Task MAN1 and Task MOB1 (figure 1) Jaco arm was not needed, and RTC_Spider2020_Jaco0 was disabled for those tasks.

Figure 6. Left: Using NDI, operator operating Spider2020. Right: Multiple screens that show different camera views (One camera view shows meter value while other camera view show surroundings).

NDI, which stands for Network Device Interface, is an SDK that can transmit a video stream via a network. By introducing NDI, we could obtain live video data from the robot's camera to the Tele-Op laptop, which is needed for remote control. We have developed a software that allows the operator to select the information of a camera to be acquired from multiple cameras by using this SDK to realize a system that allows the operator to easily understand the situation around the robot (Figure 6). Multiple processes are running simultaneously, and it is possible to get the information obtained by all the cameras in real-time. Each camera is assigned a shortcut key on the keyboard, and the operator can select a camera at will.

We also used Realsense SDK and OpenCV to compete in competitions where we survey large areas because we can store data about our surroundings. The information around the robot can be acquired as a point cloud of data. In this way, RealSense has the ability to recognize information about the object in front of it more clearly.

Figure 7. Summarized Spider2020 system overview.

Finally, the whole system can be summarized in the following figure 7. Hardware devices are colored by red,

OpenRTM-aist is colored by blue, OpenRTM-aist components are colored by light blue, and the library associated with NDI is colored by light yellow.

3 Results

Figure 8. Pictures were taken at robot demonstration at Fukushima Robot Test Field.

The results for Meter/Valve (Task DEX1) and Large Area Inspection (Task EXP1) is shown in Figure 9.

Figure 9. (a) image shows results for Meter/Valve; Deep learning was used for identified meter value automatically [8]. (b) image shows current results for Large Area Inspection.

Throughout the competition, we are finding issues and making improvements, and updating the software daily. We believe that this will lead to our own researches, the promotion of the robotics industry, and contributions to society and reconstruction.

4 Conclusion

This paper describes how REL managed to build the Spider2020 robotic system. Flexible crawler and arm and

camera sensors that cover a wide area make Spider2020 a complete disaster-response robot. However, hardware improvements are not enough to use a robotic system in a disaster, and software improvements should be made simultaneously. Since the robot's task range has a wide variety, many subsystems are running inside the system. Dividing each task for different teams improved the efficiency of students since each team needed to improve background knowledge of a specific narrow research area. Identifying issues, categorizing them, and continuously improving the system make a solid foundation to participate in events such as WRS.

References

- [1] T. Kimura, M. Okugawa, K. Oogane, Y. Ohtsubo, M. Shimizu, T. Takahashi, S. Tadokoro, *Competition task development for response robot innovation in World Robot Summit*, in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (2017), pp. 129–130
- [2] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima et al., *Journal of Field Robotics* **30**, 44 (2013)
- [3] N. Baba, *Disaster Response Robots 'Giraffe' Developed by the Members of Fukushima Consortium of Robotics Research for Decommissioning and Disaster Response*, in *4th International Forum on the decommissioning on the Fukushima Daiichi Nuclear Power Station* (2019), p. 10
- [4] *The 4th university of aizu robotics symposium*, <https://www.u-aizu.ac.jp/en/information/post-20171151.html>
- [5] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire, L.J.C. L'Ecuyer, in *Rapid Automation: Concepts, Methodologies, Tools, and Applications* (IGI Global, 2019), pp. 693–719
- [6] A. Grunnet-Jepsen, J.N. Sweetser, P. Winer, A. Takagi, J. Woodfill, Intel Support, Intel Corporation: Santa Clara, CA, USA (2018)
- [7] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, Woo-Keun Yoon, *RT-middleware: distributed component middleware for RT (robot technology)*, in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005), pp. 3933–3938
- [8] Y. Funayama, K. Nakamura, K. Tohashi, T. Matsumoto, A. Sato, S. Kobayashi, Y. Watanobe, *Artificial Life and Robotics* pp. 1–11 (2020)