Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 9-10, 2024

Development of Integrated Control System for the Nuclear Disaster Response Robot

Hyeokbeom Kwon[a], Ki Hong Im[b], Yonggyun Yu[a, b], Jongwon Park[a, b*]
[a]*University of Science & Technology, Gajeong-ro 217, Yuseong-gu, Daejeon, Republic of Korea*
[b]*Korea Atomic Energy Research Institute, Daedeok-daero 989-111, Yuseong-gu, Daejeon, 04535, Republic of Korea*
[*]*Corresponding Author: jwpark@kaeri.re.kr*

Keywords: User Interface, Robot, ROS, GUI, Hydraulic Manipulator

1. Introduction

Nuclear disaster response robot ARMstrong[1], is a mobile dual arm manipulator actuated by hydraulic system. The ARMstrong has 16 joints driven by micro HPU (Hydraulics Power Unit)[2]. Joints are actuated by hydraulic cylinder and hydraulic motor, which controlled by proportional directional control valve. Rotary encoder sensor is used for measuring joint's angular position. For energy efficiency, micro HPU uses variable speed control to minimize bypass flow rate. Miniature controller transmits its joint position utilized as reference input.

In summary, ARMstrong robot has 16 rotary encoder sensors to read, 16 proportional directional control valve and a micro HPU's motor drive to control. It produces lots of data to monitor and manage.

Therefore, it is necessary that operator could check the status of ARMstrong at a glance and operate it easily. In our previous design however, since the system of ARMstrong is operated separately and its state is not visualized, the forementioned requirements could not be fulfilled.

For those reasons, we developed the ARMstrong System Software, integrated control system for ARMstrong.

2. System configuration

The ASS compromises three major components, ARMstrong Integrated Controller, ARMstrong Control Panel, miniature controller. For ARMstrong Integrated Controller and ARMstrong Control Panel, Ubuntu 20.04 (AMD64) is used as operating system, and is developed in Python3.8. For message passing and communicating between components, ROS (Robot Operating System) is used. For configuring GUI for ARMstrong Control Panel, Pyside6.3, a Python binding for Qt, is utilized.

3. System structure

The abstract structure of ARMstrong System Software is described at Fig. 1. Blocks tinted with light gray denote user input elements, while locks tinted with dark gray denote information for user to monitor. Arrows with black filled heads denote data flow within devices, bold arrow with dotted line indicates data flow between devices. Labels with these arrows specify the transmitted data.

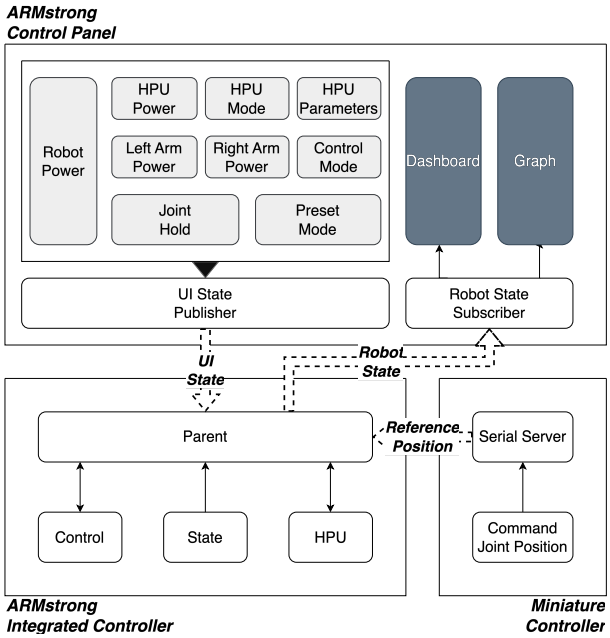

Fig. 1. Schematic of integrated control system, the ARMstrong System Software.

3.1. Miniature controller

The ARMstrong is equipped with miniature controller[3], facilitating fast adaptation of operator to perform various indeterminate tasks. Operator holds this controller, takes poses for ARMstrong to follow.

Miniature controller reads rotary encoder sensors' values of its own and organizes, sends it to ARMstrong Integrated Controller as name of *Reference Position*.

3.2. ARMstrong Integrated Controller

The ARMstrong Integrated Controller directs slave robot's action. This listens command from master robot and remote PC, reads slave robot's current position and controls valves and micro HPU. Due to the time-costly nature of reading and controlling devices, and our aim for high-frequency sensing of joint positions and controlling valves, we implemented process-based parallelization. ARMstrong Integrated Controller has four loops running parallelly, asynchronously on different processes. These loops handle sensing joints' position, controlling joints, send command to HPU, and the parent loop. The parent loop receives commands

from other components: *Reference Position* and *UI State.* It also aggregates states from its child loops and distributes to them. Since each process has separate memory space, pipe is used for inter-process communication (IPC). The parent loop sends the Robot State, a comprehensive set of states collected from its child loops to ARMstrong Control Panel. The *Robot State* includes joint positions, reference positions, command outputs, errors for each joint, HPU's feedback and targeted rpm, targeted current, and HPU's motor driver temperature.

3.3. ARMstrong Control Panel

To provide convenient and intuitive control, we configured GUI, the ARMstrong Control Panel (Fig. 2). The ARMstrong Control Panel listens robot's state and displays it as graph and table. User can send command to slave robot through this interface, toggle valves' and HPU's power individually or simultaneously. User can switch HPU's control mode and adjust control parameters. User can also choose each joint to hold at current position or specify reference position to be set at preset position.

Fig. 2 is shown for explanatory purpose. In this figure, the HPU and valves are powered off because main power is shut down, despite buttons are toggled on. The HPU is set to manual RPM mode.

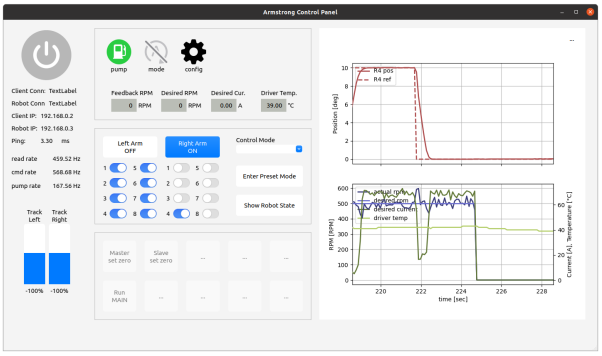

Fig. 2. Actual footage of ARMstrong Control Panel.

4. Conclusion

In this paper, we have discussed about the integrated system and its GUI, the ARMstrong System Software. With this, we anticipate that monitoring and managing the status of the robot will be considerably simplified.

Our future work will focus on optimizing the GUI based on Human-Computer Interaction (HCI) principles.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Science and ICT)(No. RS-2022-00144468).

REFERENCES

[1] J. Park, J. Lee, and K. Im, Development and Field Test of a Nuclear Disaster Response Robot, ARMstrong, Transactions of the Korean Nuclear Society Spring Meeting, 2023.
[2] J. Lee, J. Park, K. Im, and G. Shin, Design and Implementation of a Micro HPU for Nuclear Accident Response Robot, Transactions of the Korean Nuclear Society Autumn Meeting. 2023.
[3] J. Park, J. Park, and S. Jung, Master Robot Design of Hydraulic System-based Dual Arm Manipulator, Journal of Institute of Control, pp. 154–155, 2022.