

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Management of a Single-User Multi-Robot Teleoperated System for Maintenance in Offshore Plants

Seungyeol Lee and Jeon-Il Moon

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.69351>

Abstract

This chapter proposes a new approach to management method of a single-user multi-robot teleoperated system for maintenance in offshore plants. The management method is designed to perform a 1:N mode (here, “1” refers to the number of operators and “N” denotes the number of slave robots), in which a single operator teleoperates a number of slave robots directly to conduct a maintenance task, or in an autonomous cooperation mode between slave robots in order to overcome the limitations of the aforementioned 1:1 teleoperation mode. The aforementioned management method is responsible for the role sharing and integration of slave robots to divide the operation mode of the slave robots into various types according to the operator’s intervention level and the characteristics of the target maintenance task beforehand and to perform the target maintenance task using the robot operation mode selected by the operator.

Keywords: multi-robot, teleoperated system, management, maintenance, offshore plants

1. Introduction

A typical teleoperated robotic system consists of a master device that collects motion commands from a remote user for a slave robot, and a slave robot that follows those commands. The 1:1 teleoperation method is most frequently used in typical teleoperated robotic systems, and involves a master device that collects target task commands from an operator to the slave robot that carries out those commands [1, 2]. For example, Heikkila et al. proposed functional design of a manufacturing robot cell [3]. Yamada et al. introduced construction telerobot system with virtual reality [4]. Zhao et al. developed a construction telerobotic system that has wide

applications in restoration work in stricken areas [5]. Kwon et al. developed a microsurgical telerobot system [6]. Geerinck et al. introduced the operability of an advanced demonstration platform incorporating reflexive teleoperated control concepts developed on a mobile robot system [7]. The 1:1 teleoperation mode that can be expected in offshore plants, includes operations conducted by a robot manipulator, such as manipulating door knobs (open, close) and pressing buttons and other types of switches; operations of mobile platform movement, including evading obstacles; and inspection operations, such as inspecting inner pipes, gauges, and oil or water leakage by cameras and other instruments. But 1:1 teleoperation method is unsuitable due to limits of the robotic workspace or power of only one slave robot. When this method is applied to maintenance (e.g., handling large valves) in offshore plants, one slave robot is insufficient for performing the maintenance processes because of the length and weight of the mechanical parts in offshore plants. When two of these systems are used for the maintenance in offshore plants, communication between the operators and cooperation between the robots through the two master devices are not smooth, and this reduces the work efficiency. Furthermore, if user operations are repeated regularly, an operator may lose concentration due to the repetition of simple motion commands [8].

A “shared telerobotics” has recently been proposed in which a general operating command is issued and the robot autonomously performs the specific task [9–20]. Operating on a valve that is located between multiple obstacles and must not come in contact with a nuclear facility, enables an operator to simply provide a direction to the robot and move the robot and its end effector to a desired location to perform the operation through an obstacle avoidance algorithm, without having to meticulously control the robot to avoid the obstacles. In this chapter, a single-operator multi-robot (SOMR, 1:N) teleoperation system that can conduct specific operations with cooperation between multi-robots to overcome the limitations of the 1:1 teleoperation systems is briefly explained. The system presented in this chapter can perform teleoperation with respect to remotely placed robots through a single operator, and is designed to perform particular operations through autonomous cooperation between robots with minimal input from an operator. In particular, software for efficient robot management, such as role sharing and changing between robots in the SOMR teleoperation system, is explained in detail.

2. Framework of single-operator multi-robot system

Figure 1 shows the framework diagram of a system that allows single operator to remotely control multiple units of robots (N ; positive integers). As shown in **Figure 1**, operators transmit action commands to a master device; the commands are converted into action control signals for the robot system through the device and processed through a communication system, which includes the remote-control device, to be delivered to the corresponding module (such as a robot arm, mobile platform, or camera) of the slave robot. In addition, data on the contact force that occurs through interactions between robots and the surrounding environment while in operation, data on the actual robot motions, and data on the current action status are measured by various sensors on the slave robots. The measured data are then delivered to the master device through a communication system to ultimately provide necessary data (such as contact force, robot motion status, and action status) to operators through the controllers and drivers inside the device.

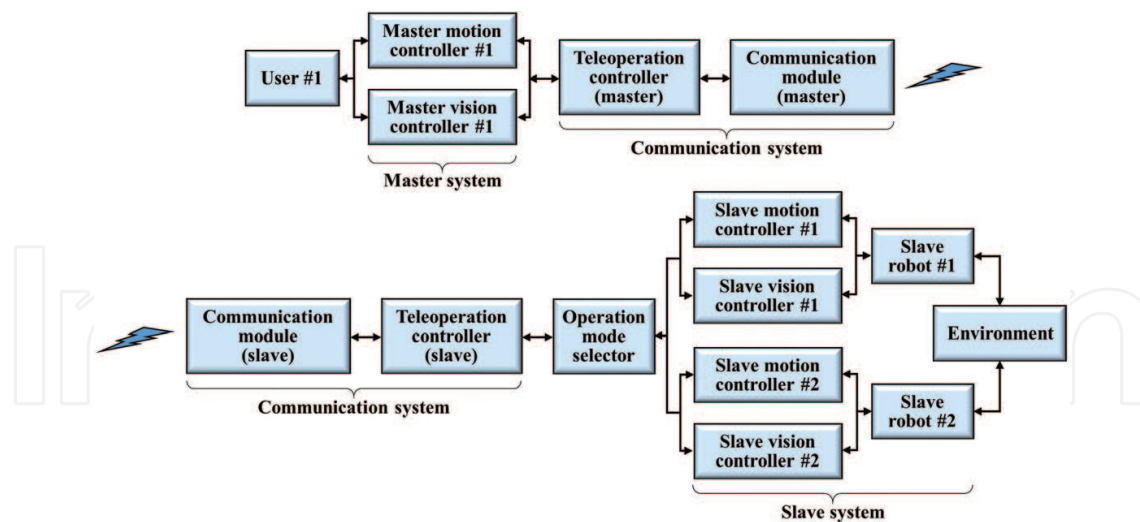


Figure 1. Framework diagram of a single-operator multi-robot system.

In addition, an operation mode selector is included in the slave circuits. This device supports predefined parameters for each of the following four cases: when separating action commands (such as robot motion and measurement Degree Of Freedom) delivered from the operators by the master device, when separating action commands delivered from the master devices by the slave system, when separating data on robot motion and action status measured from the environment by the slave system, and when separating data on the current motion of robots and action status delivered from the slave systems by the master device. Therefore, an operating scenario is determined by the operators controlling each parameter, and action commands are distributed by the operators to control the robots' cooperation. When the action commands are delivered to the slave system, they are distributed by a slave robot to ensure they are delivered to the necessary slave robots. Likewise, the measurement and observation DOF of the slave robots are distributed by the slave system, and when the data are delivered back to the master system, they are distributed by the master system to ensure they are delivered to the appropriate operators.

Figure 2 shows the structure of a typical master system (including master arms), which includes the master motion controllers inside the master system (**Figure 1**) and the signal input-output relationship. As shown in the **Figure 2**, when operators move the handle of the master arm, the rotational displacement of the joint is converted into pulse values through an encoder inside the arm and is converted again into angle values through a decoder. The converted angle values are then subjected to forward kinematics analysis to be converted into desired location coordinates of the rectangular coordinate system to which the slave robot is to be directed, and are subsequently processed through a communication module (remote-controlled) to be delivered to the slave robot. In general, the data on the contact force that occurs when slave robots come in contact with the surrounding environment are measured by force/torque sensors installed inside the robots, and the data pass through the communication module to be delivered to the master controller. The delivered data then undergo an internal conversion process to obtain the desired torque value for the master arm joint and are modified to the current values for each motor required to generate the same torque again. Therefore, the data on contact force measured by the initial slave robot through the driving mechanism

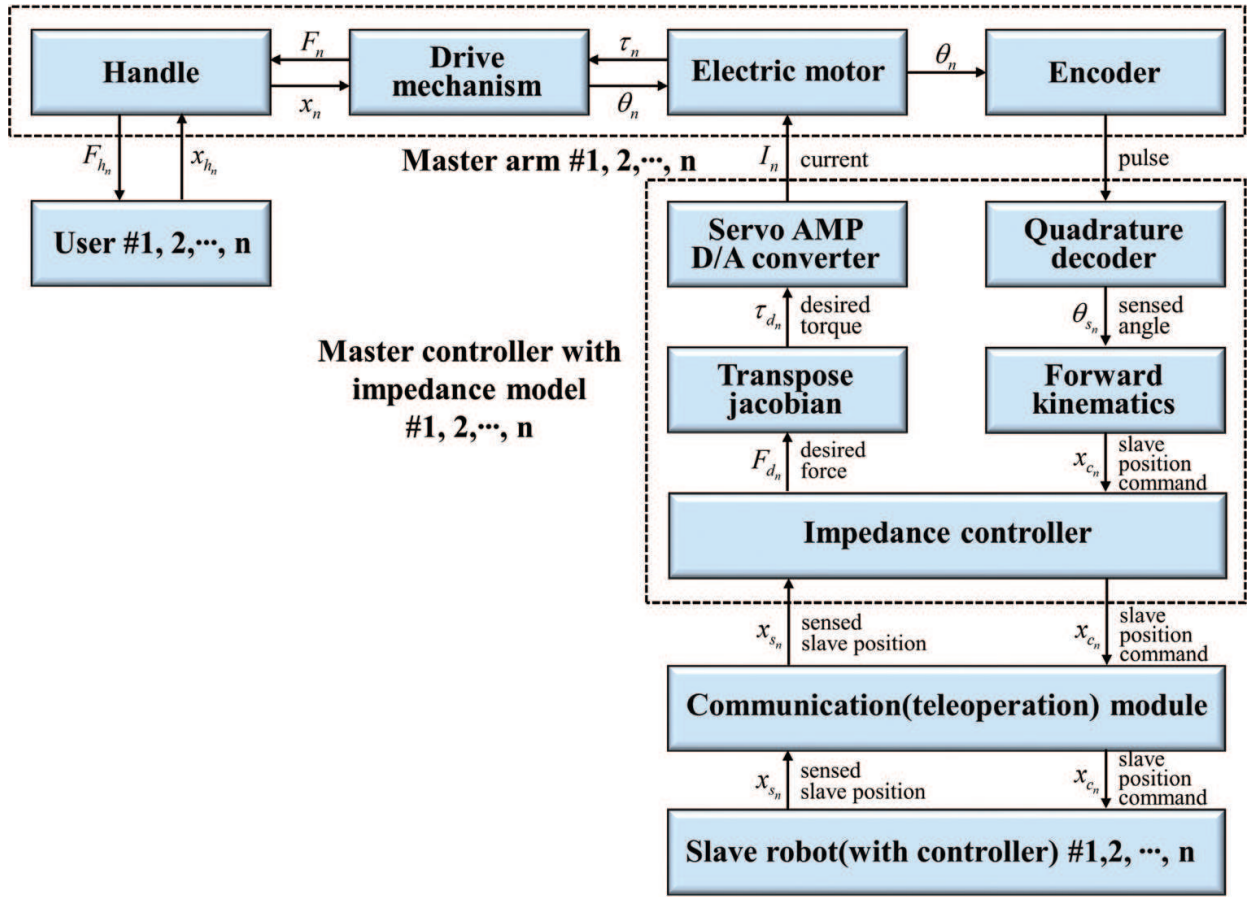


Figure 2. Structure diagram of a master device.

that includes the motor inside the master arm are ultimately converted into the data for the force/torque of the master arm and are delivered to the operators.

Figure 2 also represents the structure diagram of a master device that can provide operators with the data on contact force without force/torque sensors inside the slave robots. This means that the master device differs in the following two aspects: there are deviations between the operators' action commands inside the master controller and the current motion data of the robots and the device estimates the data of the desired contact force from a virtual environment model in Eq. (1), which is determined by impedance parameters. The virtual environment model is determined from the impedance parameters (K , D) and can be adjusted according to the operators' requests. As for a PHANTOMTM haptic device, a haptic equipment that has been launched commercially, its impedance parameters are fixed to recommended values upon factory release, and operators are unable to adjust the values. However, they are able to adjust the degree of force/torque they sense to a certain proportion by parameter " ϵ " in Eq. (1).

$$\epsilon \tilde{F}_d = K \Delta x_{ds} - D \Delta \dot{x}_{ds} \quad (1)$$

where \tilde{F}_d : is the desired contact force estimated from the impedance (virtual environment) model; K , D : are hardness and damping parameters, respectively. ϵ : is the contact force adjusting parameters that are fed back to operators.

$$\Delta \underline{x}_{ds} = \underline{x}_d - \underline{x}_s, \quad \Delta \dot{\underline{x}}_{ds} = \dot{\underline{x}}_d - \dot{\underline{x}}_s \quad (2)$$

To realize the SOMR (1:N) system, the signal input-output relationship for remote-controlled robots is shown in **Figure 3**. When the initial operator issues “commands for robot’s motion (R_u)/robot’s action (P_u),” the commands are processed through a communication system and delivered to slave (robot) systems. Then, as the robots follow the motion/action commands above, the “data on robot’s motion (R_e)/robot’s action (P_e) (current motion status of robots, surrounding environment, and action status),” which are measured through various internal sensors, are processed through the communication system again to be delivered to the master system. These data on the motion/action ultimately provide operators with various necessary data through the internal controller of the master system. This means that the motion/action commands issued from the initial operator and delivered to the robots are executed in the form of robots actually initiating the commands or observing the surrounding environment and situation. The robots then deliver the result of their motion/action to operators to inform them about the execution status of the commands and the current action status. As such, there must be a clear input-output relationship between the motion/action commands for robots that are input by the operators and the motion/action data that are output from the robots, and it must be accompanied by the 1:1 mapping between the input data and output data. Unlike on the SOMR (1:1 teleoperation) system, an operation mode selector is included in the slave system section in the 1:N teleoperated system. Parameters are included within the operation mode selector for determining the 1:N teleoperation mode. These parameters, as shown in **Figure 3**, consist of a parameter (β) that separates the action commands delivered from the master (operator) by the slave and a parameter (γ) that separates the data on the robot’s current motion/action status measured from the operation environment by the slave. Therefore, the 1:N teleoperation mode is determined because an operator selects this parameter, and cooperative control between robots becomes possible because action commands are distributed by the slave robot. The data measured or observed from the environment are distributed by the slave, providing the necessary data to the slaves.

To explain in detail, in the 1:N teleoperation mode, a single operator first establishes a distribution plan for tasks (classified into two cases: distributed as the robot motion task, and measurement task of the surrounding environment and situation by robots; the proportion of cooperation is adjusted while performing the robot motion task and measurement task simultaneously) to be performed by a specific number (N) of slave robots (assumption $N = 2$) and determines the

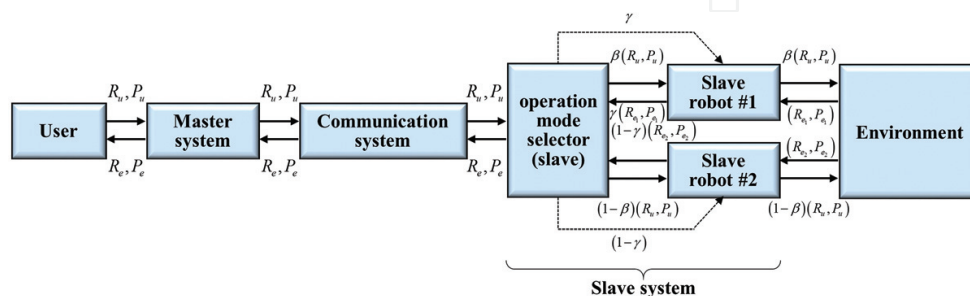


Figure 3. Structure diagram of a master device in a SOMR system.

parameter β within the operation mode selector on the basis of the task distribution plan. Then, the action command delivered to the master device from the operator is processed through a communication system and delivered to each respective slave robots as $\beta(R_{u1}, P_{u1}), (1 - \beta)(R_{u2}, P_{u2})$, as shown in Eq. (3).

$$(R_u, P_u) = \beta(R_{u1}, P_{u1}) + (1 - \beta)(R_{u2}, P_{u2}) \quad (3)$$

where (R_{u1}, R_{u2}) : is the robot action commands input into slave robots 1 and 2; (P_{u1}, P_{u2}) : is the ancillary action commands input into slave robots 1 and 2.

$$R_u = R_{u1} + R_{u2}, P_u = P_{u1} + P_{u2} \quad (4)$$

In addition, there is a parameter γ within the operation mode selector that functions to separate the data on the robot's current motion/action status measured from the environment by the slave. To explain in detail, when the data on the robot's motion (R_e) and ancillary action (P_e) are measured from the sensors inside the slave robots, the data are distributed and measured according to $\gamma(R_{e1}, P_{e1}), (1 - \gamma)(R_{e2}, P_{e2})$ by slave robots 1 and 2 by the parameter γ Eq. (4). Because the data measured by each slave must be delivered to a single operator through communication system and a single master system, they require reintegration to (R_e, P_e) within the operation mode selector, as shown in Eq. (4).

$$(R_e, P_e) = \gamma(R_{e1}, P_{e1}) + (1 - \gamma)(R_{e2}, P_{e2}) \quad (5)$$

(R_{e1}, R_{e2}) : is the data on robot's motion provided to slave robots 1 and 2; (P_{e1}, P_{e2}) : is the data on ancillary action provided to slave robots 1 and 2.

$$R_e = R_{e1} + R_{e2}, P_e = P_{e1} + P_{e2} \quad (6)$$

3. Integrated robot management software

The SOMR system allows an operator to remotely control multi-robots and aims to perform tasks that cannot be done with a single robot (e.g., large volume required or heavy objects handle works) through the cooperation of multiple robots. In the above system, each robot has a single robotic manipulator mounted in an upper part of a single mobile platform, as well as various sensors, including a camera. The development goal of the SOMR teleoperation system is to improve work efficiency by performing particular tasks with multiple robots where a single robot would be either inefficient at or incapable of functioning optimally.

Figure 4 shows the schematic diagram of the SOMR system with integrated robot management (IRM) software. An operator determines the number of robots required for executing the specific maintenance tasks, the cooperation mode of the robots, and the authority needed for carrying out the work between human operators and robots first via the GUI in the master device connected to the IRM software. An operator then selects from four modes the behavior most suited to the task required: general teleoperation with 1:1 systems, cooperation between an operator and robots, and cooperation between robots with (or without) an operator's operational command in the GUI. Once the cooperation mode determined for each process is delivered to the remote IRM device as various control signal types, the IRM

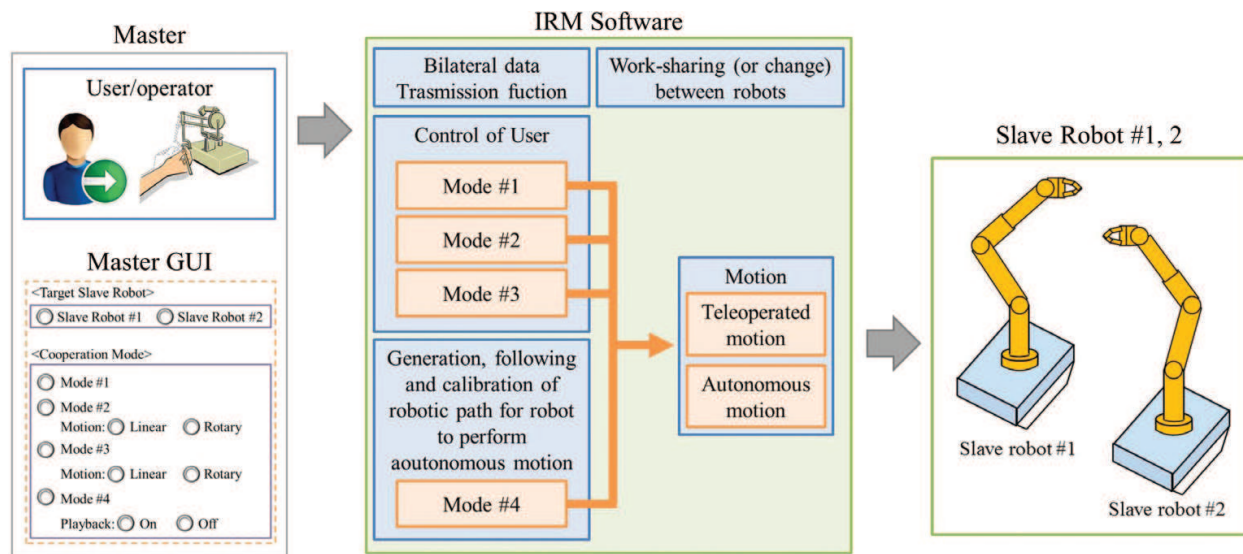


Figure 4. SOMR system with IRM (integrated robot management) software.

software performs functions, such as the switching of input/output signals between operator and robots and the generation (or calibration) of each robotic path in order to conduct cooperative motion between robots according to the cooperation mode chosen.

The IRM software is utilized in the IRM device inside the SOMR system described in **Figure 4**. This software is an integrated management software that includes role sharing (or change) between robots and the generation (calibration) of each robotic path to allow cooperation between two or more robots for tasks that are difficult for a single teleoperated robot. Initially, the IRM software classifies works in advance into four categories, primarily, by the degree of operator intervention required during the operation of the robots, based on the characteristics of the initial work (difficulty or risk level). Later on, it classifies each of the above work groups into four detailed work groups, secondarily, according to the physical motion characteristics. Thus, the various tasks that can be performed are classified into 16 types of work. Also proposed is the mode in the SOMR system that is applicable to each work group, so that the proposed SOMR system can secure universal generality and efficiency at the same time.

The proposed IRM software is mainly responsible for the bilateral data transmission function between an operator and robots, the management function of transmitted data, and the operation function of cooperation between robots for specified tasks. In more detail, the proposed IRM software includes, firstly, a function of work-sharing (or change) between robots to ensure cooperation between an operator and robots, and cooperation between robots with (or without) operator's operational commands; secondly, a function of control of operator intervention according to the nature of the tasks (difficulty or risk level) during cooperation between an operator and robots; and thirdly, a function of generation, following, and calibration of robotic paths for robots to perform autonomous motion (operator's intervention is 0%).

As shown in **Figure 5**, whether robots perform teleoperated motion or autonomous motion is determined among the remote robots according the mode (#1–#4) selected through the GUI in the master device by an operator. The mode can also be divided into straightforward or circular motion according to the physical motion characteristics of the robot that performs the

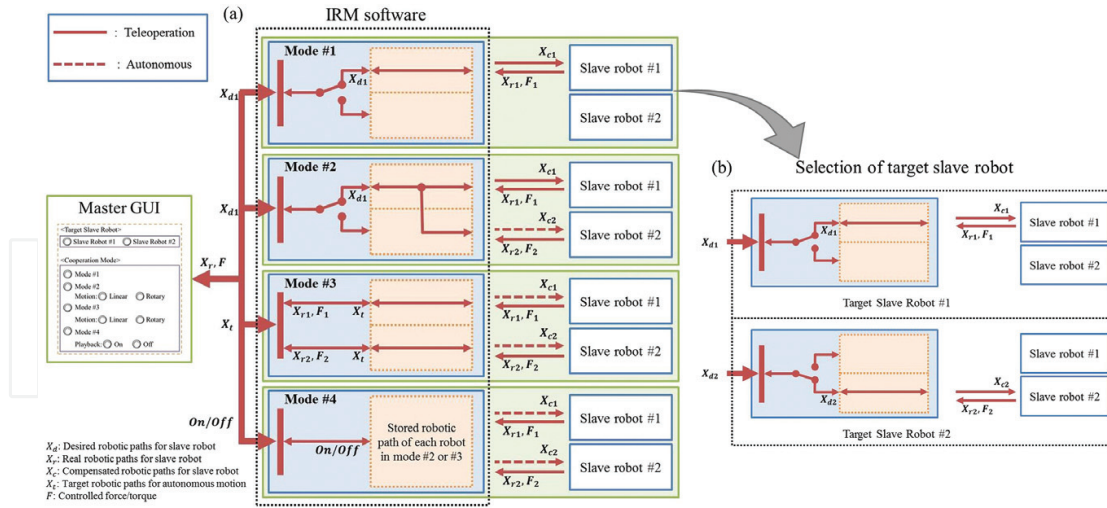


Figure 5. Structure of (a) IIR software (modes #1–#4) and (b) selection of target slave robot.

particular task. Thus, commands relating to the robotic path created via the master device by an operator are transferred to a specific robot, which performs the teleoperated motion (mode #1), while robotic paths for autonomous motion for other robots are created to assist the motion of that specific robot (mode #2). In addition, an operator can also create operational commands (from the viewpoint of motions for target objects) via the master device. These operational commands are transformed into robotic paths for autonomous motion for each robot via the IIR software (mode #3). Finally, in those instances where a task is repeatedly performed, the robotic path of each robot that corresponds to a single iteration can be stored and repeated as necessary (mode #4).

4. Verification of integrated robot management software

Figure 6 shows the simple experimental setup of SOMR system for valve handling (one of maintenance tasks) in offshore plants. The SOMR system is intended to work in places where people cannot easily access due to safety concerns and/or geographical restrictions, such as remote offshore plants, disaster relief sites, or nuclear power plants. Such simulated environments simulate the various types of maintenance work common across a variety of fields in offshore plants. In particular, the experiment aims to validate feasibility of operation of the SOMR system for the rotation of large valves, which cannot be accomplished with a single robot. The valve handling experiment was accomplished by cooperation motions between two robots through the SOMR system using the modes #2 and #3 (including linear, rotational and combination modes) of the IIR software. To evaluate the efficiency of the SOMR system and IIR software, the total time required to accomplish the valve handling task was compared with that required by two sets of the existing 1:1 teleoperated system under the same conditions, as presented in Table 1. As indicated in Table 1, the total valve handling time was reduced by about 80% of the proposed 1:N ($N = 2$) teleoperated robotic system compared to the existing 1:1 system.

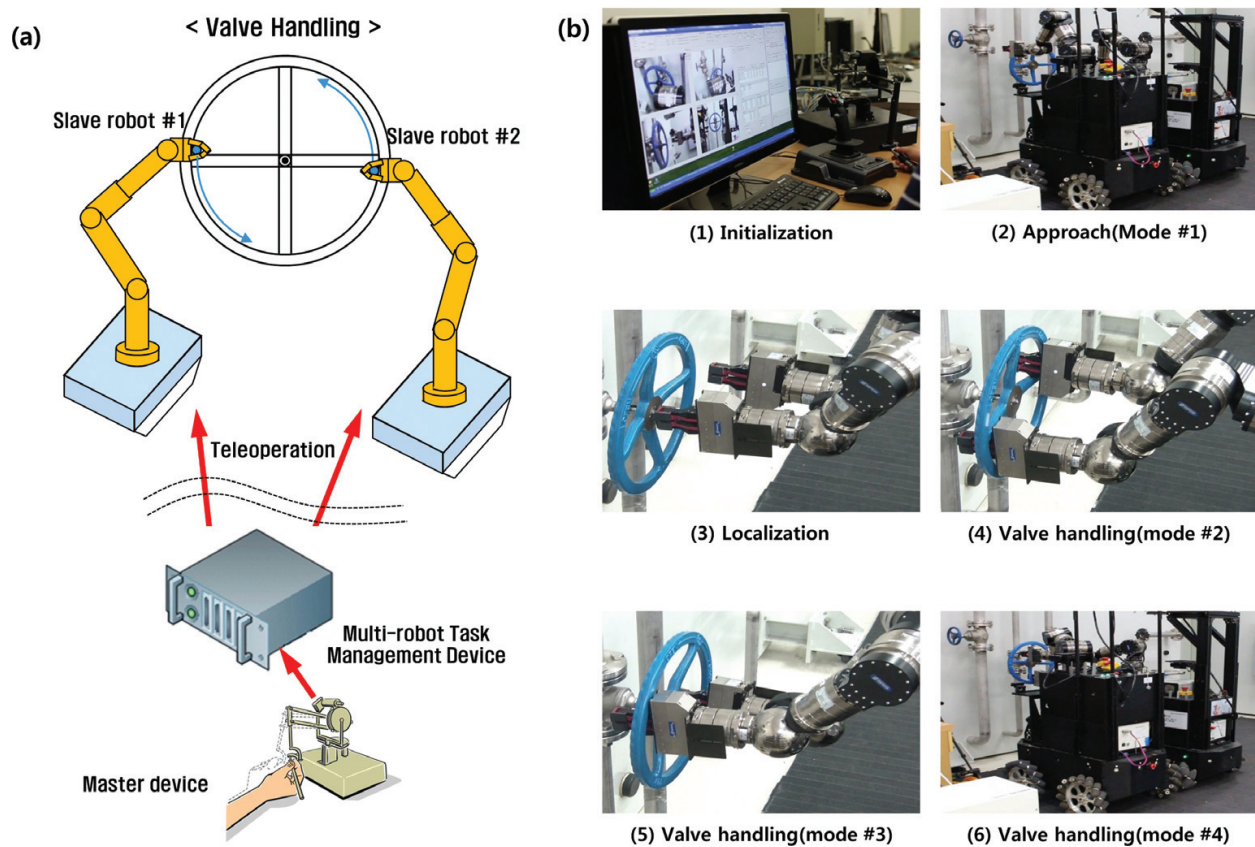


Figure 6. Experiment of a SOMR system. (a) Experimental setup and (b) experiment process.

	1:1 (s)	1:2 (s)	Decrement ratio (%)
1st	72.9	56.6	16.3
2nd	83.4	68.5	14.9
3rd	80.9	69.0	11.9
4th	99.5	70.9	28.6
5th	101.3	79.1	22.2

Table 1. Comparison of working time between 1:1 teleoperated robot system and 1:N ($N = 2$) teleoperated robot system.

5. Conclusion

This chapter is presented as a research in the area of multiagent systems. Research in the area of multiagent and multi-robot systems is concerned with the effective coordination of autonomous agents to perform tasks so as to achieve high quality overall system performance. Multiagent coordination challenges include the lack of single point of control, local views of each agent that provide only incomplete information, private goals and solution procedures of the agents, communication asynchrony, dynamic environments, and uncertainty. In the

case of robotic remote maintenance, such as the proposed 1:N teleoperated robotic system in this chapter, the research focuses on robot software architectures for multi-robot coordination, task planning for automated maintenance, detecting when multiagent motions are likely to fail, and replanning to reformulate cooperation mode to enable humans and robots to seamlessly switch control from one to another. The proposed IRM software was developed for improved maintenance work efficiency (including valve handling) in an offshore plant, but further study will be conducted to expand the application of the system to fields involving high dimensional tasks. Studies will be conducted to verify the practical application of the IRM software to a variety of industries.

Acknowledgements

This work was supported by the Industrial Strategic technology development program, 10040132, Development of a Tele-Service Engine and Tele-Robot Systems with Multi-Lateral Feedback, funded by the Ministry of Knowledge Economy (MKE, Korea) and DGIST R&D Program of the Ministry of Science, ICT, and Future Planning of Korea (17-ST-01, 17-IT-03).

Author details

Seungyeol Lee* and Jeon-Il Moon

*Address all correspondence to: syl@dgist.ac.kr

Collaboration Robotics Convergence Research Center, DGIST, South Korea

References

- [1] Sheridan TB. Teleoperation, telerobotics and telepresence: A progress report. *Journal of Control Engineering Practice*. 1995;3(2):205–214
- [2] Hokayem PF, Spong MW. Bilateral teleoperation: An historical survey. *Automatica*. 2006;42(12):2035–2057
- [3] Heikkilä T, Jarviluoma M, Juntunen T. Holonic control for manufacturing systems: Functional design of a manufacturing robot cell. *Integrated Computer-Aided Engineering*. 1997;4(3):202–218
- [4] Yamada H, Tao N, Dingxuan Z. Construction tele-robot system with virtual reality. In: *Proceedings of the Robotics, Automation and Mechatronics*; September 2008; China. USA: IEEE; 2008. pp. 21–24
- [5] Zhao D, Yamada H, Huang H, Gong W, Xia Y. 6 DOF presentation of realistic motion in operating a construction tele-robot system. In *Proceedings of the JFPS International Symposium on Fluid Power*. 2002;2002:507–512

- [6] Kwon DS, Woo KY, Song SK, Kim WS, Cho HS. Microsurgical tele-robot system. In: Proceedings of International Conference on Intelligent Robots and Systems; 17–17 October 1998; Victoria BC. USA: IEEE; 1998. 945–950
- [7] Geerinck T, Colon E, Berrabah SA, Cauwerts K, Sahli H. Tele-robot with shared autonomy: Distributed navigation development frame work. *Integrated Computer-Aided Engineering*. 2006;**12**(4):329–345
- [8] Backes PG, Mark WS. Bilateral teleoperation: An historical survey. *Automatica*. 2006; **42**(12):2035–2057
- [9] Backes PG, Tso KS. UMI: An interactive supervisory and shared control system for telerobotics. *IEEE International Conference on Robotics and Automation*; 13–18 May 1990; Cincinnati, OH. USA: IEEE; 1990. pp. 1096–1101
- [10] Douglas A, Xu H. Real-time shared control system for space telerobotics. *Intelligent Robots and Systems '93, IROS '93*. In: Proceedings of the 1993 IEEE/RSJ International Conference on; 26–30 July 1993; Yokohama, Japan. USA: IEEE; 1993. pp. 2117–2122
- [11] Hayati S, Venkataraman ST. Design and implementation of a robot control system with traded and shared control capability. In: Proceedings, 1989 International Conference on Robotics and Automation; 14–19 May 1989; Scottsdale, USA. USA: IEEE; 1989. pp. 1310–1315
- [12] Crandall JW, Goodrich MA. Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2002;**3**:1290–1295
- [13] Ali KS, Arkin RC. Multiagent tele autonomous behavioral control. *Machine Intelligence and Robotic Control*. 2000;**1**(2):3–10
- [14] Boer ER, Nakayama O, Futami T, Nakamura T. Development of a steering entropy method for evaluating driver workload. *International Congress and Exposition*. 1999;**1**:1–10
- [15] Conway L, Volz RA, Walker MW. Teleautonomous systems: Projecting and coordinating intelligent action at a distance. *IEEE Transactions on Robotics and Automation*. 1990; **6**(2):146–158
- [16] Crandall JW, Goodrich MA. Experiments in adjustable autonomy. *IEEE Systems, Man, and Cybernetics*. 2001;**3**:1624–1629
- [17] Fong T, Thorpe C, Baur C. A safeguarded teleoperation controller. *IEEE International Conference on Advanced Robotics (ICAR)*. ISBN 963-7154-05-1, Budapest, Hungary, August, 2001
- [18] Krotkov E, Simmons R, Cozman F, Koenig S. Safeguarded teleoperation for lunar rovers: from human factors to field trials. *Proc. IEEE Workshop on Planetary Rover Technology and Systems*, Minneapolis, MN, April, 1996:1–20
- [19] Sheridan TB. *Telerobotics, Automation, and Human Supervisory Control*. Cambridge: The MIT Press; 1992:180–181
- [20] Stein MR. Behavior-based control for time-delayed teleoperation [PhD Dissertation]. University of Pennsylvania; 1994

