Cooperative Control of a Single-user Multi-robot Teleoperated system for maintenance in offshore plants

Sunghoon Eom[a], Seungyeol Lee[a], Daejin Kim[a] Dongbin Shin[a] and Jeon Il Moon[a]

[a]Division of Robotic System Research, DGIST, Republic of Korea
E-mail: sheom@dgist.ac.kr, syl@dgist.ac.kr, djkim1018@dgist.ac.kr, dbs@dgist.ac.kr, jimoon@dgist.ac.kr

Abstract -

In this paper, we propose a cooperative control method of a single-user multi-robot (SUMR) teleoperated system for maintenance in offshore plants that is designed to perform in a 1:N mode (here, "1" refers to the number of user, and "N" denotes the number of slave robots), in which a single user teleoperates a number of slave robots directly to conduct a specific operation or in an autonomous cooperation mode between slave robots in order to overcome the limitation of the 1:1 teleoperation mode. This paper is also designed to extend compatibility in the SUMR teleoperated system's controller. A haptic device, which is one of the master device, is used for duplex transmission of control and status data between a robotic manipulator and user. In this paper, a new haptic library is also designed to connect between a haptic device (PHANToM premium, Sensable) and a controller based on LabVIEW. The designed new haptic library (DLL, dynamic linking library) that is created using C++ is called in LabVIEW, which is a GUI (graphical user interface) based software development tool.

Keywords -

Cooperative control; Single-user multi-robot; Teleoperation; Maintenance; Haptic; Offshore Plants

1 Introduction

In general, a teleoperated robotic system consists of a master device that collects task commands from users and a slave robot that follows the collected task commands run in a 1:1 teleoperation mode. Such a 1:1 teleoperation mode becomes limited in real sites if the target work piece exceeds the payload of the slave robot, a required workspace exceeds a the slave robot's workspace, or the target work piece is blocked by obstacles such that a user cannot perform teleoperation via visual information [1]. Furthermore, in this mode, if a user performs simple repetitive tasks continuously, the user can easily feel fatigue because of the repeated task commands generated using the master device, thereby negatively affecting the task quality.

This study proposes cooperative control method of a SUMR teleoperated system for maintenance in offshore plants that is designed to perform in a 1:N mode (here, '1' and 'N' refer to the number of a user and slave robots), in which a user teleoperates a number of slave robots directly to conduct a specific task, or in an autonomous cooperation mode between slave robots in order to overcome the limitations of the aforementioned 1:1 teleoperation mode [2], [3]. The proposed control method is responsible for the role sharing and integrated management of slave robots to divide the operation mode of the slave robots into various types according to the user's intervention level and the characteristics of the target task beforehand and to perform the target task using the robot operation mode selected by the user. The control method also includes technologies for the following: (1) a role change between slave robots that are controlled remotely and slave robots that perform the autonomous operation, (2) path generation for slave robots that perform the autonomous operation, and (3) path compensation when slave robots contact the obstacles while performing the task under various environments where one of the slave robots simply follows the task command given by a user via the software and the rest of the slave robots follow the operation of a slave robot that is remotely controlled, or where all slave robots perform an autonomous operation, to perform the target task set by the user [4]. Further, when the target task is a simple repetitive task that is performed continuously, the software includes a remote storing and playback technology using which the slave robots edit and store the task command created via remote storing from a remote user, thereby following the stored task commands repeatedly without any task intervention for the subsequent repetitive tasks performed to achieve the target task.

The cooperative control method proposed in this paper is also designed to extend compatibility in the SUMR teleoperated system's controller. A haptic device, which is one of the master devices, is used for duplex transmission of control and status data between a robotic manipulator and user. In this paper, a haptic device (PHANToM premium, Sensable) and a haptic device's controller based on LabVIEW are connected using proposed haptic library. The proposed haptic library (DLL, dynamic linking library) that is created using C++ is called in LabVIEW, which is a GUI (graphical user interface) based software development tool.

2 Cooperative Control of a Single-User Multi-Robot (SUMR) Teleoperation System

The SUMR teleoperation system allows a user to remotely control multi-robots and aims to perform tasks that cannot be done with a single robot (e.g. bulky or heavy objects handling) through the cooperation between slave robots [3], [5]. 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 number of cameras [6]. The development goal of the SUMR 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 performing specific tasks [7], [8].

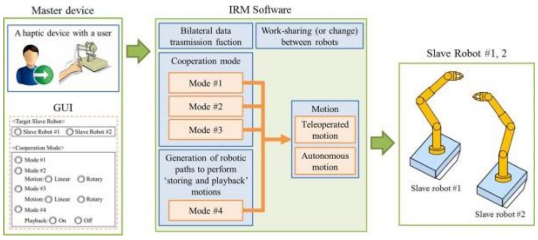

Figure 1. Single-User Multi-Robot (SUMR) teleoperated system with Integrated Robot management (IRM) software

Figure 1 shows the schematic diagram of the SUMR teleoperation system proposed in this paper, in which the number of users and robots can be changed according to the characteristics of the specific tasks. As shown in Figure 1, a user determines the number of users and robots required for performing the specific tasks, the cooperation mode of the robots, and the authority needed for carrying out the task between users and robots first via the GUI in the master device connected to the IRM software. A user then selects from four modes the behaviour most suited to the task required: general teleoperation with 1:1 systems, cooperation between a user and robots, and cooperation between robots with (or without) a user's task commands 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 software performs functions such as the switching of input/output signals between user 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.

3 Integrated Robot Management (IRM) Software for SUMR Teleoperation System

The proposed IRM software as shown in Figure 1 is 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 slave robots for tasks that are difficult for a single teleoperated robot [9].

3.1 IRM Software

The IRM software classifies maintenance tasks in offshore plants into four categories primarily by the degree of user intervention required during the operation of the robots, based on the characteristics of the maintenance task (difficulty or risk level). Later on, it then classifies each of the above categories into four detailed task groups secondarily according to the physical motion characteristics of each maintenance task. Thus, the various tasks that can be performed are classified into 16 types of task. Also proposed is the mode in the SUMR teleoperation system that is applicable to each task group, so that the proposed SUMR teleoperation 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 a user and robots, the management function of transmitted data, and the operational function for cooperation between robots for specified tasks [10]. In more detail, the proposed IRM software includes, firstly, a function of work-sharing (or change) between robots to ensure cooperation between a user and robots, and cooperation between robots with (or without) user's task commands; secondly, a function of control of user intervention according to the characteristics of the tasks (difficulty or risk level) during cooperation between a user and robots; and thirdly, a function of generation, following, and calibration of robotic paths for performing autonomous motion (user's intervention is 0%). Figure 1 describes the main functions of the proposed IRM software.

Your manuscript must be prepared for A4-size (210 x 297 mm) paper. Use the margin settings specified in Table 1 and do not number the pages of the paper.

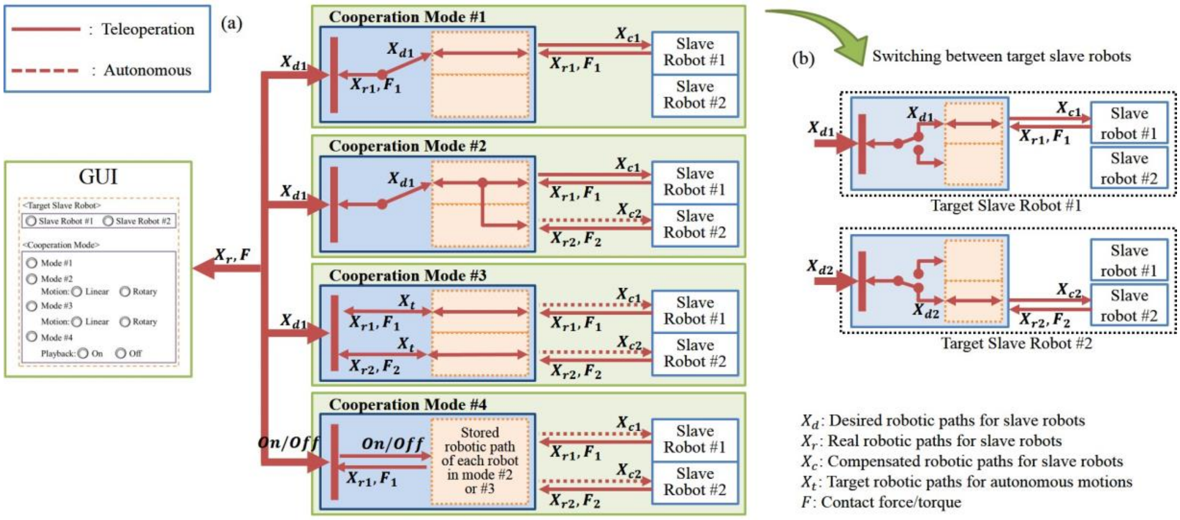

Figure 2. Structure of (a) IRM software (Cooperation Mode #1~#4), and (b) Switching between target slave robots

3.2 Structure of IRM Software

The IRM software structure proposed in this paper is shown in Figure 2. As shown in the figure, whether slave robots perform teleoperated motion or autonomous motion is determined according the cooperation mode (#1 ~ 4) selected through the GUI in the master device by a user. The cooperation mode can also be divided into linear or rotary motion according to the physical motion characteristics of the slave robot that performs the specific task. Thus, commands relating to the robotic path created via the master device by a user are transferred to a specific slave robot, which performs the teleoperated motion (Mode #1), while robotic paths for autonomous motion for other slave robots are created to assist the motion of that specific slave robot (Mode #2). In addition, a user can also create task commands (from the viewpoint of motions for target work piece) via the master device. These task commands are transformed into robotic paths for autonomous motion for each slave robot via the IRM software (Mode #3). Finally, in those instances where a task is repeatedly performed, the robotic path of each slave robot that corresponds to a single iteration can be stored and repeated as necessary (Mode #4). In Figure 2, ‘Mode #1’ is same structure as general 1:1 teleoperation systems.

The IRM software includes a function that allows a user to switch a robot for control in the style of the 1:1 teleoperation system, even within the SUMR teleoperation system. A user can easily switch the target slave robot in the GUI of the master device if he/she wants to control robot #2 immediately after teleoperation robot #1 during ‘Mode #1’. Figure 2 (b) shows the switching structure between teleoperation target slave robots in the IRM software.

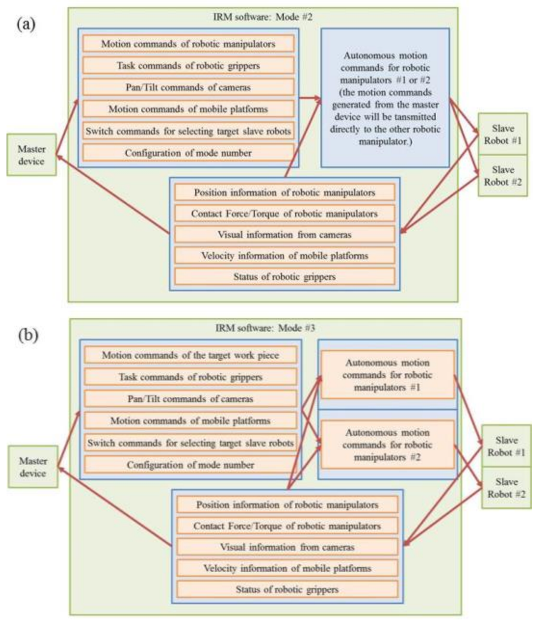

Figure 3. Structure of (a) Mode #2, and (b) Mode #3 in IRM software

The ‘Mode #2’, as shown in Figure 3 (a), is a method of allowing cooperation between robots by having the secondary robots assist the primary teleoperated robot’s motion. Once a user creates a robotic path for the target slave robot teleoperated motion through the mater device, the IRM software transmits the path to that robot while it also creates (including calibration) robotic paths for other slave robots autonomous motion based on the motion information of the teleoperated robot. And 'Mode #3' is a method in which all robots perform autonomous motions based on commands for cooperative operation between robots to perform particular tasks. Once a user creates commands for cooperative operation between robots (motion commands about the target work piece) through the master device, the IRM software creates (including calibration) robotic paths for autonomous motion for each robot (Figure 3 (b)). Furthermore, 'Mode #4' is a method that is used when tasks are repeated iteratively. The IRM software stores the path of each slave robot in a specific subtask and has each slave robot perform its tasks by following the path repeatedly as needed. In particular, 'mode #4' has a structure that can store and playback the robotic paths stored in modes #1~3 selectively as needed.

4 Library for Compatibility between Haptic Device and Controllers

In a teleoperated robot system, a master device is necessary to control a robotic manipulator as much as the slave robot system [11]. A haptic device, which is one of the master devices, is used for duplex transmission of control and status data between a robotic manipulator and user [12]. Recently, haptic devices are being used in various applications, such as industrial robots, medical or surgery robots, assistance robots, and laboratory based robotic platforms. Systems implementing haptic devices effectively replicate wrist motion and have been proven to be effective for manipulation of teleoperated robotic systems. Although haptic devices have been widely used in teleoperated robot applications, its functionality is unsatisfactory for teleoperated applications that require compatibility. This is because the software subsystems of these haptic devices have been created using packaged libraries [12], [13]. To assign compatibility, new haptic library is proposed to interface between a haptic device and a SUMR teleoperated system's controller.

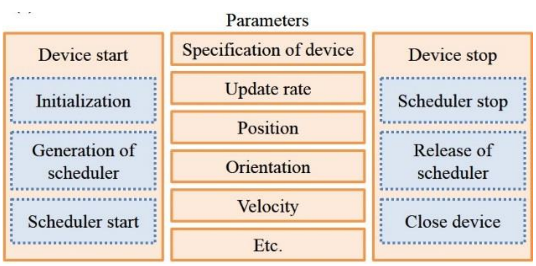

Figure 4. Dynamic Linking Library (DLL) structure for a new haptic library

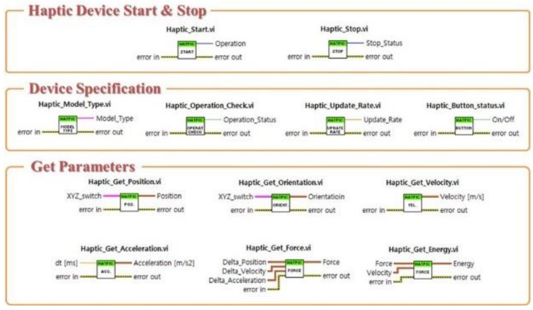

Figure 5. SubVI (Virtual Instruments) with new haptic library

According to this study, a DLL (dynamic linking library) was generated in C++ for linking the haptic device with the embedded controller in the LabVIEW environment. Figure 4 shows the configuration of the generated DLL used for the haptic device. The generated DLL consists of a 'Haptic Device Start' feature for connecting with the haptic device, 'Parameters' of the haptic device, and a 'Haptic Device End' feature for releasing the haptic device. Here, the functions responsible for the start and termination of "Haptic Device Start" and "Haptic Device End" include functions used for initiating the haptic device and managing memory and threads. 'Parameters' include device information, location, direction, linear velocity, and angular velocity of the haptic device.

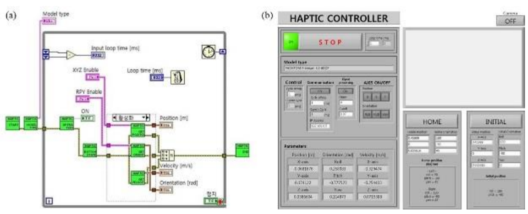

Figure 6. VI (Virtual Instrument) with haptic library to operate a haptic device for SUMR teleoperated system: (a) block diagram, (b) front panel

The DLL of a haptic device is generated by extracting based on each function (Solid rectangle of Figure 4). The DLL of the haptic device, written in C++, uses a library call function provided in LabVIEW, which is generated by the lower layer VI (SubVI) based on each function, as shown in Figure 5. The block diagram and the front panel, shown in Figure 6 (a), (b), respectively, for driving the embedded controller-based haptic device using LabVIEW were designed using the subVI.

5 Conclusion

In this paper, a SUMR teleoperated system with a haptic device using a new haptic library and cooperative control method, which adopts a strategy of cooperation between a user and multi-robots, were proposed to improve work efficiency. Since the proposed control method was initially developed with the aim of improving the interaction technology linking a user with robots, future research will continue to refine the system for use in a variety of applications, to industries and fields where high-dimensional and sophisticated task is required by combining systematically an experienced worker's skill and intelligence with a robot's accuracy and physical power. More specifically, both human-robot interface technologies that can perceive the various and complex intentions of workers accurately, and robot-based safety technologies that can prevent accidents occurring due to user's error will be studied in the future.

References

[1] Sheridan T. Teleoperation, telerobotics and telepresence: A progress report. *Control Engineering Practice*, 3(2):205-214, 1995.

[2] Peter F. Mark W. Bilateral teleoperation: An historical survey. *Automatica*, 42(12):2035-2057, 2006.

[3] Uchiyama M. Dauchez P. A symmetric hybrid position/force control scheme for the coordination of two robots. In *1988 IEEE International Conference on Robotics and Automation*, pages 350-356, Philadelphia, America, 1988.

[4] Kosuge K. Ishikawa J. Furuta K. Sakai M. Control of single-master multi-slave manipulator system using VIM. In 1990 IEEE International Conference on Robotics and Automation, pages 1172-1177, Cincinnati, America, 1990.

[5] Yokokohji Y. Yoshikawa T. Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment. *Robotics and Automation*, 10(5):605-620, 1994.

[6] Hwang G. Szemes P. Ando N. Hashimoto H. Development of a single-master multi-slave tele-micromanipulation system. *Advanced Robotics*, 21(3-4):329-349, 2007.

[7] Sirouspour S. Setoodeh P. Multi-operator/Multi-robot Teleoperation: An Adaptive Nonlinear Control Approach. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1576-1581, Alberta, Canada, 2005.

[8] Imad H. Amit G. Ning X. Chow M. Yun H. Toshio F. Design and Analysis of Internet-Based Tele-Coordinated Multi-Robot Systems. *Autonomous Robots*, 15(3):237-254, 2003.

[9] Furuta K. Kosuge K. Shiote Y. Hatano H. Master-slave manipulator based on virtual internal model following control concept. In *1987 IEEE International Conference on Robotics and Automation*, pages 567-572, Raleigh, America, 1987.

[10] Keyvan H. Septimiu E. Analysis of Control Architectures for Teleoperation Systems with Impedance/Admittance Master and Slave Manipulators. *The International Journal of Robotics Research*, 20(6):419-445, 2001.

[11] Lee S. Lee K. Lee S. Kim J. Han C. Human-robot cooperation control for installing heavy construction materials. *Autonomous Robots*, 22(3):305-319, 2007.

[12] Gregor Š. Simon Z. Jure Č. Riko Š. Virtual User Interface for the Remote Control of a Nano-Robotic Cell Using a Haptic-Device. *Mechanical Engineering*, 56(7-8):423-435, 2010

[13] Stolle C. Fatikow S. Control System of an Automated Nanohandling Robot Cell. In *ISIC 2007 IEEE 22nd International Symposium on Intelligent Control*, pages 664-669, Suntec, Singapore, 2007.