2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Task Sequencer Integrated into a Teleoperation Interface for Biped Humanoid Robots

Shin'ichiro Nakaoka, Mitsuharu Morisawa, Rafael Cisneros, Takeshi Sakaguchi,
Shuuji Kajita, Kenji Kaneko and Fumio Kanehiro

***Abstract*— This study describes our teleoperation interface, through which an operator can dictate various tasks to a biped humanoid robot. In this interface, the robot model and the environmental data are presented in three-dimensional computer graphics, and the robot actions are specified by several operational markers. The interface also provides a *task sequencer* function, enabling the operator to perform a task merely by proceeding a *task sequence*, which structures the procedure of a particular task. The task sequencer provides a much more efficient operation than the direct use of operational markers. We verified the interface by participation in the DARPA Robotics Challenge (DRC) Finals. Our humanoid robot HRP-2kai operated through our teleoperation interface completed most of the DRC tasks.**

I. Introduction

Robots are expected to be deployed as disaster response agents at hazardous disaster sites. Because many disaster environments are designed for humans, humanoid robots are potentially more suitable than other robot types for performing various tasks in such environments.

Based on this philosophy, we have developed a humanoid robot named HRP-2Kai [1], which is an improved version of HRP-2, for undertaking disaster response tasks. Alongside, we have developed the technology by which human operators dictate tasks to the robot through teleoperation. To verify our developed technology, we participated in the DARPA Robotics Challenge (DRC) Finals [2], held in California in June 2015. This study describes the teleoperation interface developed as part of the above project, and its verification results at the DRC Finals.

II. Overview of Our Teleoperation Interface

In many teleoperation interfaces for humanoid robots, the view from the robot's head is projected toward the operator's view and the physical movement of the operator is reflected to the robot [3][4]. Although this design allows intuitive operation based on the operator's own body, the robot is manually operated through the interface. Therefore, an autonomous robot guided by this interface would be difficult to realize. In addition, the interface places much physical burden on the operator, and requires special input devices and a high-bandwidth, low-latency communication path between the interface and the robot. Such configurations are not necessarily available at disaster sites.

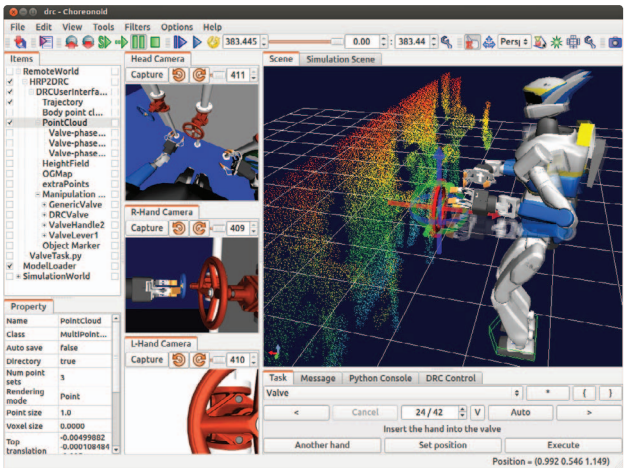


Fig. 1. A screenshot of our teleoperation interface

Our goal is to design an optimal teleoperation interface for a disaster response robot by dissociating the interface from the operator's body. In particular, the following functionalities should be achieved:

- Full exploitation of the robot's autonomy.
- Minimal physical burden on the operator.
- Functionality in a poor communication environment.
- Low-cost availability.

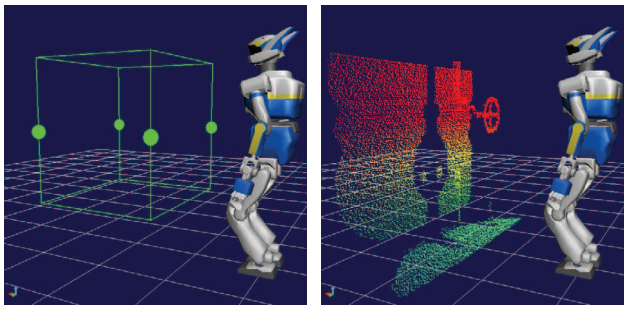
Following the above policy, we constructed a teleoperation interface on Choreonoid [5][6], an integrated graphical user interface (GUI) framework for robots. Figure 1 is a screenshot of the interface. The additional functional elements required for the teleoperation are implemented as a *DRCUserInterface* plugin. By combining those functions with the existing functions of Choreonoid and various RT-components [1], including motion planners and robot controllers, we achieved a system that integrates all of the functions required for the teleoperation. The system works on standard PCs and requires no special input devices. The system is also integrated with the dynamics simulation function of Choreonoid, and the interface can be easily tested through simulations.

This work is partially supported by the NEDO project "The international R&D and demonstration project in environment and medical device sector / The international R&D and demonstration project on robotic field / R&D project on disaster response robots."

The authors are with National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, Japan, s.nakaoka at aist.go.jp

[1]RT-components are the software components that conform to the specifications of the RT-Middleware [7]

978-1-4799-6885-5/15/$31.00 ©2015 IEEE



(a) (b)

Fig. 2. (a) Measurement region marker and (b) point cloud data

III. Basic Functions of Our Interface

In the developed teleoperation interface, the current real-world situation obtained from the robot sensors is presented on the interface using three-dimensional (3D) computer graphics (CG). The operator then instructs the robot on its next action through various markers presented on the interface. These processes are iterated in the basic teleoperation procedure. This section briefly introduces the real-world information and the operational markers presented on the interface. For details, the reader is referred to [8].

A. Status of the Robot

The current position and posture of the robot are presented as a 3DCG model on the interface. The model status is continuously updated as the robot moves.

B. Camera Images

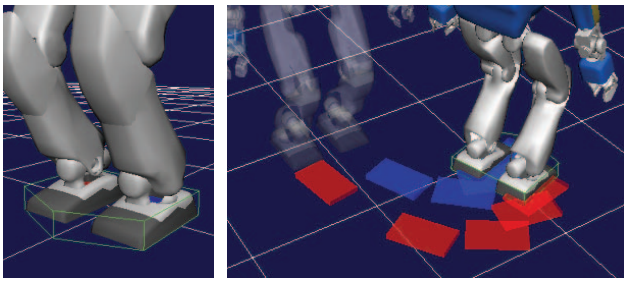
The images obtained from the cameras mounted on the robot are presented on the interface. The number of presented camera images is arbitrary and depends on the robot specifications. HRP-2kai captures and presents four images from the cameras mounted on its head, back, and right and left hands.

C. Point Cloud

3D point cloud data representing the environments around the robot can also be presented on the interface. The environment in front of HRP-2kai is measured as a 3D point cloud by a laser range finder mounted on the robot's head, while synchronously controlling the head pitch joint. In the interface, the measurement region can be specified using the *measurement region marker* shown in Figure 2(a). The robot then begins the measurement and sends the result to the interface. Finally, the measured point cloud data are presented as shown in Figure 2(b).

D. Walk Destination Marker

The robot moves by walking. To implement walking, the destination must be first specified using the *walk destination marker* shown in Figure 3(a). When the marker position is updated, the foot-step path to the destination is planned and presented as shown in Figure 3(b). If no problem is identified in the planned path, the operator fixes the destination and the robot starts to walk.



(a) (b)

Fig. 3. (a) The walk destination marker and (b) a foot-step plan

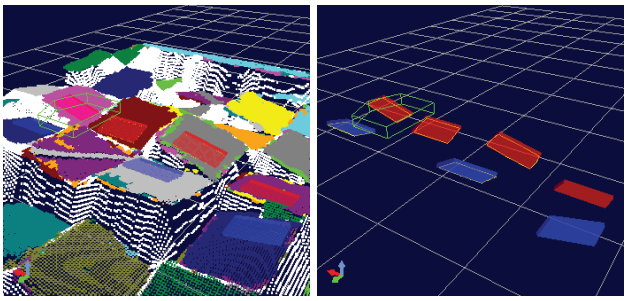


Fig. 4. A foot-step path planned for walking on uneven terrain

Note that the system recognizes the terrain around the feet as height-field data estimated from the measured point cloud, and the foot-steps are adjusted to the terrain. Figure 4 shows an example of a walk plan on uneven terrain. This function and our walking controller enable the robot to walk on uneven terrain [9].

E. Hand Markers

The reaching destination of each hand is specified by a *hand marker*, as shown in Figure 5. When the position of the hand marker is updated, a reaching path to the position is searched, and the path is presented if the solution is found. The path is planned to avoid the collisions between the hand and the environments represented by the point cloud data [10]. If no problem is identified in the planned path, the operator fixes the reaching plan and the robot initiates the reaching motion.

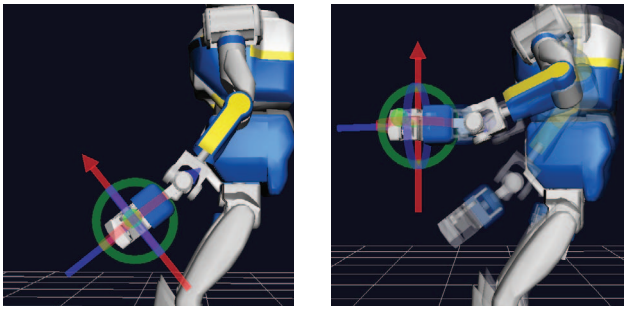


Fig. 5. A hand marker

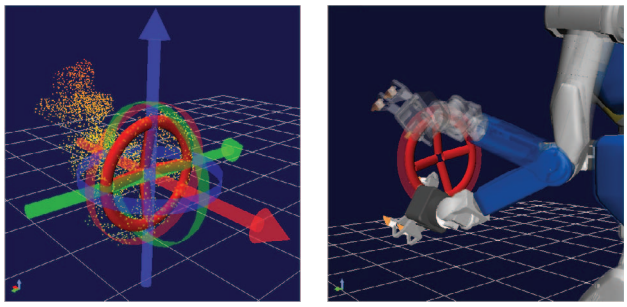


Fig. 6. The manipulation marker

F. Manipulation Marker

Object manipulation is specified by the *manipulation marker* shown in Figure 6. This marker has the following three functions:

1) Visualizing the object to be manipulated.
2) Recognizing the object position.
3) Specifying the goal position of the manipulated object.

Function 1 visualizes the estimated shape and position of the target object as a corresponding 3DCG model. The shape is specified by choosing one of the registered shape models. For some types of objects, a *variable shape model* is available; the shape of the model can be varied by setting parameters that best fit the shape to the actual object. For example, in the variable valve model, the diameter, handle width, and the number of spokes of the valve can be dynamically adjusted to match the actual valve.

Function 2 recognizes the current position of an object by matching the model to the measured point cloud. This function is necessary for manipulating the object. By combining the initial position alignment manually performed by the operator and the automatic alignment algorithm such as the iterative closest point (ICP), the object position is efficiently recognized by the system.

Function 3 specifies the object manipulation. The operator first specifies whether the manipulation is to be performed by the right hand, the left hand, or both hands. The manipulation goal, defined as the object position after the manipulation, is then specified by moving the manipulation marker. When the operator fixes the goal position, the hand trajectories that achieve the manipulation goal are planned. Based on these trajectories, the robot moves its hands to perform the manipulation.

Note that the *object marker*, which lacks Function 3, is also available. This marker is useful for manipulating a pair of objects. For example, when an object must be attached to or detached from another object, the object marker is implemented on the fixed counterpart object.

IV. Task Sequencer

Functions similar to the basic functions described in Section III are becoming standard low-level functions in the teleoperation system developed for the DRC [11][12][13]. Combining these functions achieves various operations, but

Fig. 7. A screenshot of the task sequencer interface

applying these basic functions to perform actual tasks remains a challenging problem. Each function can be achieved by manually operating the corresponding marker in the 3DCG interface. However, such reliance on manual operations would place great burden on the operator for the following reasons:

- Manually setting the markers to appropriate positions requires the operator's skill and concentration.
- Since the basic functions correspond to low-level, small granularity actions, a task may require many operations.

These burdens increase both the influence of the operator's skill and mental state on the task performance and the time taken to complete the task. Consequently, the ability to complete the task quickly and reliably is heavily compromised.

To resolve this problem, if the task is roughly known in advance, the number of manual operations can be reduced by preparing a script describing the task procedure. The task is then performed by following this procedure. For this purpose, we integrated a *task sequencer* system into our teleoperation interface. The task sequencer system is outlined below:

- A *command* defines minimum operation unit of a task, and a *phase* is an action unit comprising one or more commands.
- A *task sequence* defines the procedure of performing the whole task by executing a number of phases.
- An arbitrary number of task sequences can be registered, and the operator can switch to the task sequence relevant to the current task.
- Task sequences are mainly written as Python scripts. The scripts can be updated on the fly during the teleoperation.
- The commands contained in the current phase are presented as corresponding buttons on the *task sequencer* interface, which the operator clicks to proceed through the task sequence.
- A command can execute basic functions, transit to another command or phase, and perform various other operations.
- Transitions of commands or phases enable conditional branching, loops, and automation in the task sequence.
- Manual operations of basic functions can be combined to assist command executions or to recover from unexpected situations.

Figure 7 is a screenshot of the task sequencer interface. In this figure, the selected sequence is the pre-prepared "valve task" sequence, which governs the task of rotating a valve.

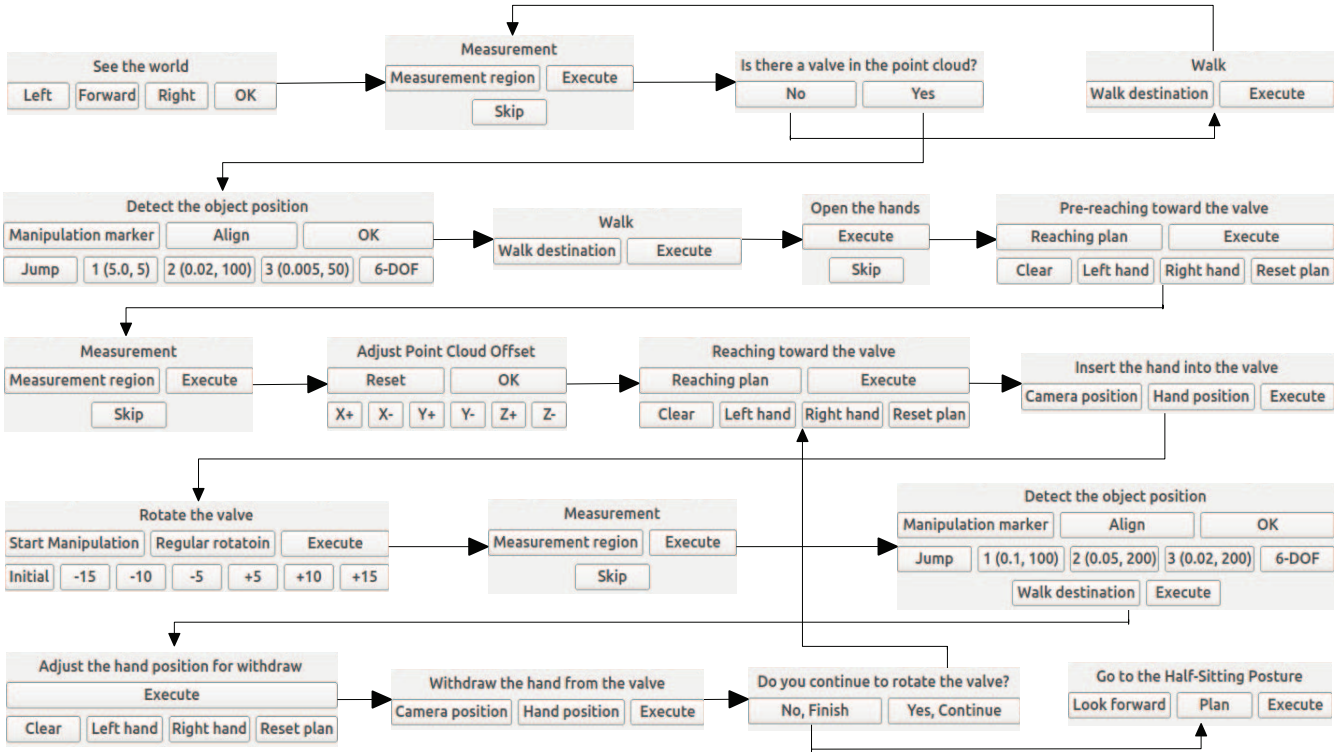


Fig. 8. A task sequence for the valve task. Each block encloses a task phase and the arrows show the transitions between phases. Note that this is a simplified version of the original valve task sequence, which consists of 40 phases including fully automated ones.

Under the selected phase, the robot walks to the position where it can operate the target valve. This phase comprises two commands: setting the walk destination and executing the walk. By pushing the corresponding buttons, the standing position is adjusted, and the phase is incremented. The above operation can be automated. In fact, the first command is automatically applied by default when this phase becomes the current phase. Furthermore, if the automatic mode is enabled by the "Auto" button, the second command is automatically applied as well.

The task sequencer system releases the operator from the detailed operations of the markers. The operator needs only to focus on the task progression and the current statuses of the robot and its environments. Figure 8 outlines the whole sequence of the valve task, with the phase transitions indicated by arrows. In each phase, the transition to the next default phase is assigned to the top-right button, which the operator presses to proceed the task.

Although some of the phases in a task sequence cannot be automated, this does not present a large problem because commands for inputting the operator's decisions can be added to each phase. Such commands are usually placed below the first command row in the task sequencer interface, as shown in Figure 8. Dependence on these commands constitutes a type of manual operation, but only within the context of the task sequence. Therefore, the operator burden on the operator is much less than when all basic functions are manually operated.

The task sequencer is designed so that automatic and interactive operations can be mixed as necessary. When the operator encounters unexpected circumstances during the task operation, the task sequence may be continued by temporarily applying manual operations instead of proceeding the task sequence. This flexibility is necessary in a practical teleoperation system.

V. Communication System

In the DRC finals, communication between the robot and the teleoperation terminal, called the Operator Control Station (OCS), is limited to two paths:

1) Bidirectional, 9600 bps communication path with continuous connection
2) Unidirectional, up to 300 Mbps communication path without continuous connection

Although the connection by path 1 is stable, its speed is only 9600 bps. Communication path 2 is sufficiently fast to send large data, but is enabled only from the robot side to the OCS side, and does not ensure continuous connection. Communication breakdown (or blackout) occurs at a certain frequency when the robot is assumed to be indoors, and lasts for up to 30 sec. Five out of the eight assigned tasks are affected by blackout episodes. Note that field computers (FCs), which work on additional processing, can be placed at the robot side of the communication path,

To maximize the ability to perform tasks under the above conditions, we designed the following communication system:

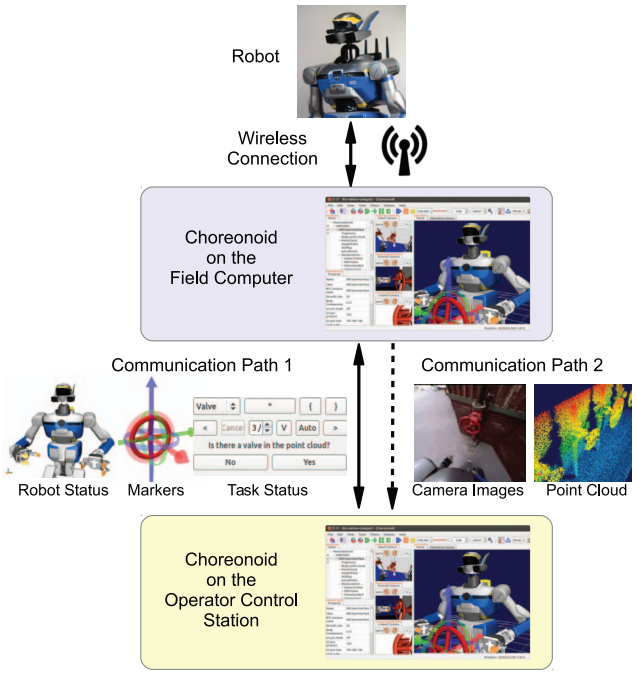


Fig. 9. Overview of the communication system

- Choreonoid with the teleoperation interface runs in both FC and OCS.
- At the FC side, Choreonoid runs in normal mode and handles all of the processing, including internal processing.
- At the OCS side, Choreonoid runs in a special mode that handles only the front-end of the interface. Its behavior is synchronized with that of the FC-side Choreonoid.
- The statuses of the robot model, operational markers, and the task sequencer are synchronized through communication path 1.
- The information on the camera images and point clouds are synchronized through communication path 2.

Figure 9 overviews the communication system. In this design, both ends of the communication are not different software modules, but are handled by the same applications loaded with the same data. This structure improves the simplicity and robustness of the communication system because only the synchronization of the front-end elements is required in the communication and the correctness of the implementation can be visually verified.

Synchronizing the statuses of the robot model, markers, and task sequences requires little traffic volume, so the communication limits negligibly affect the operation. The task sequencer is especially important for maintaining fine operability under limited communication. Since both Choreonoid processes are loaded with the same task sequence data, synchronizing the task sequencer status requires only the index values of the current phase and command. Therefore, the operator can handle complex operations with low communication traffic.

The data of the camera images and point clouds are relayed to OCS via communication path 2. Consequently, the blackout interferes with the data visualization. To minimize the impact of the blackout, we shuffled the order of sending data packets so that the operator can roughly comprehend the environmental situation even when many data packets are lost during the blackout. This is useful for proceeding a task sequence because the precise object position is finally detected by the FC-side Choreonoid, which always has the entire environmental data, and the role of the OCS-side interface is to give a hint of the object position. Moreover, the lost data packets are resent again immediately after the blackout. Adopting these implementation techniques, we managed to present the environmental data on the OCS-side interface.

VI. Verification at the DRC Finals

Employing our teleoperation system, we developed the sequences for the tasks allocated in the DRC Finals, namely, the door, valve, wall, plug[2], button, lever, terrain, and stairs [1][14].

The other task sequences were prepared similarly to the valve task sequence. In test-field runs of the developed task sequences, we confirmed that HRP-2kai could perform all of these tasks through our teleoperation interface.

We then participated in the DRC finals, achieving five points out of eight. The points were scored from the driving [3], door, valve, plug, and terrain tasks. Figures 10 and 11 photograph HRP-2kai performing the valve and terrain tasks, respectively. Unfortunately, the robot toppled at the final step of the terrain task. Although the robot had scored its point for this task before falling over, it sustained critical damage after crashing to the ground, and could not proceed to the next task, which was the stairs task. Consequently, we achieved the tenth position among 23 teams participating in the competition.

The above verification proved that our teleoperation interface sufficiently directs various tasks. In particular, the reliability of the teleoperation depends on the task sequencer. The valve task was successfully performed by the automatic recognition of the valve position using the manipulation marker with the variable valve model. In this task, most of the operations were automated by the task sequencer. Despite the frequent communication blackouts during the valve task, our robot completed the task within the expected time.

On the other hand, the blackout lengthened completion time of the wall and plug tasks by our robot. This resulted from the inability to automatically recognize the precise positions of the target objects in those tasks. To assist the object recognition, the operator was required to adjust the object position in some phases, while viewing the camera images or point clouds presented on the interface. During blackout periods, the operator had to wait for the camera images or the

[2]The plug, button and lever tasks are the candidates of the surprise task, which is decided at the start of each competition day

[3]The system implementing the driving task was mainly developed by the AIST-CNRS Joint Robotics Laboratory

Fig. 10. The valve task performed by HRP-2kai at the DRC Finals [15]

Fig. 11. The terrain task performed by HRP-2kai at the DRC Finals [16]

point clouds to be sent, which extended the operation time. This delay was responsible, at least partly, for our fewer points than the topmost teams. By improving the automatic object recognition function, we could improve the autonomy of the task performance and therefore the performance in poor communication environments. This important issue will be the focus of our future work.

REFERENCES

[1] K. Kaneko, M. Morisawa, S. Kajita, S. Nakaoka, T. Sakaguchi, R. Cisneros, and F. Kanehiro, "Humanoid robot HRP-2Kai - improvement of HRP-2 towards disaster response tasks -," in *Proceedings of the 15th IEEE-RAS International Conference on Humanoid Robots*, Seoul, Korea, November 2015.

[2] "DARPA Robotics Challenge," 2015, http://www.theroboticschallenge.org/.

[3] H. Hasunuma and K. Nakashima, "The tele-operation of the humanoid robot - workspace extension of the arm with step motion," in *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan, December 2005, pp. 245–252.

[4] C. L. Fernando, M. Furukawa, T. Kurogi, S. Kamuro, K. Sato, K. Minamizawa, and S. Tachi, "Design of telesar v for transferring bodily consciousness in telexistence," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, October 2012, pp. 5112–5118.

[5] "Choreonoid Official Site," http://choreonoid.org/.

[6] S. Nakaoka, "Choreonoid: Extensible virtual robot environment built on an integrated GUI framework," in *Proceedings of the 2012 IEEE/SICE International Symposium on System Integration (SII2012)*, Fukuoka, Japan, December 2012, pp. 79–85.

[7] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "RT-Middleware: Distributed component middleware for RT (Robot Technology)," in *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, August 2005, pp. 3555–3560.

[8] S. Nakaoka, M. Morisawa, K. Kaneko, S. Kajita, and F. Kanehiro, "Development of an indirect-type teleoperation interface for biped humanoid robots," in *Proceedings of the 2014 IEEE/SICE International Symposium on System Integration (SII2014)*, Tokyo, Japan, December 2014, pp. 590–596.

[9] M. Morisawa, N. Kita, S. Nakaoka, K. Kaneko, S. Kajita, and F. Kanehiro, "Biped locomotion control for uneven terrain with narrow support region," in *Proceedings of the 2014 IEEE/SICE International Symposium on System Integration (SII2014)*, Tokyo, Japan, December 2014, pp. 34–39.

[10] F. Kanehiro, E. Yoshida, and K. Yokoi, "Efficient reaching motion planning and execution for exploration by humanoid robots," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, October 2012, pp. 1911–1916.

[11] C. Phillips-Grafflin, N. Alunni, H. B. Suay, J. Mainprice, D. Lofaro, D. Berenson, S. Chernova, R. W. Lindeman, S. Chernova, R. W. Lindeman, and P. Oh, "Toward a user-guided manipulation framework for high-dof robots with limited communication," *Intelligent Service Robotics*, vol. 7, no. 3, pp. 121–131, 2014.

[12] A. Romay, S. Kohlbrecher, D. C. Conner, A. Stumpf, and O. von Stryk, "Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots," in *Proceedings of the 14th IEEE-RAS International Conference on Humanoid Robots*, Madrid, Spain, November 2014, pp. 979–986.

[13] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. DArpino, R. Deits, M. DiCicco, D. Fourie, T. Koolen, P. Marion, M. Posa, A. Valenzuela, K.-T. Yu, J. Shah, K. Iagnemma, R. Tedrake, and S. Teller, "An architecture for online affordance-based perception and whole-body planning," *MIT CSAIL Technical Report 2014-003*, 2014.

[14] R. Cisneros, T. Sakaguchi, S. Kajita, S. Nakaoka, M. Morisawa, K. Kaneko, and F. Kanehiro, "Task-level teleoperation manipulation for the HRP-2Kai humanoid robot," in *Proceedings of the 15th IEEE-RAS International Conference on Humanoid Robots*, Seoul, Korea, November 2015.

[15] "DARPA Robotics Challenge, Team AIST-NEDO at the Valve Task," 2015, http://www.theroboticschallenge.org/sites/default/files/20150604_JD_AIST_valve.JPG.

[16] "DARPA Robotics Challenge, AIST-NEDO's HRP2+ at the rubble," 2015, http://www.theroboticschallenge.org/sites/default/files/AIST-NEDO'sHRP2+attheDebrisPhoto3-Day2resave.jpg.