# Task-Level Teleoperated Manipulation for the HRP-2Kai Humanoid Robot

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

***Abstract*— This paper presents the strategy used by the team AIST-NEDO at the DARPA Robotics Challenge to deal with the designated manipulation tasks by means of a task-level teleoperation of the HRP-2Kai humanoid robot, considering a disaster-hit scenario that is inherently non-structured and a limited communication between the user and the robot. The strategy, based on the information provided by a laser rangefinder and a set of cameras installed at the head and at both hands, consisted in the alignment of 3D models representing the desired manipulation targets with a measured point cloud, in order to provide a reference frame to describe the manipulation motion required for each task. Each motion was carefully planned in advance by assuming minimum information of the object representing the manipulation target. In order to exemplify the before mentioned approach, two representative tasks of the DARPA Robotics Challenge are described, as well as the corresponding results obtained during the competition.**

## I. Introduction and Motivation

Disaster response is attracting attention from the robotics research community, and even more since the Fukushima Daiichi nuclear power plant accident that followed the 2011 Great East Japan earthquake and tsunami. As a concrete materialization of this increasing interest, a challenge was proposed by the American Defense Advanced Research Projects Agency (DARPA) to use robots in disaster-hit facilities that were made too hazardous for direct human operator intervention. It is worth noticing that the challenge did not impose any constraint on the design of the robot, but since the environment (industrial ladders, doors, valves, cars) as well as the tools (levers, drills, hammers) were meant to comply with the human morphology, it was a natural option to develop the necessary means to make the humanoid robots capable of performing inspection and disaster recovering actions inside a non-structured environment [1].

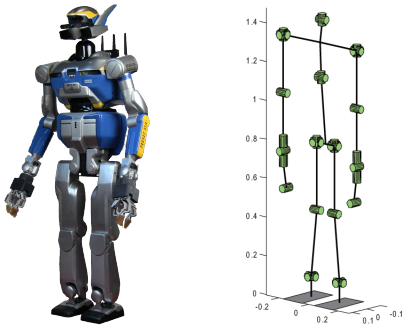
This environment can be considered to be "kind of" known in the sense that we know which actions are required in advance and that we have a rough idea of the spatial distribution of this environment, altered due to the disaster itself. Given these conditions, only very limited assumptions about the structure of the environment can be made beforehand, in contrast to structured scenarios where semantic knowledge of their structure can be leveraged for highly autonomous robots operating in them [2].

R. Cisneros, T. Sakaguchi, S. Kajita, S. Nakaoka, M. Morisawa, K. Kaneko and F. Kanehiro are with the National Institute of Advanced Industrial Science and Technology (AIST), 305-8568 Tsukuba, Japan. { rafael.cisneros, s.kajita, sakaguchi.t, s.nakaoka, m.morisawa, k.kaneko, f-kanehiro } @aist.go.jp

Furthermore, it is also mandatory to consider that within a disaster-hit facility it is not possible to rely on a stable, wide bandwidth wireless communication system with the robot. The signal may be degraded and blackouts may occur frequently. Then, it is not feasible to consider a purely teleoperated robot. First, because of the high dimensionality of its control system, and second because the capabilities of the robot and the operator should include near real-time feedback without disruptions in the communications as well as transmission of large amounts of data to the operator. On the other hand, a fully autonomous robot navigating and interacting in an non-structured environment should include extensive databases of information about possible objects of interest to be found, highly efficient grasping algorithms and the ability to react to unforeseen situations, which are still unsolved problems [3]. Furthermore, failures are critical and up to now no fully autonomous robotic system has shown to be highly reliable, especially under unexpected conditions [4]. A feasible alternative is the development of supervised semi-autonomous high degrees-of-freedom (dof) robotic systems; that is, task-level teleoperated systems in which the operator cognitive burden is minimized by lowering the control space dimensionality [5], such that these operators function as supervisors setting high level goals, assisting the robot with complex perception tasks, directly changing robot parameters to improve its performance and making decisions when facing unexpected situations [2].

## II. Related Work

From some years ago, there has been plenty of research on fully autonomous robots capable of performing tasks in structured environments (kitchens, offices, etc.), as the one presented by Blodow et al [6] or Beetz et al [7]. For that purpose many different control architectures have been proposed, some of them are described by Medeiros [8]. For non-structured environments there is one basic paradigm called *supervised autonomy*, proposed first by Cheng and Zelinsky [9], which has become the current state of art for the DARPA Robotics Challenge (DRC) since the Trials and also for the Finals. During these competitions, each team relied on an Graphic User Interface (GUI) showing the a 3D model of the robot and the environment, in such a way that the operator(s) could control the robot beyond the joint-level, by specifying task-level commands that were robot-centric and/or object centric, as described by the teams Tartan Rescue [10], MIT [11], RoboSimian [12] and ViGIR [3]. In this paper, we describe our implementation of this approach.



(a) Actual robot. (b) Structure.

Fig. 1: HRP-2 Kai humanoid robot.

## III. HRP-2Kai humanoid robot

HRP-2Kai stands for Humanoid Robotics Platform no. 2 Improved (Kai means improvement in Japanese). This humanoid was developed in phase two of the Japanese national project HRP (Humanoid Robotics Project) and was recently improved to be able to cope with disaster response tasks [13].

This robot, depicted in Fig. 1a, has the kinematic structure shown in Fig. 1b. As seen in this diagram, the robot has 32 dof: 6 at each leg, 2 at the waist, two at the head, 7 at each arm and 1 at each hand. This robot features a set of exteroceptive sensors that are actively used during the manipulation tasks: a 3D scanner system built in its head and four cameras, placed at the head, at the back and at each hand. The 3D scanner system was implemented with a Laser Range Finder (LRF) synchronized with the head pitch joint, as shown in Fig. 2a. The hand camera is mounted in each hand as shown in Fig. 2b, together with a LED light and a laser. It is worth to mention that this camera is not lined up with the longitudinal axis at the center of the hand.

## IV. Graphical User Interface for Teleoperation

The Graphical User Interface (GUI) used for the teleoperation of HRP-2Kai was implemented in Choreonoid, an integrated robotics GUI environment [14] [15]. A snapshot of this GUI is shown in Fig. 3. As can be seen, the teleoperation interface features several windows, each one of them corresponding to (1) the scene view, (2) the task sequencer, (3) the head camera, (4) the right hand camera, (5) the left hand camera, (6) the item view, and (7) the property view.


(a) 3D scanner and head camera. (b) Left hand camera.

Fig. 2: Exteroceptive sensors of HRP-2 Kai [13].

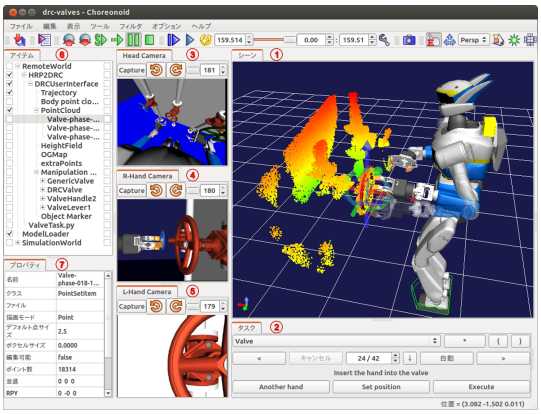


Fig. 3: Teleoperation interface in Choreonoid [16].

The scene view is used to teleoperate the robot through the direct control of a 3D model showing the robot's current configuration, as well as the planned motion represented by one (or several) translucent version(s) of the robot, one per each key configuration. The *point cloud* of data corresponding to the measured points on the surface of the objects in the environment is captured by the 3D scanner and presented also in the scene view. From this information it is possible to extract the *height field* of the floor, used by the walking control system to achieve a stable gait over uneven terrain [17], as well as to detect obstacles and objects of interest in the environment in order to perform a proper whole-body collision-free posture planning, as required by the manipulation task [18].

The object that is the target of the manipulation task can be identified in the point cloud by overlapping a simplified 3D model of it, as shown in Fig. 3. This model, referred as *Manipulation Marker*, provide a reference frame with respect to which the manipulation task can be described once it is aligned with the corresponding points. This alignment can be done manually by using a set of arrows and rings provided by the interface to translate and rotate the marker, or automatically with the aid of a built-in function included in the Point Cloud Library (PCL), used by Choreonoid.

The manipulation motion is specified by providing the attitude (position and orientation) of the hands of the robot, and calculated by solving the whole-body inverse kinematics problem [18]. This motion can be either planned in advance or manually changed during the execution of the task by using *Hand Markers* (3D models of the hands that can also be translated and rotated).

The *task sequencer* is provided by the interface to execute a previously planned manipulation task step by step, such that the complex perception be entrusted to the operator, which can also supervise every step of the task. For some tasks the point cloud does not provide enough information for them to be performed effectively. Then, the operator can also rely on the image captured by every camera, and shown in the corresponding window. Finally, the item window enlists the components required by Choreonoid to implement the teleoperation interface, shown together with their properties.

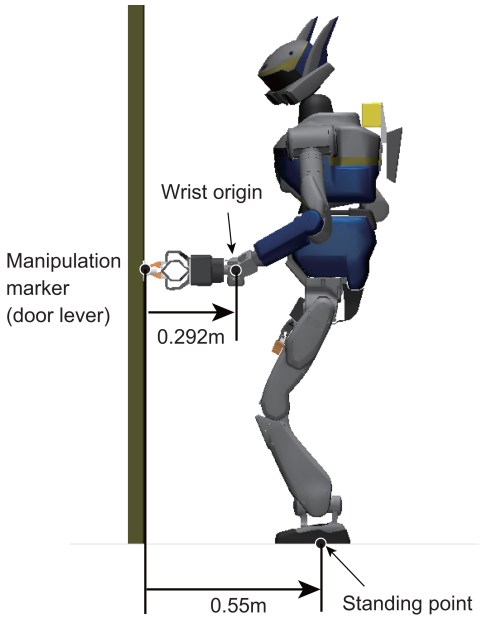



Fig. 4: Open-door pose

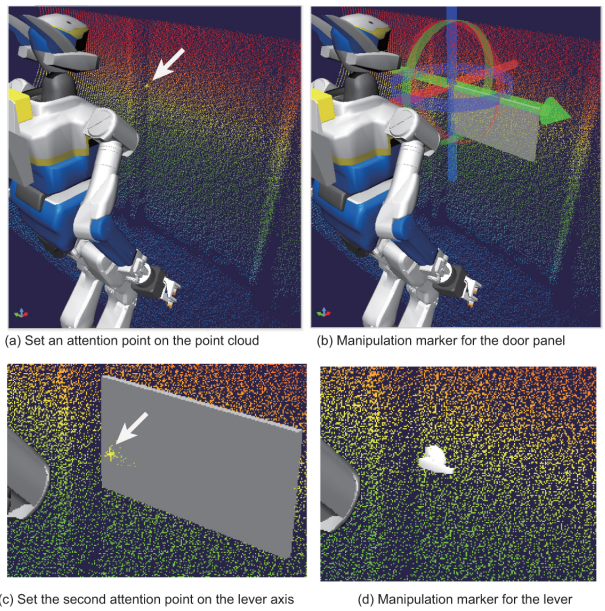



Fig. 5: Detection of the door lever in the control window

By using this interface we implemented all the manipulation tasks required during the DARPA Robotics Challenge: (1) opening a door, (2) turning a valve, (3) cutting a wall with a drill, and (4) a surprise task which consisted in opening a box and pushing a button (rehersal day), pulling down a lever (first day of the competition) or pulling a plug out of a socket and inserting it into another one (second day of the competition). Given that the approach to implement each one of them was similar, we will only explain two representative ones: the door task and the plug task.

## V. Door task

### A. Outline of the door task

At the DRC Finals, the door task was specially important because without passing through it, the robot wouldn't be able to perform the five remaining tasks (valve, wall, surprise, debris or terrain, and stairs).

In general, we can divide the execution of the door task into the following phases:

1) Walking to the front of the door.
2) Grasping the door lever.
3) Turning the lever and opening the door.
4) Walking through the door.

At the end of the second phase, the robot ends up in a standing configuration with a hand grasping the door lever. Let us call this the *open-door pose* as illustrated in Fig. 4. In our approach, we determine a specific standing stance and a wrist attitude with respect to the door lever as shown in the figure. In this way, the remaining phases 3) and 4) start from the same configuration; thus we can use a pre-programmed sequence for operating the door and passing through it. Even if we face variations of the door's geometry, they can be handled by means of a minimum modifications.

In the following subsections, we explain in detail phases 1) and 2); that is, the way to control the robot to achieve the open-door pose by using sensor information and teleoperation. Phases 3) and 4) can be easily implemented. Especially at the DRC Finals, once the door had been opened wide enough, it ended up opening completely by means of gravity and remaining open, since the door hinge was inclined 2.6 degrees away from the vertical. This helped the robots to walk through the door without worrying about any collision with the opened door panel, which would eventually hit the robot by means of the action of the wind if the door hinge was vertical.

### B. Walking to the front of the door

To navigate the robot to the pose specified in Fig. 4, we used the point cloud data measured by the LRF. We took a two-step manual operation to identify the door orientation and the position of the door lever as illustrated in Fig. 5.

First, the operator specifies a point on the left edge of the door panel (Fig. 5(a) the 'x' mark pointed by the arrow). By applying the least square method to the portion of the point cloud located at the right of the specified point, we can calculate the orientation of the door panel. The result is shown by the *door marker*, a gray rectangle in Fig. 5(b). It covers a part of the door panel and we can interactively manipulate it on the point cloud GUI. By adjusting the door marker, we can mask the points of the door panel and extract the points of the door lever as shown in Fig. 5(c). In this way, we can confidently mark the rotation center of the lever (pointed by the arrow). Fig. 5(d) shows the manipulation marker for the door lever placed on the point cloud. Its position and orientation are used to navigate the robot to the desired pose for opening the door.

### C. Grasping the door lever

Now the robot is standing on a good place to grasp the door lever. Instead of grasping the door lever directly, we move the robot hand to an *approach point*, which is placed at a certain distance (0.13m) from the lever. This is because the hand position may not be accurate enough due to the LRF measurement noise, calibration error, and slip occurring while walking.

Figure 6(a) shows the robot at the open-door pose and its hand at the approach point (right window) in Choreonoid simulator. The left window shows the simulated view of the left hand camera. Both images show that the left hand is not appropriate to grasp the door lever (too left). Note that during the real robot control, only the camera view is available.

The operator can manually adjust the hand position by using some GUI buttons: *Left*, *Right*, *Down*, and *Up*, as seen in the left bottom window. In this case, the operator can adjust the hand position by hitting *Right* button several times. Figure 6(b) shows the adjusted hand position to grasp the lever.

Figure 6(c) shows the final state getting a good grasp of the door lever. From this moment, the point cloud based manipulation marker is no longer used. For example, the rotation center of the lever is determined using the hand position calculated from the internal sensors (joint encoders and posture sensors).

### D. Result at the DRC Finals

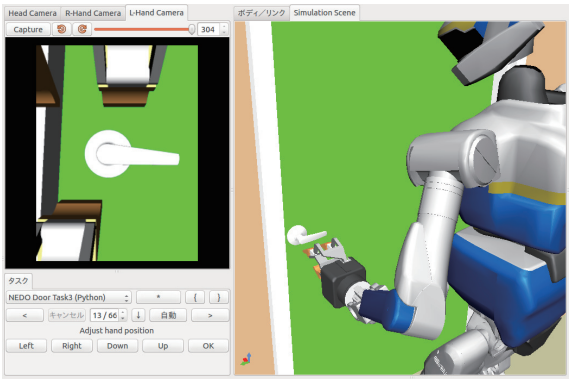
At the DRC Finals 2015, HRP-2Kai cleared the door task at the rehearsal on June 4 and during the challenge on June 6. Fig. 7 shows the robot (a) while scanning with the LRF to obtain a point cloud, (b) walking to the the open-door pose, (c) grasping the door lever, and (d) opening the door successfully. The robot spent 5 minutes and 21 seconds from the LRF scanning to fully pass the door.

During the challenge on June 5, HRP-2Kai failed to operate the door lever at first, requiring a second attempt to grasp it again, which was done by manual teleoperation. During this operation, we experienced a low level control system crash, and the robot fell down (Fig. 8). Since the computer stopped working and the hardware was damaged, we had to abort the challenge for that day.

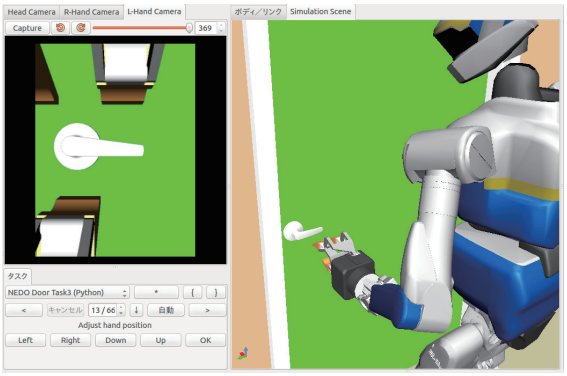
The robot failed to operate the lever because the doors in DRC Finals had different latch properties, as shown in Table I. On day 1, we hard-corded the latch rotation angle as 30 degrees, which worked at the rehearsal. Of course, such implementation is not acceptable for the actual disaster response robots.

TABLE I: Door latch properties at DRC finals

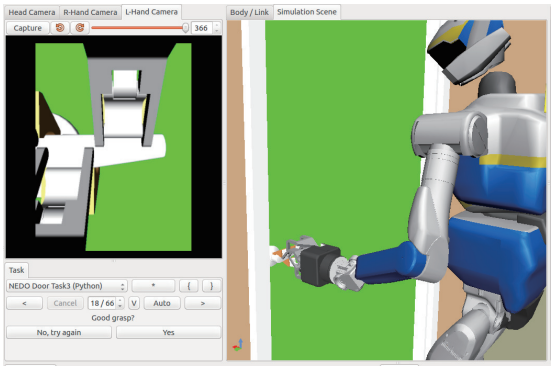
| Course | Latch release angle | Date | Door task result |
|---|---|---|---|
| Green | 30 deg | June 4 (rehearsal) | Success |
| Yellow | 70 deg | June 5 (day 1) | Fail |
| Blue | 50 deg | June 6 (day 2) | Success |




(a) Initial state for door lever approaching




(b) Hand position was adjusted




(c) The door lever grasped

Fig. 6: Approach for grasping the door lever

## VI. Pulling and Inserting a Plug

### A. Outline of the plug task

One of the surprise tasks at the DRC consisted of pulling out a plug from one socket and putting it back into another socket, in a set-up like the one shown in Fig. 9. The separation between both sockets was not known in advance, neither the height at which they were placed. Only the shape of the plug was known in advance.

To carry out this task, we can consider its execution as divided into the following phases:

1) Detection the socket and plug.
2) Grasping the plug.
3) Pulling and adjusting the plug.
4) Inserting the plug.

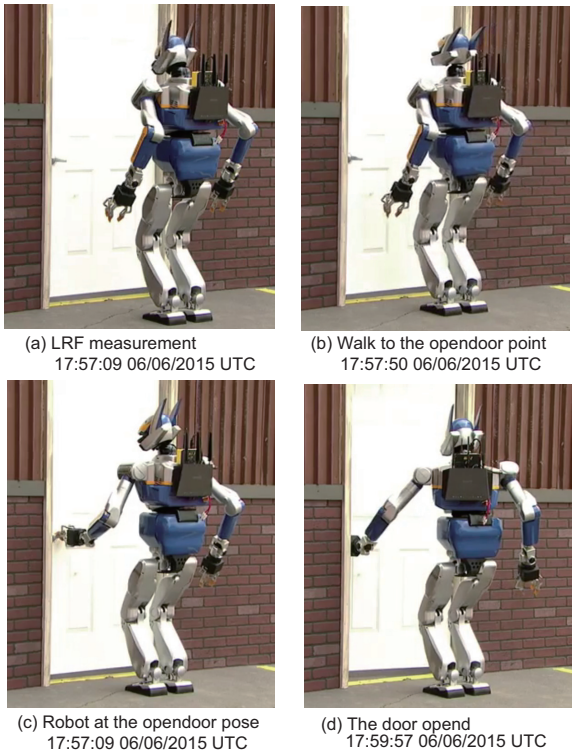



Fig. 7: Door task at DRC Finals on June 6 [19]


Fig. 8: Control system crash at DRC Finals on June 5 [19]

### B. Detection of the socket and plug

In order to perform this task it is first necessary to identify the plug and the socket where it is originally inserted, by placing the corresponding Manipulation Marker within the point cloud. The reason to identify both objects instead of just the plug is because almost half of it is not visible, as it is inside of the socket. As a consequence, the matching precision attained by aligning just the plug is lower than the one attained by aligning both objects.

To do that we first detect all the planes in the scene, assuming that one of them will correspond to the wall where the sockets are installed. Then, given that the front face of the sockets is parallel to this wall, it is possible to calculate their orientation with respect to the robot. Having done this, one point belonging to the plug can be manually selected in order to compute an approximate initial position for the Manipulation Marker representing the socket and the plug.

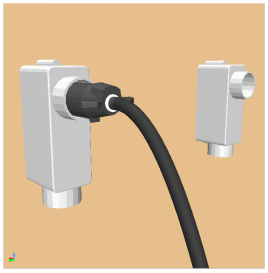


Fig. 9: Plug Task.

This initial position can be later refined, after the robot has arrived to the desired stance and after the point cloud has been adjusted by using one hand of the robot as a reference (given that its attitude can be calculated from the internal sensors), as shown in Fig. 10. This is because the measured point cloud has some intrinsic noise, mainly caused as a result of the sunlight.

### C. Grasping the plug

Having detected an approximate attitude for the socket, the robot first approaches the plug with one hand while aligning the camera installed at the other hand with the axis of the plug. By doing this it is possible for the operator to make slight adjustments of the grasping hand by using the visual information provided by the hand camera. Fig. 11 shows this configuration for the situation in which the plug is initially inserted into the left socket. Then, the robot uses its right hand to grasp the plug and the left hand for the camera. Were the plug initially inserted into the other socket, then the role for both hands would be reversed.

The size of the visible part of the plug is not that big compared to the hand of the robot, and because of that the allowable tolerance for grasping the plug is small. Then, it is required to grasp it at the point in which the hand also touches the socket. This can be done by translating the hand along the axis of the plug until sensing 30 N of force (enough for considering that it hits to the socket). This strategy is effective (when adjusted properly) as the grasp can be done every time at the desired point with enough precision even in the presence of uncertainties, as shown in the dynamical simulation depicted in Fig. 12.

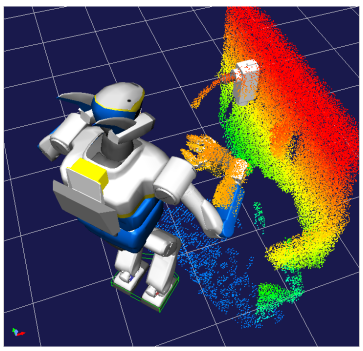


Fig. 10: Detection of the socket and the plug.

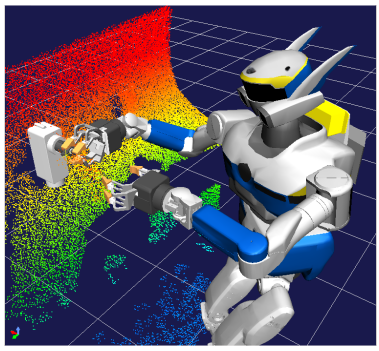


Fig. 11: Configuration of the robot before grasping the plug.

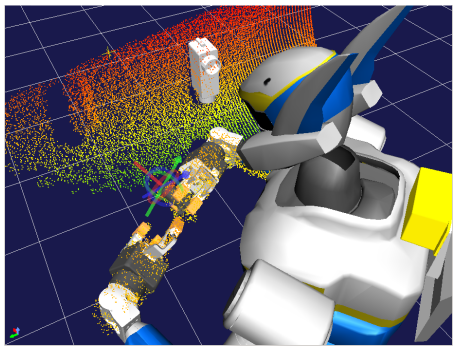


Fig. 13: After pulling the plug its attitude is adjusted.

### D. Pulling and adjusting the plug

After grasping the plug, the robot has to pull it. Due to the waist motion occurring during the stabilization of the robot, the pulling motion may not be performed exactly along the axis of the plug, and this one may hit the inner walls of the socket while being pulled, modifying the planned relative attitude between the plug and the grasping hand.

For this reason, before inserting the plug into the other socket, the robot first brings the plug in front of its chest, takes an updated point cloud, and uses the camera placed at the head and at the other hand to look at the plug from two perpendicular directions, as seen in Fig. 13. By using the point cloud, together with the visual information of both cameras, it is possible to fix the actual attitude of the Manipulation Marker representing the plug to match the attitude of the real one with respect to the hand.

### E. Inserting the plug

Once this relative attitude is known it is possible to align the plug with the destination socket, whose position can also be approximated by adjusting the corresponding marker within the point cloud. However, it is still worth to consider that this position may not be accurate enough, making it unfeasible to attempt an immediate insertion. Instead, the camera at the other hand can be used once again together with the camera at the head to look at the plug from two different points of view (as seen in Fig. 14), in such a way that the operator can use this visual information to slightly adjust the position of the plug before inserting it.

Then, once the task is completed, the grasping hand opens and an updated point cloud is taken again, in order to plan the returning motion without hitting the cable of the plug.

### F. Result at the DRC Finals

During the second day of the DRC Finals 2015 (June 6), HRP-2Kai was able to carry out the plug task within 16 minutes and 34 seconds, mainly because we were supervising every motion of the robot in order to prevent unexpected collisions, and also due to the blackouts that prevented us to make a faster identification of the plug in the point cloud. Even without knowing the disposition of the sockets within the field, HRP-2Kai was able to identify the plug, approach to it, grasp it, pull it and insert it into the other socket. Then, before returning to its home position, the hand of the robot grazed the plug and it fell down. However, the point corresponding to the task was granted, as the plug was inserted into the socket when the robot opened its hand. Some snapshots taken during the task at the DRC Finals are shown in Fig. 15, together with the description of the current process in accordance with the explanation given before.

Table II summarizes the performance obtained by the teams that were able to accomplish the plug task. For comparison effects, the following information is presented: the place obtained by each team during the competition, the time spent during the execution of the plug Task, the effective time of the task (by considering only the time in which the robot is moving) and the number of adjustments performed during the insertion of the plug.

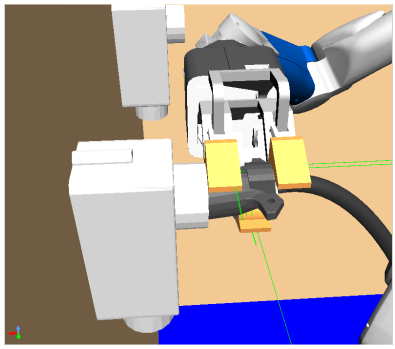


Fig. 12: The plug being effectively grasped.

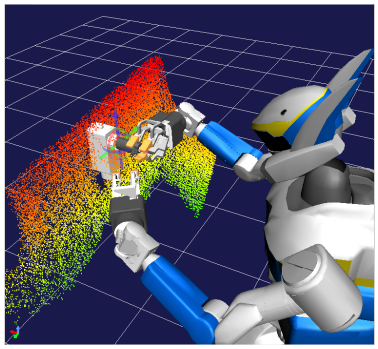


Fig. 14: The plug is inserted by using visual feedback.

(a) Identify plug in socket (b) Adjust point cloud offset

(c) Prepare for pre-grasping (d) Align hand to grasp

(e) Pull out the plug (f) Identify plug in hand

(g) Prepare for pre-insertion (h) Plug is inserted

Fig. 15: Plug task at DRC Finals on June 6 [19]

TABLE II: Teams that completed the plug task.

| Team | Place | Task time | Effective time | Insertion adjust. |
|---|---|---|---|---|
| KAIST | 1 | 11:01 | 2:18 | 14 |
| IHMC Robotics | 2 | 6:31 | 2:32 | 8 |
| Tartan Rescue | 3 | 18:33 | 3:21 | 17 |
| NIMBRO Rescue | 4 | 8:16 | 2:29 | 16 |
| WPI-CMU | 7 | 5:07 | 1:42 | 7 |
| AIST-NEDO | 10 | 16:34 | 1:34 | 3 |

## VII. Conclusions

The task-level teleoperation strategy presented in this paper was successfully applied to the HRP-2Kai humanoid robot, which was able to carry out all the manipulation tasks proposed for the DRC, even though there was no accurate model of the environment. Our principal restriction was the time limit, considering that most of the time spent during the task was the one required by the manual identification of the objects in the environment and by the verification of the robot motion by the operator.

With respect to the plug task it is worth to notice that even though its execution lasted 16:34 minutes, the effective time was just 1:34 and the number of adjustments required to insert the plug was only 3. This was the result of assuring a stable grasp of the plug, as well as the use of markers to identify the plug and the sockets.

As a future work we want to improve our recognition system, in order to speed up the execution of every task.

## Acknowledgment

This work was partially supported by NEDO's "research and development project on disaster response robots".

## References

[1] K. Bouyarmane, J. Vaillant, et al. Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2012.

[2] Stefan Kohlbrecher, Alberto Romay, et al. Human-robot teaming for rescue missions: Team vigir's approach to the 2013 darpa robotics challenge trials. *Journal of Field Robotics*, 32(3), 2014.

[3] Alberto Romay, Stefan Kohlbrecher, et al. Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

[4] Daniel P. Stormont and Vicki H. Allan. Managing risk in disaster scenarios with autonomous robots. *Systemics, Cybernetics and Informatics*, 7(4), 2009.

[5] Kapil D. Katyal, Christopher Y. Brown, et al. Approaches to robotic teleoperation in a disaster scenario: From supervised autonomy to direct control. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[6] Nico Blodow, Lucian Cosmin Goron, et al. Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[7] Michael Beetz and Maren Bennewitz. Planning, scheduling, and plan execution for autonomous robot office couriers. In *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments, volume Workshop Notes*. AAAI Press, 1998.

[8] Adelardo A. D. Medeiros. A survey of control architectures for autonomous mobile robots. *Journal of the Brazilian Computer Society*, 4(3), 1998.

[9] Gordon Cheng and Alexander Zelinsky. Supervised autonomy: A paradigm for teleoperating mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1997.

[10] Christopher Dellin, Kyle Strabala, et al. Guided manipulation planning at the darpa robotics challenge trials. In *Int. Symposium on Experimental Robotics (ISER)*, 2014.

[11] Maurice Fallon, Scott Kuindersma, et al. An architecture for online affordance-based perception and whole-body planning. Technical report, Massachusetts Institute of Technology, 2014.

[12] Paul Hebert, Jeremy Ma, et al. Supervised remote robot with guided autonomy and teleoperation (surrogate): A framework for whole-body mobile manipulation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[13] Kenji Kaneko, Mitsuharu Morisawa, et al. Humanoid robot hrp-2kai - improvement of hrp-2 towards disaster response tasks. In *IEEE-RAS Int. Conf. on Humanoid Robots (submitted)*, 2015.

[14] Shin'ichiro Nakaoka. Choreonoid. `http://www.choreonoid.org/en/`, 2015.

[15] Shin'ichiro Nakaoka. Choreonoid: Extensible virtual robot environment built on an integrated gui framework. In *IEEE/SICE International Symposium on System Integration*, 2012.

[16] Shin'ichiro Nakaoka, Mitsuharu Morisawa, et al. Task sequencer integrated into a teleoperation interface for biped humanoid robots. In *IEEE-RAS Int. Conf. on Humanoid Robots (submitted)*, 2015.

[17] Mitsuharu Morisawa, Shuuji Kajita, et al. Balance control based on capture point error compensation for biped walking on uneven terrain. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2012.

[18] Oussama Kanoun, Florent Lamiraux, et al. Kinematic control of redundant manipulators: generalizing the task priority framework to inequality tasks. *IEEE Transactions on Robotics*, 27(4), 2011.

[19] DARPA. Darpa robotics challenge finals 2015. `http://www.theroboticschallenge.org/`, 2015.