

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

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

Abstract— This paper presents the strategy used by our 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 a 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.

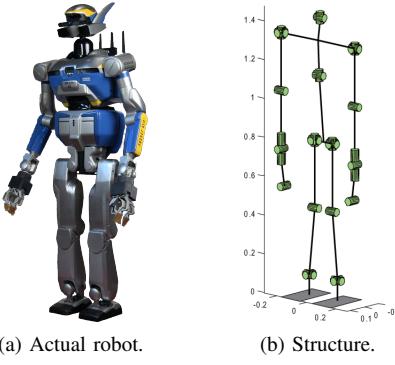


Fig. 1: HRP-2 Kai humanoid robot.

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.

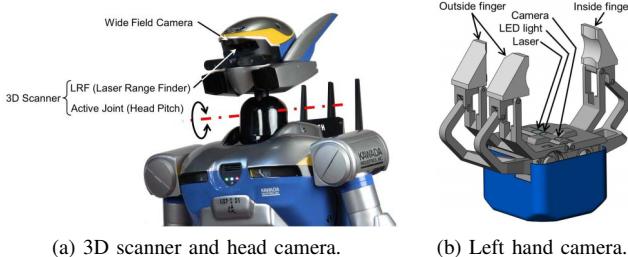


Fig. 2: Exteroceptive sensors of HRP-2 Kai [13].

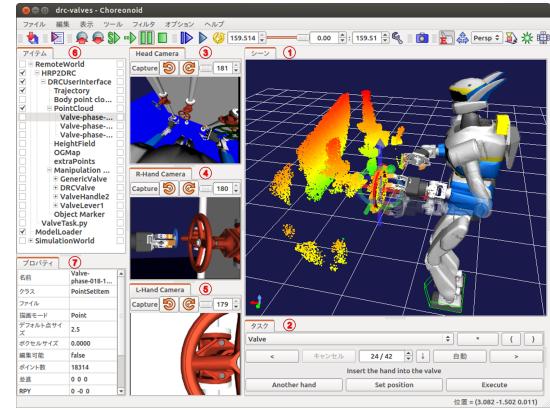


Fig. 3: Teleoperation interface in Choreonoid [16].

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.

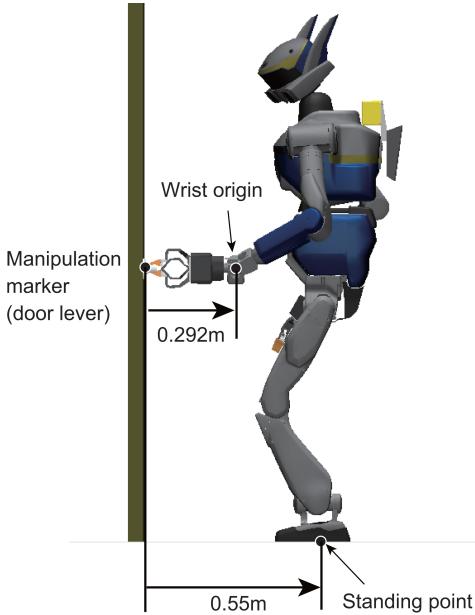


Fig. 4: Open-door pose

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 (rehearsal 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) Walk to the front of the door.
- 2) Grasp the door lever.
- 3) Turn the lever and open the door.
- 4) Walk 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. In this way, even if we face variations of the door's geometry, they can be handled by means of a minimum modifications.

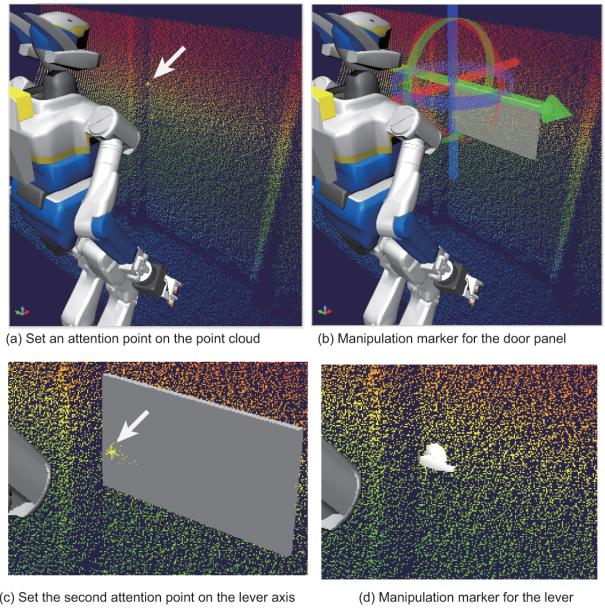


Fig. 5: Detection of the door lever in the control window

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 during a general situation by means of the action of the wind.

B. Walk 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. Grasp 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 appropriate to grasp the door lever. 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 DRC Finals

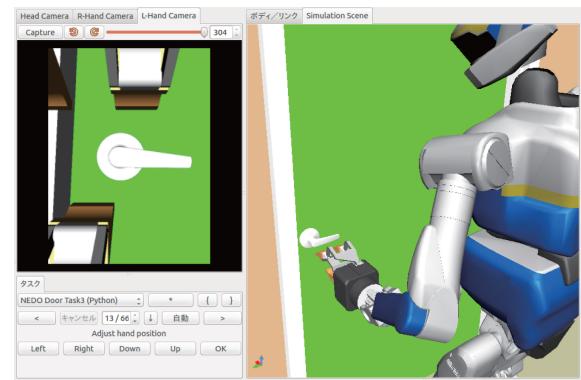
At the DRC Finals 2015, HRP-2Kai cleared the door task at the rehearsal on June 4 and during the challenge on June 6. Fig. 7 shows the robot (a) while scanning with the LRF to obtain a point cloud, (b) walking to the the open-door pose, (c) grasping the door lever, and (d) opening the door successfully.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

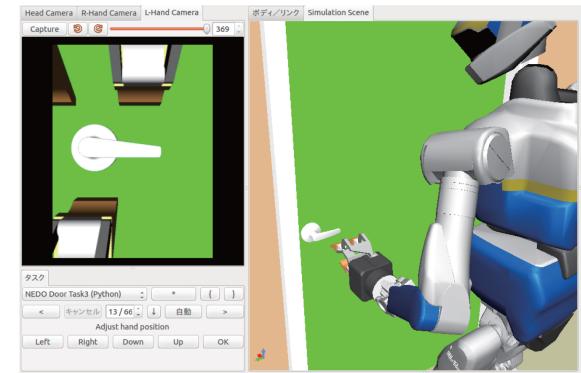
The robot failed to operate the lever because the doors in DRC Finals had different latch properties, as shown in Table I. On day 1, we hard-corded the latch rotation angle as 30 degrees, which worked at the rehearsal. Of course, such implementation is not acceptable for the actual disaster response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

TABLE I: Door latch properties at DRC finals

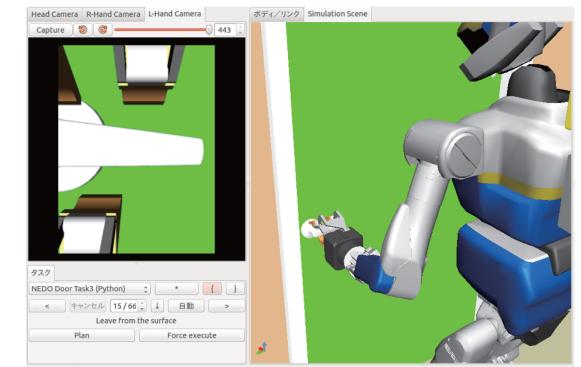
Course	Latch release angle	Date	Door task result
Green	30 deg	June 4 (rehearsal)	Success
Yellow	70 deg	June 5 (day 1)	Fail
Blue	50 deg	June 6 (day 2)	Success



(a) Initial state for door lever approaching



(b) Hand position was adjusted



(c) Touch the door panel

Fig. 6: Approach for grasping the door lever

VI. PULLING AND INSERTING A PLUG

One of the surprise tasks at the DARPA Robotics Challenge 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.

A. 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, making it too difficult just to match the plug, given the low density of points belonging to it due to its small size.

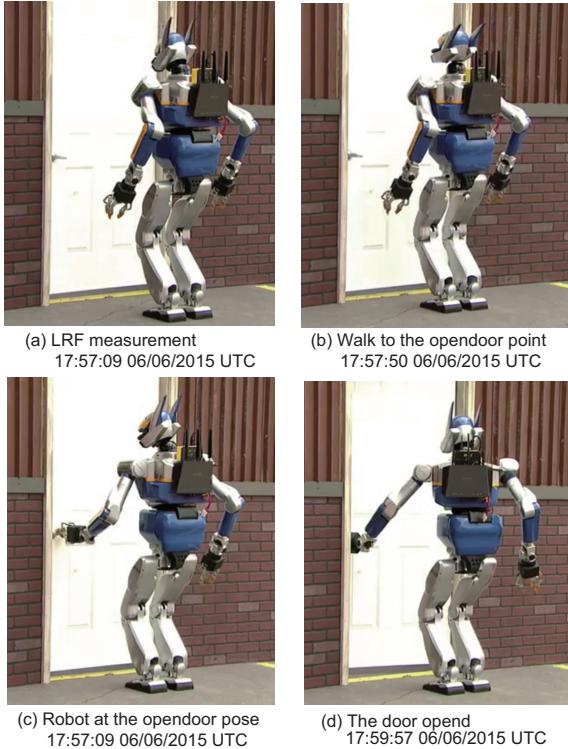


Fig. 7: Door task at DRC Finals on June 6



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

By using a PCL's built in function it is possible to detect all the planes in the scene, one of them being 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 use the corresponding plane equation to calculate the socket's orientation with respect to the robot, in such a way that the plug is directed towards it. 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. This one is later refined, after the robot has arrived to the desired stance and the point cloud has been adjusted by using the robot's hands as a reference, as seen in Fig. 10 (and explained in Subsection ??).

B. 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. This is to be able to make slight adjustments of the grasping hand by using visual feedback.

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 tolerance for grasping the plug is minimum. Then, it is required to grasp the plug at the point in which the medial side of the hand touches the cylindrical shaped part of the socket. This can be done by reducing the angle of the gripper and then, by moving the hand along the axis of the plug until sensing 30 N of force (enough for considering that it arrived to the socket). This configuration of the robot just before moving the hand towards to the socket is depicted in Fig. 11.

This strategy is very effective (when adjusted properly) as the grasp can be done every time at the desired point even in the presence of uncertainties, as shown in the dynamical simulation depicted in Fig. 12.

C. Pulling and adjusting the plug

After grasping the plug, the robot pulls the plug. Due to the balancing process this motion may not be performed exactly along the axis of the plug, and it may hit the inner walls of the socket, modifying the planned relative attitude between the plug and the 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 this information it is possible to fix the actual attitude of the plug with respect to the hand.

D. Inserting the plug

Once the Manipulation Marker representing the plug is fixed at the hand, it can further be properly aligned with the destination socket. This one can be represented with another marker (Object Marker), which can be adjusted within the point cloud before taking the plug to the pre-insertion position. However, because of inaccuracies of the point cloud, the camera at the other hand is used once again together with the camera at the head to look at the plug, and use this visual feedback to adjust its position to properly



Fig. 9: Plug Task.

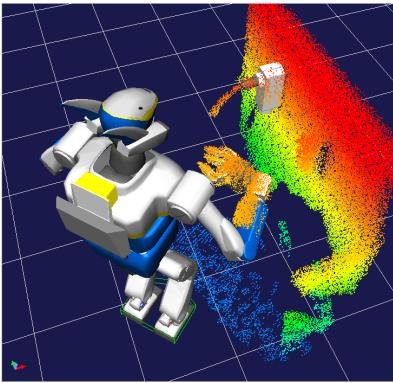


Fig. 10: Detection of the socket and the plug.

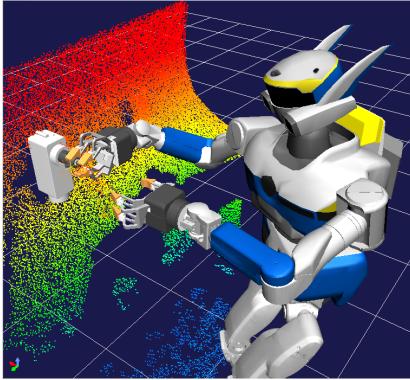


Fig. 11: Configuration of the robot before grasping the plug.

insert the plug within few movements. This is represented in Fig. 14.

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

E. Results

With respect to the plug task, during the DRC Finals we were able to complete the task and get the point within 16 minutes and 34 seconds, mainly because we were supervising every motion of the robot in order to prevent any collision

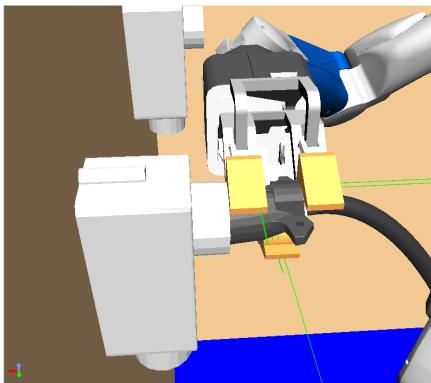


Fig. 12: The plug being effectively grasped at the desired point.

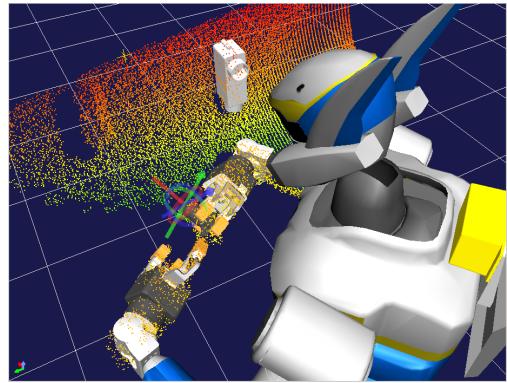


Fig. 13: After pulling the plug its attitude is adjusted.

with the environment, and also due to blackouts. 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.

VII. CONCLUSIONS

With respect to the plug task it is worth to notice that even though its execution lasted about 16 and a half minutes due to recognition, verification and blackouts, the number of slight adjustments required to insert the plug was only 3. This was the result of assuring a stable grasp by following the strategy explained before, as well as the use of a reference representing the target socket. Other teams that chose to grasp the cable instead of the plug itself required between 8 and 14 adjustments in order to complete the insertion.

REFERENCES

- [1] K. Bouyarmane, J. Vaillant, K. François, and A. Kheddar. Exploring humanoid robots locomotion capabilities in virtual disaster response scenarios. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2012.
- [2] Stefan Kohlbrecher, Alberto Romay, Alexander Stumpf, Anant Gupta, and Oskar von Stryk. 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, David C. Conner, Alexander Stumpf, and Oskar von Stryk. Template-based manipulation in unstructured environments for supervised semi-autonomous humanoid robots. In *IEEE-RAS Int. Conf. on Humanoid Robots*, 2014.

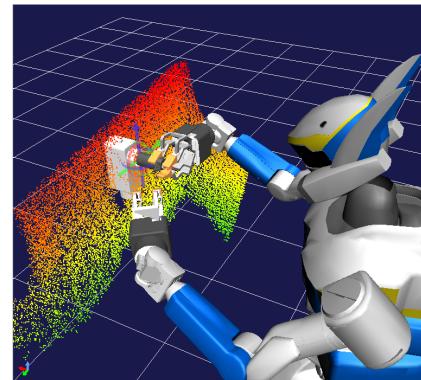


Fig. 14: The plug is inserted by using visual feedback.

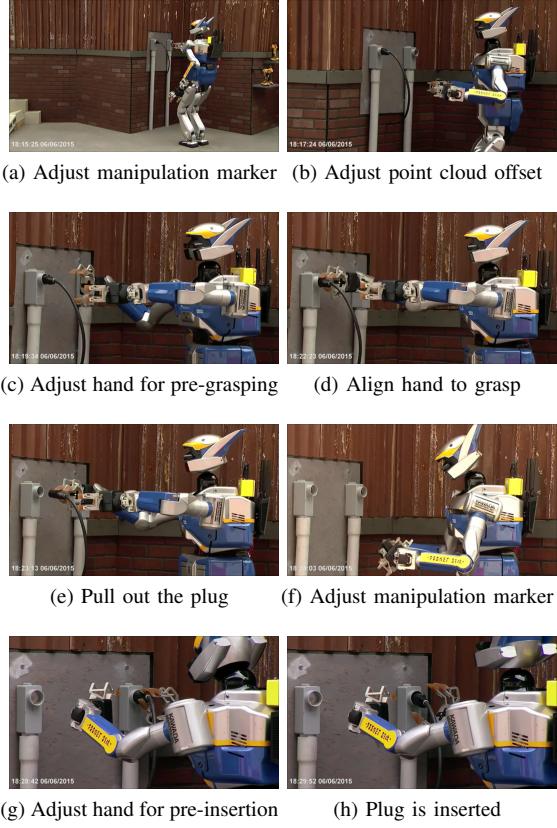


Fig. 15: Plug Task at the second day of the DARPA Robotics Challenge [19]

- [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, Steven A. Hechtman, Matthew P. Para, Timothy G. McGee, Kevin C. Wolfe, Ryan J. Murphy, Michael D.M. Kutzer, Edward W. Tunstel, Michael P. McLoughlin, and Matthew S. Johannes. 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, Zoltan-Csaba Marton, Dejan Pangercic, Thomas Rühr, Moritz Tenorth, and Michael Beetz. 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, Galen Clark Haynes, David Stager, and Siddhartha Srinivasa. Guided manipulation planning at the darpa robotics challenge trials. In *Int. Symposium on Experimental Robotics (ISER)*, 2014.
- [11] Maurice Fallon, Scott Kuindersma, Sisir Karumanchi, Matthew Antone, 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, James Borders, Alper Aydemir, Max Bajracharya, Nicolas Hudson, Krishna Shankar, Sisir Karumanchi,

Bertrand Douillard, and Joel Burdick. 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, Mitsuhiro Morisawa, Shuuji Kajita, Shin'ichiro Nakaoka, Takeshi Sakaguchi, Rafael Cisneros, and Fumio Kanehiro. 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, Shuuji Kajita, and Kazuhito Yokoi. Intuitive and flexible user interface for creating whole body motions of biped humanoid robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1675–1681, Taipei, Taiwan, October 2010.
- [16] Shin'ichiro Nakaoka, Mitsuhiro Morisawa, Rafael Cisneros, Takeshi Sakaguchi, Shuuji Kajita, Kenji Kaneko, and Fumio Kanehiro. Task sequencer integrated into a teleoperation interface for biped humanoid robots. In *IEEE-RAS Int. Conf. on Humanoid Robots (submitted)*, 2015.
- [17] Mitsuhiro Morisawa, Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kanako Miura, and Kazuhiro Yokoi. 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, and Pierre-Brice Wieber. 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.