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

Development of Humanoid Robot System for Disaster Response Through Team NEDO-JSK's Approach to DARPA Robotics Challenge Finals

Yohei Kakiuchi, Kunio Kojima, Eisoku Kuroiwa, Shintaro Noda,
Masaki Murooka, Iori Kumagai, Ryohei Ueda,
Fumihito Sugai, Shunichi Nozawa, Kei Okada, Masayuki Inaba

Abstract— **This paper presents Team NEDO-JSK's approach to the development of novel humanoid platform for disaster response through participation to DARPA Robotics Challenge Finals. This development is a part of the project organized by New Energy and Industrial Technology Development Organization. Technology for this robot is based on the recent research of high-speed and high-torque motor driver with water-cooling system, RTM-ROS inter-operation for intelligent robotics, and generation of full-body fast dancing motion, due to the generic 10 year's research of HRP-2 as a platform humanoid robot. Development target is the robot support in a variety of unsafe human tasks teleoperated by humans in case of a disaster response, equipped with body structure capability for use of human devices and tools in human environment, performance for dynamic full-body actions covering human-sized speed and power, and basic function for intelligent and integrated robot platform system for performing various tasks independently. we also describes NEDO-JSK team's approach to design methodology for robot hardware and architecture of software system and user interface for DRC Finals as a test case of disaster response.**

I. Introduction

In this paper, we describe our approach to development of a disaster response robot. It is promoted as a part of the project, "the International R&D and Demonstration Project in Environment and Medical Device Sector / the International R&D and Demonstration Project on Robotics Field / the R&D on Disaster Response Robot (USA)", organized by New Energy and Industrial Technology Development Organization (NEDO). We present the developed robot, named JAXON, which has participated in DARPA Robotics Challenge Finals (DRC) [1] as NEDO-JSK team. We also present our software system integration for a disaster response robot,

Purposes of NEDO's project are development and substantiative experiments of robot system cooperating with over-sea's partners. They aim to solidify Japanese position in the robotics field in the world, and spread our robot system to the world. From experiences of kind cooperation with other countries at the Great East Japan Earthquake, this project especially aim to contribute to the development of robotics applications for humanitarian assistance and disaster recovery. They are needed for early construction of the corresponding system to various disasters that occur in various parts of the world with cooperation among relevant organizations.

NEDO's project consists of three sub-projects. One is a project for development of new humanoid robots for disaster response. Our robot was developed through this sub-project. Another one is a competition for disaster response humanoid robots using computer simulation. This is described in Section V. Last one is research and development of tasks for disaster response humanoid robots.

Fig. 1. 2nd Day of DRC Finals, 6th, June, 2015. (Images Are Captured from youtube channel of DARPAtv.)

In this paper, we describes NEDO-JSK team's approach to the hardware design methodology for robot hardware and the architecture of software system and user interface for DRC Finals as one of test cases of disaster response. Then, we conclude required hardware design methodology and software architecture for disaster response humanoid robot through DRC Finals.

II. Development of Disaster Response Robot

A. Hardware Concept of Robot

For this project, we decided to build new robot which is aimed as the humanoid robot platform for assistance and support in a disaster site. For assistance and supports in a disaster site, such as outdoor condition, desired points for a humanoid robot suitable for the environment are listed below.

- A robot should have similar configuration to a human worker, which have two arms and two legs. they are

Y. Kakiuchi, K. Kojima, E. Kuroiwa, S. Noda, M. Murooka, I. Kumagai, R. Ueda, F. Sugai, S. Nozawa, K. Okada and M. Inaba are with Graduate School of Information Science and Technology The University of Tokyo, 7-3-1 Hongo, Bunkyo-Ku, 113-8656 Tokyo, Japan {youhei, k-kojima, kuroiwa, s-noda, murooka, iori, ueda, sugai, nozawa, k-okada, inaba}@jsk.imi.i.u-tokyo.ac.jp

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

Fig. 2. Appearance of JAXON

- similar proportions to human, because a human like robot can work easily in the infrastructure matched to the human body structure.
- A robot should have a power source, such as batteries, inside the body. It should perform task in stand alone without a tether.
- A robot should locomote not only with two legs, but also using four limbs, two arms and two legs, on rough terrain. It should perform task while getting up even if falling down.

We decided to develop humanoid robot named "JAXON" for a humanoid robot platform for disaster response with listed three features. It was decided to use existing powerful motors with water cooling driver system, in order to achieve motions comparable performance with human. This feature derived from developed humanoid robot "STARO" [2],

Fig. 2 show the appearance of JAXON. The basic specifications are shown in Fig. 4. JAXON is 188[cm] tall and weights 127[kg]. JAXON's total degree of freedom is 32 without hands (6 D.O.F. legs, 8 D.O.F. arms, 3 D.O.F. torso, and 2 D.O.F. neck) as shown in Fig. 4.

The length of leg links was decided from the seat height the car. The length of leg links is 380[mm]. We decided to use a water-cooling double motor Fig. 3 units for pitch joints of crotch and knee, and normal (air-cooling) single motor units for other joints.

Gears and actuators were chosen with consideration for required torques. Required torques calculated by the simulation, stepping a stair of 400[mm] height. Table I shows specification of JAXON, such as max torques and gear reducer types. In this table, "*" shown on ID indicates that the joint is using double-motors unit. The joint's max torques shows smaller value of torques calculated from the current limit and ratcheting torque of the gear reducer. We selected CSD32-100 for crotch-pitch, knee-pitch, torso-roll, torso-pitch joint, CSD25-160 for crotch-roll joint, and CSD20(SHD)-160 in other joints. More detail descriptions about hardware design concepts are in [3].

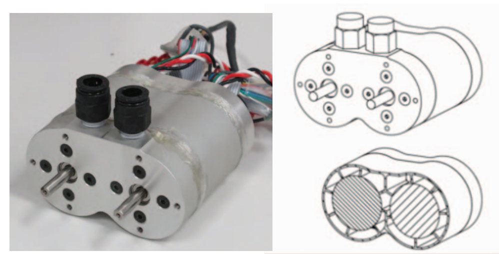

Fig. 3. Appearance of water cooled double motor

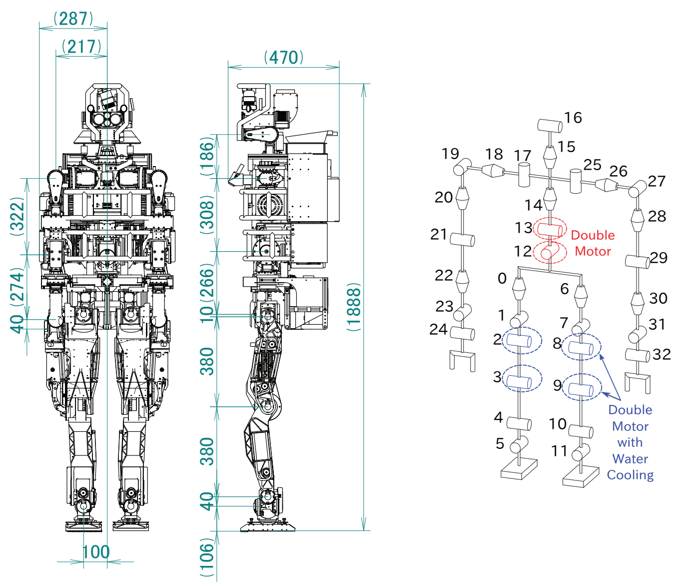

Fig. 4. Dimension and D.O.F. configuration of JAXON

B. Motor drive system

Our motor drive system is based on STARO, which came from the design of high torque and speed leg [4]. Joint driving system of each joint consists of a motor, an adjunctive board for the motor, a communication and control board, a switching board, a water cooling head, and an absolute encoder (Fig. 5). The water cooling head is for cooling FET on a switching board for driving a motor. All motors on the robot are the same model. The model is MAXON EC-4pole, the diameter is 30 [mm], nominal output is 200 [W]. Some joints needed high torque are driven by the double motor unit and cooled with water using water jacket.

The adjunctive board have features gathering signals from a motor and sending them to a communication and control board. Signals gathered by the board are hall effect sensor signal as a DC brushless motor, motor encoder pulse (500 pulse per round), casing temperature from a serial digital

TABLE I
JAXON SPECIFICATION

Limb	ID	Min angle [deg]	Max angle [deg]	Max torque [Nm]	GearType
Head	15	-32.0	30.0	220	CSD20-160
	16	-32.0	39.0		
Torso	*12	-8.0	8.0	787.2	CSD32-100
	*13	0.0	32.0		
	14	-60.5	60.5	393.6	SHD20-160
Arm	17	-17.6	81.4	220	CSD20-160
	18	-180.0	180.0		
	19	-180.0	-15.8		
	20	-180.0	180.0		
	21	-125.5	60.0		
	22	-180.0	180.0		
	23	-89.0	87.0		
	24	-80.0	59.0		
Leg	0	-58.8	62.9	220	SHD20-160
	1	-41.5	30.0	393.6	CSD25-160
	*2	-121.4	45.0	492	CSD32-100
	*3	0.0	158.9		
	4	-79.4	84.4	220	CSD20-160
	5	-60.0	60.0		

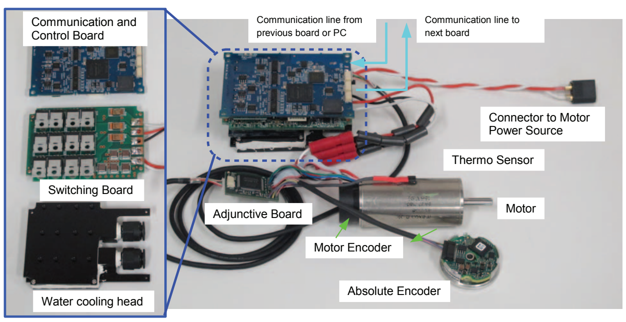

Fig. 5. Joint control system for each joint

output thermometer, and a joint angle from an absolute encoder. Using absolute encoder is Heidenhain ECI 1118, this is light weight and thin.

The communication and control board have features receiving reference joint angle and speed for PD joint control loop from "Control PC", sending back servo status, joint angle calculated from encoder, motor speed and absolute encoder angle to "Control PC". This board also performs PD control of a joint, calculating target motor current from difference angle between a reference joint angle and an angle from motor encoder.

The switching board consists of FET bridges for switching currents in order to drive a brushless DC motor, gate drive circuits for FET, and motor current sensors for each phase. Used model of FET was chosen accoring to power loss on FET such as resistance loss, loss from output capacity, and recovery loss of diode. For legs and torso, IRF B4110 was used as switching devices, and two FET were used for each phase. For arms, IRF B4310Z was used, and one FET was used for each phase.

For the communication between a "Control PC" and a communication and control board, PCI adapter board mounted on a the PC communicates to a communication and control board. We use a 4-pole cable for connections between the boards using RS-422 as electrical layer, it contains upstream lines and downstream lines. Each board have 2 physical ports and are connected in the daisy chain configuration. For error correction, Reed Solomon (RS) code is adopted. 64 [bit] data and some headers are encoded by RS code to 120 [bit] packet. Error correction is performed at communication between each board, so there is 10 [us] of latency time between each board. Baud rate for data transfer is 26.7 [Mbps], effective data transfer rate which contains error correction is about 4 [Mbps].

C. Power Source of Robot

Even though its small and light-weight feature, we developed the power system for robots adopting lithium ferrite batteries. The power system separated into a high voltage system and a low voltage system. A high voltage battery is 72.6 [V] (using serial connected 22 cell, capacity 2.5 [Ah], 2.0 [kg], 295 x 58 x 85 [mm]) for motors. A low voltage battery is 12.8 [V] (using serial connected 4 cell, capacity 15 [Ah], 1.9 [kg], 185 x 50 x 185 [mm]) for computers and control boards. Fig. 6 show using batteries.

We use 3 of high voltage batteries for robot motion and 2 of low voltage ones for computers, and 3 of low voltage ones for control boards. They covered about 80 minutes operation for tasks of DRC Finals.

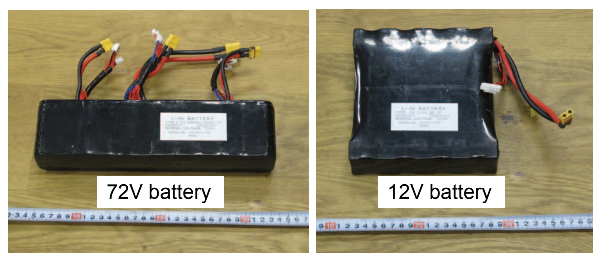

Fig. 6. Batteries for JAXON

D. Robot Hardware Control System

Whole robot body is controlled by one computer on the body, called "Control PC". "Control PC" has 4-core Core i7-2600K 3.40 GHz and 8GB memory, using Linux low-latency kernel for a real-time control loop.

The main control loop sends and receives data from "Control PC" to each communication and control board. This loop keeps 1000 [Hz] as a real-time control loop. A shared memory is used for sending data and receiving command from upper layer. The main control loop writes data and reads command from the shared memory, upper layer writes command and reads data to the shared memory. This is used as inter process communication for software on different layer. Fig. 7 shows the robot control system inside the robot body.

A 6-axis force sensor communicate with "Control PC" through another PCI board. A IMU sensor communicate with "Control PC" through USB-serial conversion adapter. Update rate of a 6-axis force sensor and a IMU is 1000 [Hz]. These sensor data also were written to shared memory in order to send data to control software on upper layer. Using 6-axis force sensor is JR3, 67M25S3M, nominal force is 1200 [N] for z-axis, 600 [N] for x,y-axis, nominal torque is 40 [Nm]. Using IMU sensor is Xsens MTi-300.

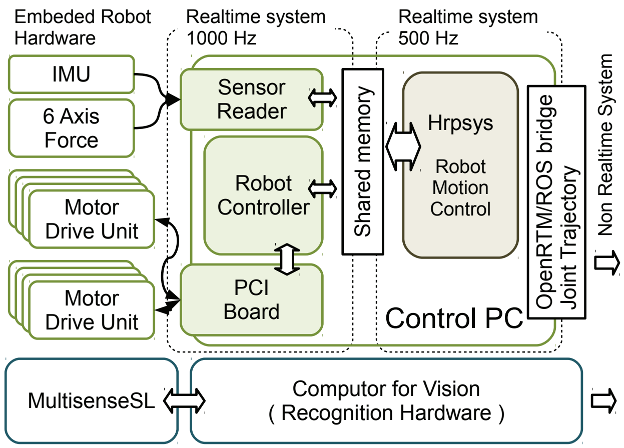

Fig. 7. Robot hardware control system

III. Development of Software System

The structure of our software system is divided into four layers by the abstraction level. Fig. 8 shows whole software system described in this section. It also indicates scopes of four layers.

In order to shorten development time, we used many open-source software libraries, and integrated them to our previous works. we also open our software [1], [2] .

Software layer which directly operates the robot hardware is 'Hardware Control' layer. This layer is connected to the second layer with hardware abstraction interface using shared memory. The second layer is connected to the upper layer with the interface using ROS [5] and OpenRTM [6]. Communication using OpenRTM is converted to ROS communication using openrtm_ros_bridge [3]. The third layer and the fourth layer are layers for using application software with ROS system. The third layer is software application layer without limited communication to the robot. The fourth layer is also software application layer with limited communication. The third layer and the fourth layer are communicated with developed "Tunnel" program in order to control transferred data size. Detail description of each layer is listed below.

1) Hardware Control layer: In this layer, controlling the robot hardware is performed. We can directly control the robot hardware without upper layer software. "Soundness check for joints" is a program for checking soundness of joints just after rebooting system. Checking items are reading data from sensors, moving joint, and confirming correspondence motor encoder with absolute encoder. "Reset encoder" is a program for resetting motor encoder counts using an absolute encoder. When ratcheting occurs at gear reducers, a joint angle calculated from a motor encoder is differed from a correct value. "Reset encoder" calculate difference between the motor encoder and the absolute encoder, and move joint to correct angle. This program achieved on-line correcting encoder error.

2) Robot Body Control layer / Recognition Hardware layer: In this layer, we use hrpsys-base [4] for basic robot body level control. This robot body control programs contain sequencer for joint angles, joint protection plugin, self interference detection plugin, impedance controller, stabilizer for humanoids, autobalancer for keeping COG position, and so on. Cameras and LIDAR are controlled in this layer. They work as device drivers for ROS system, provide sensor data as ROS message to the upper layer.

3) Field Computer layer (application software): This layer is an application software layer for performing tasks. There are perception modules for each task, which use point clouds from a stereo camera and a LIDAR. The manipulation planner generates whole body motions using results from perception modules. The manipulation planner is based on Euslisp[5] [7] as a scripting language for rapid prototyping several motions.

4) Operator Control Station layer (application software): This layer is an application software layer for user interface and controlling of input devices. Visualization of robot state, images and recognition results are performed for users who send commands to FC layer for controlling robot.

From the point of view of software time scheduling, the first layer and the second layer are using real-time scheduling system. Real-time layers are asynchronously received commands from non real-time layer through ROS API.

Field Computer (FC) layer and Operator Control Station (OCS) layer are described by the terms from DRC Finals. FC layer is an application software level, which can communicate to the robot without bandwidth limitation and latency. OCS layer is an application software level, which communicate to FC level and the robot hardware with bandwidth limitation and latency.

A. Data Communication

For DRC Finals, there were two limited communication lines between FC and OCS. One was very narrow bandwidth, bit-rate was 9600 [bps], and it was bi-directional without latency. the other was relatively broad, bit-rate was 300 [Mbps], but it was uni-direction (from a robot to a operator) with from 1 to 29 second burst every 30 second.

We developed simple network protocol for streaming with UDP, because a hand shake procedure is not suitable for restricted communication. We built two communication paths. One is narrow path for narrow bandwidth communication. This path consists of a uni-directional line and a bi-directional line. On the uni-directional line, compressed

[1]https://github.com/jsk-ros-pkg
[2]https://github.com/start-jsk
[3]https://github.com/start-jsk/rtmros_common
[4]https://github.com/fkanehiro/hrpsys-base
[5]https://github.com/euslisp

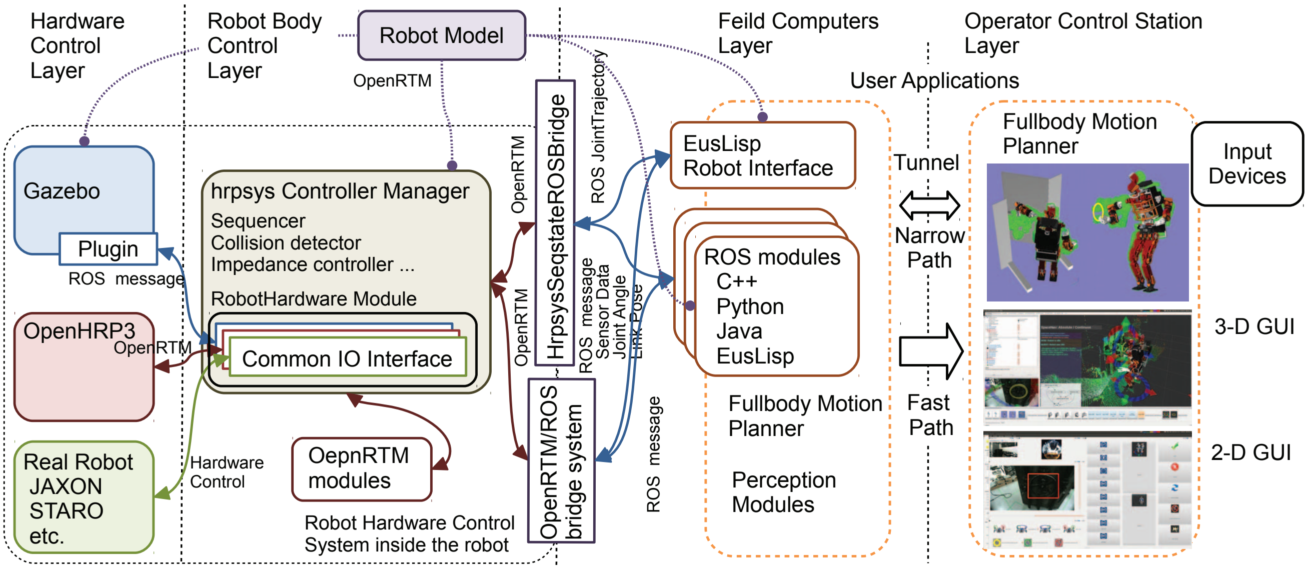

Fig. 8. Overall software system for DRC Finals

data such as joint angles, odometry and battery voltage were sent from FC side to OCS side. On the bi-directional line, OCS side sends commands to and receives results from FC side.

The other one is fast path. On this path, not compressed data was sent from FC to OCS as streaming. Data structure on this path is simple stacking raw serialized ROS messages. After receiving data on OCS, data was published as ROS messages on OCS site. There are problem for using single ROS network such as unable to control transferred data size, unable to compress, unavoidable hand shake. Then, different ROS networks were used on FC site and OCS site. Fast path was works like tunnel between different ROS networks. More detail descriptions about communication and data structure are in [8].

B. User Interface for DRC Finals

We progressed our robot tele-operation interface with the existing robots, because new robot hardware was developed around the same time. We aimed to develop tele-operation interface which can switch between autonomous robot behavior with operator's suggestion and direct operation by a operator. Our user interface consists of a 2-D GUI, a 3-D GUI and input devices. They are in OCS layer.

2-D GUI displays camera images from a stereo camera and a fish-eye camera. These image viewers are used to specify the region-of-interest for object perception. Operators can indicate the region-of-interest by drawing a rectangle on the viewer using a mouse. 2-D GUI also has set of buttons to specify the task context such as 'Valve task', 'Drill task' and 'Walk-To somewhere'. There are radio buttons to specify parameters for robot's motion, such as which arm to be used, where robot to stand.

3-D GUI is mainly used for operators confirming results of recognition and motion planning. Recognition results are visualized on 3-D GUI and they are super-imposed on stereo image viewer in 3-D GUI. It is difficult to send a motion sequence, which is a result of motion planning, from OCS layer to FC layer because of bandwidth limitation. So, we send only a target coordinates to the motion planner on FC layer. The same motion planner modules are on FC layer and OCS layer. Motion planning results on FC layer are sent to robot body control layer as a robot motion. Motion planning results on OCS layer are used to visualize the robot motion for operator's confirmation.

After operator's confirmation of the motion planning result, the motion sequence is sent to the robot to perform the motion. In case of unacceptable result of motion plan, the operator can correct motion by modifying the pose of recognized target, and moving standing location of the robot by 3-D interactive marker [9].

Operators can use several input devices, such as a gamepad, a 3d-mouse, a MIDI controller (button, rotary volume and liner control), as well as an ordinary mouse and keyboard. A 3-D mouse and a gamepad can be used to control 6-D pose of the target marker. A MIDI controller (BEHRINGER B-Control) is used to change parameters, such as time duration of motion, and radius of recognized results of the valve. These devices can improve operational efficiency of operators.

IV. Software Integration using Multi Robots

This project was started on July, 2014. Then we started to design new robot "JAXON", there was not enough time to prepare for DRC Finals using actual robot. ("JAXON" became to be able to perform tasks on March, 2015.)

We developed software system described at Section III using "HRP-2" [10] and "STARO" before building up "JAXON". "JAXON" and "STARO" were adopted almost the same software structure and the same motor drive system. "HRP-2" also have the same software structure, lower than Robot Control layer. Developed software system can be used for different robots because of using compatible robot models. In order to use different types of the robot model on the software system, a file conversion system between different robot models (Fig. 9) was adopted.

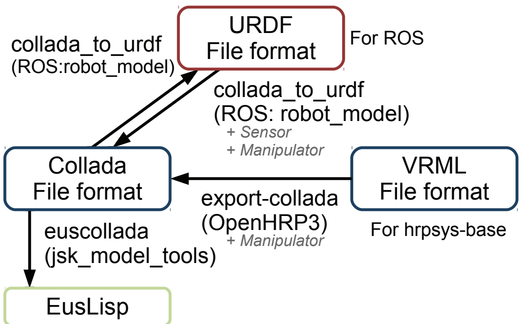

Fig. 9. Model conversion system for software integration

An evaluation platform through humanoid robot design to high level software application is required. Gazebo (robot dynamics simulator) [11][6] were adopted as an evaluation platform. Connecting Gazebo with Robot Control layer as described at Section III hold compatibility between a real robot and a simulator.

V. Future Works

JVRC[7] is a competition for the computer simulation of disaster response humanoid robots. JVRC is also a part of NEDO's project. This is a event similar to Virtual Robotics Challenge (VRC) described at [12].

For disaster response humanoid robots, many trial and error should be required in order to know what kind of tasks robots are desired to do, and to verify humanoid robots have capabilities to perform desired tasks. Learning through trial and error with real robots needs much resource. Especially, academic and industrial field for disaster response humanoid robots need human resources such as researchers and technical experts. Holding a robot competition using simulation is a nice idea, in order to encourage someone who is interested in disaster response humanoid robots to join the field and get an idea for robots performing in the actual tasks.

Results from DRC Finals and JVRC will be presented on International Robot Exhibition(IREX)[8]. We will demonstrate with actual humanoid robots to perform disaster response tasks, such as tasks in DRC Finals and in JVRC.

VI. Conclusions

Through participation and preparation for DRC Finals, we extends knowledge for disaster response humanoid robots such as robot hardware design methodology, how humanoid robot performs at disaster site, and user interfaces for disaster response humanoid robots.

We developed "JAXON" as a humanoid robot which can work practically for support activities on disaster sites. The methodology for hardware design is required in order to humanoid robot performing task while getting up even if falling down. Our robot can work after falling down in DRC Finals, but a reset procedure was needed to communicate the robot.

For methodology for designing software architecture, we used open source middle-ware in order to integrate rapidly new features and previous works, and be able to evaluate rapidly base system structure. Using middle-ware is easy to rabidly construct software system, but it needs computation resource and have less robustness. For practical realization after rapid evaluation of design, methods for removing unnecessary parts from middle-ware prototyping are needed in order to reduce using computation resource and increase robustness.

[6]http://gazebosim.org
[7]http://www.jvrc.org/en/index.html
[8]http://www.nikkan.co.jp/eve/irex/english/index.html

References

[1] Defense Advanced Resaerch Projects Agency, ”DARPA DRC”, 2015. [Online]. Available: www.theroboticschallenge.org

[2] Y. Ito, T. Nakaoka, J. Urata, K. Kobayashi, S. Nozawa, Y. Nakanishi, K. Okada, and M. Inaba, "Development and verification of life-size humanoid with high-output actuation system," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, may 2014, pp. 3433–3438.

[3] K. Kojima, T. Karasawa, T. Kozuki, E. Kuroiwa, S. Yukizaki, S. Iwaishi, T. Ishikawa, R. Koyama, S. Noda, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Development of life-sized high-power humanoid robot jaxon for real-world use," in *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots (Humanoids 2015)*.

[4] J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design of high torque and high speed leg module for high power humanoid," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 10 2010, pp. 4497–4502.

[5] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[6] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "RT-middleware: distributed component middleware for RT (robot technology)," 2005, pp. 3933–3938.

[7] T. Matsui and M. Inaba, "Euslisp: an object-based implementation of lisp," *Journal of Information Processing*, vol. 13, no. 3, pp. 327–338, 1990.

[8] R. Ueda, M. Murooka, Y. Ohara, I. Kumagai, R. Terasawa, Y. Furuta, K. Kojima, T. Karasawa, F. Sugai, S. Iwaishi, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Humanoid integrated ui system for supervised autonomy with massive data visualization over narrow and unreliable network communication for drc competition," in *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots (Humanoids 2015)*, November 2015.

[9] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, March 2012, pp. 1–8.

[10] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *Pro IEEE Int. Conference on Robotics and Automation*, 2004, pp. 1083–1090.

[11] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, 2004, pp. 2149–2154 vol.3.

[12] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. Mccrory, *et al.*, "Summary of team ihmc's virtual robotics challenge entry," in *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*. IEEE, 2013, pp. 307–314.