

Development of the Humanoid Disaster Response Platform DRC-HUBO+

Taejin Jung¹, Jeongsoo Lim, Hyoin Bae¹, Kang Kyu Lee, Hyun-Min Joe¹, and Jun-Ho Oh¹, *Member, IEEE*

Abstract—This paper describes a humanoid robotics platform (DRC-HUBO+) developed for the Defense Advanced Research Projects Agency Robotics Challenge (DRC) Finals. This paper also describes the design criteria, hardware, software framework, and experimental testing of the DRC-HUBO+ platform. The purpose of DRC-HUBO+ is to perform tasks by teleoperation in hazardous environments that are unsafe for humans, such as disaster zones. We identified specific design concepts for DRC-HUBO+ to achieve this goal. For a robot to be capable of performing human tasks, a human-like shape and size, autonomy, mobility, manipulability, and power are required, among other features. Furthermore, modularized joints and a user-friendly software framework were emphasized as design concepts to facilitate research on the robot tasks. The DRC-HUBO+ platform is based on DRC-HUBO-1 and HUBO-2. The torque of each joint is increased compared to that in DRC-HUBO-1 owing to its high reduction ratio and air-cooling system. DRC-HUBO+ is designed with an exoskeletal structure to provide it with sufficient stiffness relative to its mass. All wires are enclosed within the robot body using a hollow shaft and covers to protect the wires from external shock. Regarding the vision system, active cognition of the environment can be realized using a light-detection and ranging sensor and vision cameras on the head. To achieve stable mobility, the robot can transition from the bipedal walking mode to the wheel mode using wheels located on both knees. DRC-HUBO+ has 32 degrees of freedom (DOFs), including seven DOFs for each arm and six DOFs for each leg, and a solid and light body with a height of 170 cm and a mass of 80 kg. A software framework referred to as PODO, with a Linux kernel and the Xenomai patch, is used in DRC-HUBO+.

Index Terms—End effectors, embedded software, humanoid robots, legged locomotion, manipulators, mobile robots, operating systems, software systems.

I. INTRODUCTION

THE Fukushima nuclear disaster highlighted the requirement for research on robots that can accomplish tasks in disaster conditions. Moreover, the disaster motivated the Defense Advanced Research Projects Agency (DARPA) to further

Manuscript received May 4, 2017; revised October 2, 2017; accepted October 26, 2017. Date of current version February 5, 2018. This paper was recommended for publication by Associate Editor P. Fraisse and Editor I.-M. Chen upon evaluation of the reviewers' comments. (*Corresponding author: Jun Ho Oh.*)

The authors are with the Humanoid Research Center, School of Mechanical, Aerospace and Systems Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-338, South Korea (e-mail: jolaluxury@kaist.ac.kr; yjs0497@kaist.ac.kr; pos97110@kaist.ac.kr; guecom@kaist.ac.kr; kharisma34@kaist.ac.kr; jhoh@kaist.ac.kr).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2017.2776287

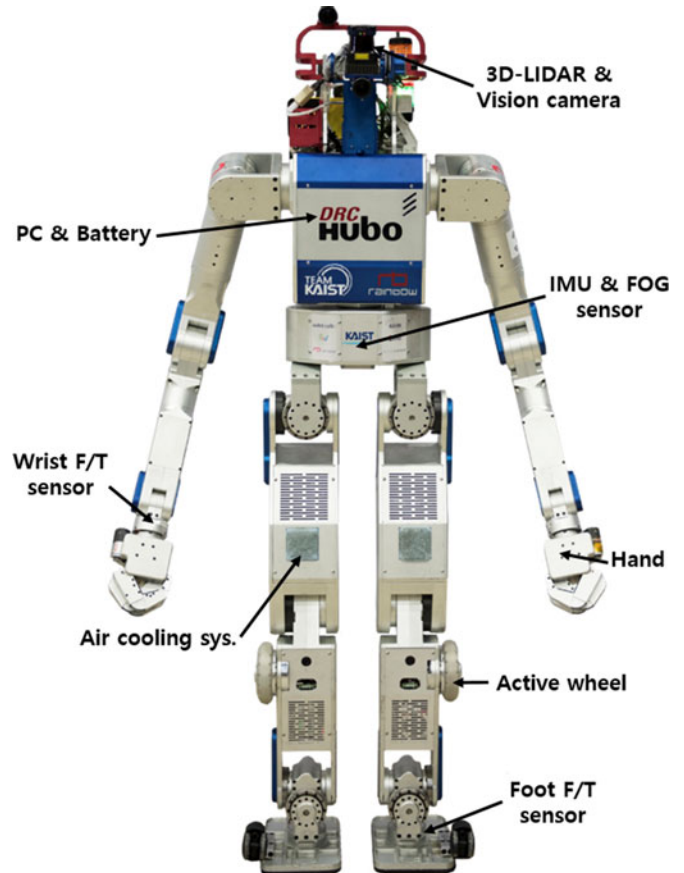


Fig. 1. DRC-HUBO+.

investigate the use of robots in such situations. They analyzed the Fukushima disaster and attempted to simulate the disaster environment and conditions. One outcome of their research was a robotic competition referred to as the DARPA Robotics Challenge (DRC), in which 25 international teams participated with various robot platforms. The DRC Finals included eight tasks required in actual disaster situations [1], i.e., “Driving,” “Egress,” “Door,” “Valve,” “Wall,” “Surprise,” “Rubble,” and “Stairs.” To participate in the DRC Finals, we, Team KAIST, developed DRC-HUBO+ (shown in Fig. 1), which is a humanoid-type robot based on DRC-HUBO-1.

To complete the mission tasks, a robot platform must be equipped with mobility to move to the worksite and manipulability to perform required functions. Humanoids are prime examples of robots that possess these two characteristics. These robots can accurately replicate almost every degree of human

motion. They can walk to assigned positions and use human-like arms to perform assigned tasks. There are various humanoid robots. Atlas [2] [Boston Dynamics, with a height of 188 cm, a mass of 150 kg, and 28 degrees of freedom (DOFs)] is a representative humanoid robot that is driven by hydraulic actuators. The torque and speed of Atlas surpass those of other robots that use electric driven motors. Most humanoid robots except Atlas are designed with electrically driven joints. The electrically actuated humanoid robots that participated in the DRC Finals included HRP-2Kai [3] (height: 171 cm, mass: 65 kg), THOR [4] (Virginia Tech, 34 DOFs), Valkyrie [5] (NASA-JSC, height: 187 cm, mass: 125 kg), and JAXON [6] (NEDO in Japan, height: 188 cm, mass: 127 kg). These humanoid robots can traverse even terrain and overcome obstacles such as ladders and stairs. However, they face stability problems when walking because even slight unsuspected incorrect sensor information or a minor disturbance from the external environment can result in falling down.

In addition to humanoid robots, wheel-based robots have the aforementioned mobility and manipulability. Nimbro Rescue [7], RoboSimian [8] (NASA-JPL), and CHIMP [9] are the wheel-based robots that participated in the DRC Finals. These wheel-based robots demonstrated stable and high-mobility performance on even terrain, such as that in office environments. However, the mobility of such robots is limited when faced with stairs, ladders, or extremely uneven ground.

The previous versions of DRC-HUBO+ are HUBO-2 [10] (height: 125 cm, mass: 45 kg) and DRC-HUBO-1 (height: 155 cm, mass: 60 kg), which are electrically driven humanoid robots. As the design purpose of HUBO-2 is to study biped walking and human-like motion generation, it is too small and weak to perform actual human work. DRC-HUBO-1 was developed to overcome this problem by modifying HUBO-2. Compared to HUBO-2, DRC-HUBO-1 has longer arms and legs, increased motor capacity, and greater harmonic drive for the arms. However, except for the arm joint modules, all joint modules of HUBO-2 were reused, and the overall force and stiffness proved insufficient to perform human work.

The above-mentioned robots have advantages and disadvantages. Humanoid and wheel-based robots have problems with walking stability or crossing obstacles, whereas the joint torques of lightweight and small robots are insufficient to perform certain tasks. To overcome these problems, DRC-HUBO+ was designed to achieve enhanced mobility and manipulability compared with those of other robots. In addition to mobility and manipulability, sufficiently high joint torques for difficult tasks, system robustness, light and rigid body design, modularized joint design, a three-dimensional (3-D) light detection and ranging (LIDAR) system for autonomy, and a real-time software framework suitable for motion development were design objectives. We combined the strengths of humanoid and wheel-based robots to enhance mobility. Specifically, DRC-HUBO+ can overcome obstacles such as stairs and rough terrain through biped walking, while the wheel mode ensures stable locomotion even if the robot moves across inclined ground or if high external force is applied to it. New arms and hands were designed to enhance its manipulability. An air-cooling method was employed to achieve

TABLE I
SPECIFICATION OF DRC-HUBO+

Height		170 cm
Weight	Inc. batteries	80 kg
	Exc. batteries	72 kg
DOF	Total	32 DOFs
	Head	1 DOF
	Arm	2 Arms \times 7 DOFs
	Hand	2 Arms \times 1 DOF
	Waist	1 DOF
	Leg	2 Legs \times 6 DOFs
	Wheel	2 Wheels \times 1 DOFs
Payload	Arm	10 kg
	Hand (fingertip grip)	2 kg
	Hand (encompassing grip)	20 kg
Max walking speed		1.8 km/h
Walking cycle		0.8 s/step
Max speed (wheel mode)		3.5 km/h

sufficient joint torque. Finally, we included hollow shafts and covers to protect the electronic parts from external shocks and malfunctions and thereby improve the robustness of the system. In addition, we developed the PODO software framework for DRC-HUBO+. PODO allows for real-time control of the robot platform and easy integration of the routines from multiple developers.

This paper describes the DRC-HUBO+ platform development and test results in detail. Even though the previous paper [11] provides a brief description of the platform and the specific strategy and software for each task, it does not describe the platform development process and design criteria in detail, which is the focus of this paper. The remainder of this paper is organized as follows. Section II presents the hardware design. Section III discusses the software architecture. Section IV describes the performance verification experiments for the hardware and software, and finally Section V summarizes the performance of the robot in the eight competition tasks.

II. HARDWARE DESIGN

A. DRC-HUBO+ Specifications

The configuration and size of a robot are the primary aspects that must be determined. As tools and infrastructure are designed according to human specifications, we designed our robot as a humanoid with a configuration and size similar to those of an adult man. All joints of DRC-HUBO+ consist of a harmonic drive and one or two motors, except for the hands and head. Its structure is similar to that of HUBO-2 [10] and Schaft [12], [13], i.e., the motors rotate the wave generator of the harmonic drive via timing belts and pulleys. Table I shows the basic specifications of DRC-HUBO+. As indicated in Fig. 2, DRC-HUBO+ is 170 cm high and has a mass of 80 kg, including two batteries. DRC-HUBO+ has 32 DOFs, including seven DOFs for each arm for kinematic redundancy, one DOF for each hand, six DOFs for each leg with one DOF for the waist to allow for human-like walking, one DOF for the head to allow for the perception of the surrounding environment, and

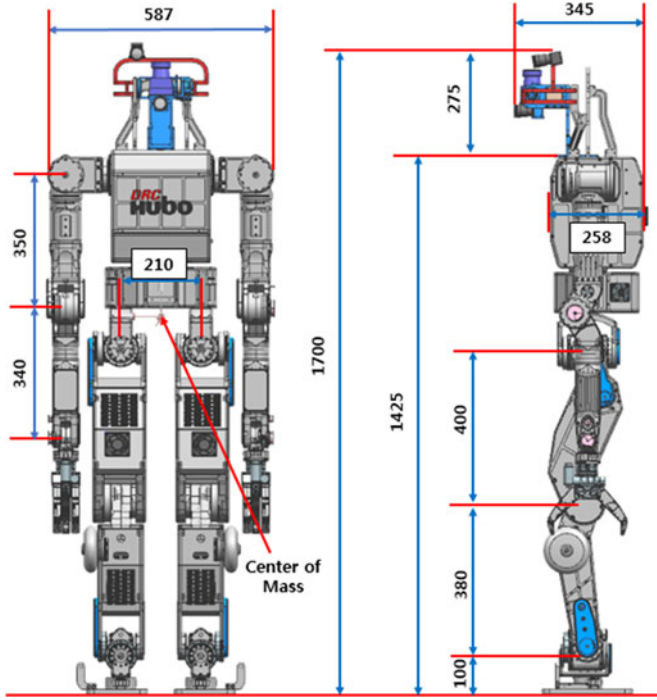


Fig. 2. Dimensions of DRC-HUBO+.

two DOFs for active wheels for wheel-based movement. The upper and lower sections of each leg have lengths of 400 mm and 380 mm, respectively, and the mass of each leg is 14 kg. The upper and lower sections of each arm are 350 mm and 340 mm long, respectively. The payload of each arm is 10 kg, and the payload of each hand is 2 kg for fingertip grip and 20 kg for encompassing grip. These features enable DRC-HUBO+ to perform challenging tasks. Furthermore, several sensors are installed to recognize the robot's status. A force/torque (F/T) sensor is located at the end of each limb, and an inertia measurement unit (IMU) and a fiber-optic gyro (FOG) are attached to its pelvis. Two optical flow sensors are attached to the shins. A camera and a 2-D LIDAR sensor are mounted to the head. In addition, DRC-HUBO+ contains two Ni-ion batteries (dc voltage: 4 V, capacity: 11.4 Ah, mass: 4 kg), which allow for normal operation for up to 4 h.

B. Robust and Lightweight Robot Hardware Platform

We designed the hardware platform to be lightweight and robust. To achieve sufficient stiffness relative to its mass, DRC-HUBO+ was designed with an exoskeletal structure, and to reduce the mass of the motors and harmonic drives, we used the smallest motors and harmonic drives for each joint that could satisfy the joint torque and speed requirements for the eight tasks. Moreover, because having sufficient joint torque is more important than the joint speed for performing tasks, high gear ratios were used to enable even small motors to generate more torque.

The electronic parts of a robot system must be protected from external shocks and malfunctions to enhance its robustness. Hollow shafts and wire protectors were included in DRC-HUBO+

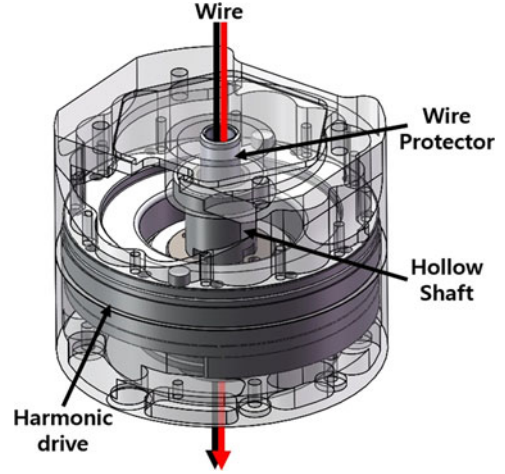


Fig. 3. Wiring by using hollow shaft protect the wire from the external shock and increase the joint limit up to $\pm 180^\circ$.

to achieve this objective. As shown in Fig. 3, a hollow shaft enables wires to pass through the middle of the shaft. This routing scheme not only decreases exposure of the electronic parts, but also increases the ranges of joint motion. Moreover, DRC-HUBO+ was designed to have minimal position error from assembly error and sagging. Assembly error, which typically results from hole tolerance, was minimized by manufacturing all parts to fit together precisely, with narrow tolerance. To prevent misalignment of the harmonic drive and frames, alignment pins were used instead of bolts in the two or three bolt holes. Sagging was minimized by strengthening areas subject to larger forces by performing stress simulations. In addition, because a cantilever is weak when subjected to a moment, the overall stiffness was increased by including support at both ends. Furthermore, as shown in Fig. 4, each component is modularized to facilitate repair. Each modularized component that embeds motors and control boards can be easily assembled using connection links.

C. Arm Design for Manipulability Enhancement

Each arm of DRC-HUBO+ was designed to have seven DOFs for redundancy, and arm length was increased compared to that of DRC-HUBO-1 to increase the workspace. Table II summarizes the ranges of motion of the upper joints of DRC-HUBO-1 and DRC-HUBO+. These ranges were optimized for the robot workspace by considering the physical limits of each joint. Moreover, the ranges of motion were designed to be more advantageous than the corresponding ranges for humans to facilitate tasks such as closing a valve and driving a vehicle. For tasks that require the robot to rotate its joints several times, the wrist yaw joint of the robot was enabled to rotate infinitely using a slip ring. This feature improves stability and time efficiency because it is possible to rotate several times at once instead of turning repeatedly. Even though a human can easily use the method of turning and holding, this method is difficult for robots because they have difficulty grasping objects accurately.

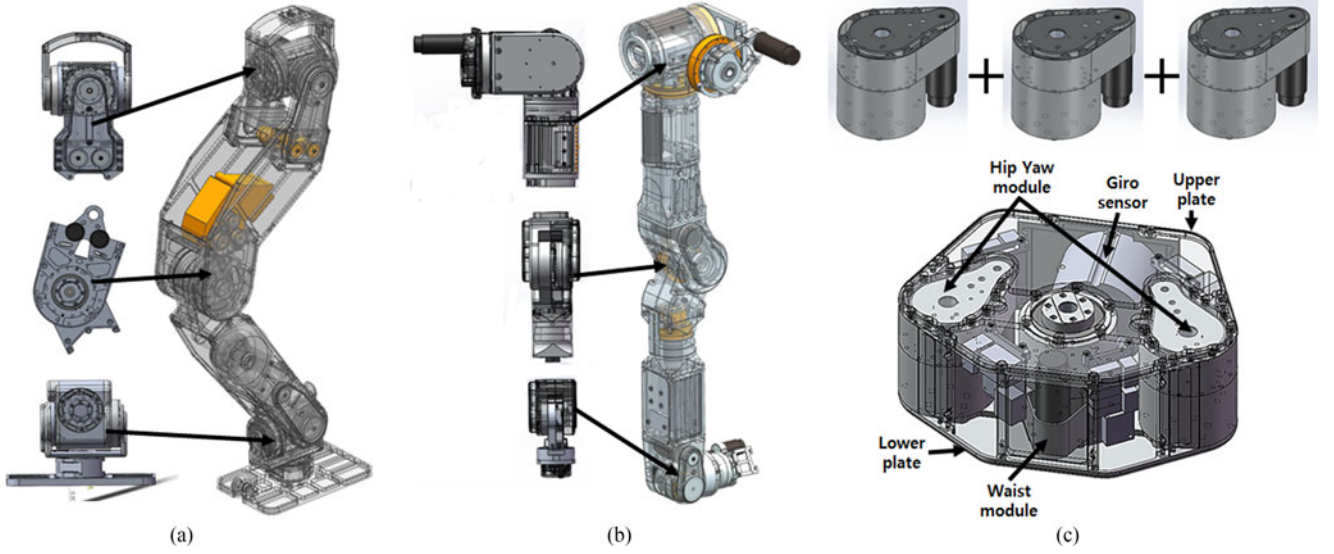


Fig. 4. Modularized DRC-HUBO+ parts. (a) DRC-HUBO+ leg. (b) DRC-HUBO+ arm. (c) DRC-HUBO+ pelvis.

TABLE II
UPPER BODY JOINT LIMIT OF DRC-HUBO AND DRC-HUBO+

Joint		Angle range (deg)	
		DRC-HUBO-1	DRC-HUBO+
Waist		-160-160	-350-350
Shoulder	Roll	-30-210	-30-210
	Pitch	-160-160	-240-80
	Yaw	-160-160	-160-160
Elbow	Pitch	-170-0	-170-30
Wrist	Yaw1	-160-160	-170-170
	Pitch	-160-160	-90-90
	Yaw2	-160-160	-∞-∞

D. Waist Joint Design for Manipulability and Mobility Enhancement

We introduced a specifically designed stopper to allow for the rotation of the waist joint up to $\pm 350^\circ$ so that the robot could perform work regardless of direction, as shown in Fig. 5. This design enables the robot to perform work by rotating its waist without stepping forward or backward. In addition to contributing to manipulability, this ability enhances the mobility of the robot. This feature enables the robot to move freely in either direction. For example, during a staircase or terrain task, contact between the knees and stairs or bricks can be prevented by moving the lower body backward.

E. Hand Design for Manipulability Enhancement

We developed a robot hand design suitable for DRC-HUBO+. The Adaptive Gripper produced by ROBOTIQ Inc., which has been used by several other teams, has strong grasp power and the ability to grasp variable objects. However, because it is too heavy and large to attach to DRC-HUBO+, we developed a new and lighter hand design that is more



Fig. 5. Mechanism of specifically designed stopper in the waist joint.

compatible with DRC-HUBO+. To perform the required tasks, we identified the following necessary features for the new hands.

- 1) Ability to grasp various objects.
- 2) Light weight.
- 3) Strong gripping force.

To reduce hand mass, one motor was assigned to three fingers instead of assigning a separate motor for each finger. As illustrated in Fig. 6, one active module and two passive modules were assembled, and gears were used to move the three fingers simultaneously.

We used two four-bar linkages to enable the robot to grab objects of various shapes and sizes, as shown in Fig. 6. Note that this mechanism is extensively discussed in [14]. The advantages of this type of gripper are that it enables the robot to have a shape-adaptive grasp and to grip thin objects by keeping the distal phalanges of the two facing fingers parallel. The red four-bar linkage serves to drive the finger, and the blue parallel four-bar linkage holds the distal phalanx straight using mechanical limits and a spring until the object is in contact with the finger at the proximal phalanx. A shape-adaptive grip is performed after contact. However, under-actuated fingers that use wires or four-bar linkages have disadvantages. Possibly the most significant drawback is that an ejection phenomenon can occur. To prevent ejection, it is necessary to design the geometric parameters of the four-bar linkage analytically. We adopted the results

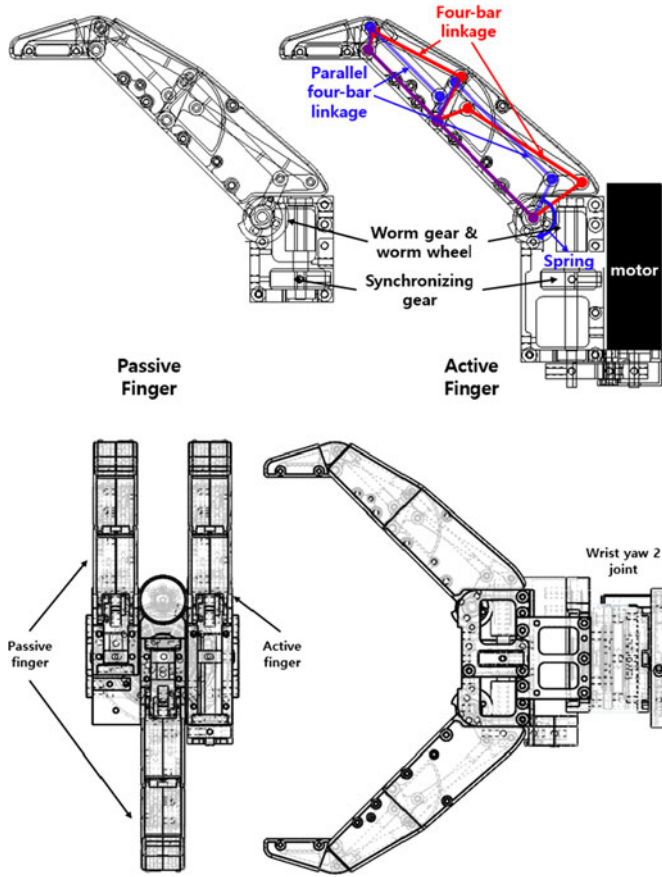


Fig. 6. Modularized DRC-HUBO+ hand.

of the under-actuated finger research performed by Birglen and Gosselin to eliminate ejection [15].

For some tasks such as climbing a ladder or lifting a heavy object, it is necessary for a robot to have a very strong grasping force. When climbing a ladder, the robot hands require considerable power to support the mass of the robot. However, a strong motor with sufficient power for this task is typically too large and heavy to be used for the hands. Therefore, a worm gear set was applied to a small motor to make the new hands nonback-drivable unless the user commands them to release. Therefore, because the hands are not required to apply grasping force continuously, they can use considerably higher torque than the rated torque of the motor. The new hand design was developed to satisfy the requirements mentioned previously. It consists of three shape adaptive fingers and a small mass of 460 g and allows for wrap grasps with holding forces in excess of 100 N and pinch grasps with forces in excess of 20 N.

F. Dual-Locomotion Mode Strategy for Mobility Enhancement

Robot walking stability and robustness are not ensured to be sufficient. Even if a minor problem arises, the robot can fall down, causing substantial damage to the robot system. To overcome such problems, we present a dual-mode strategy to extend the mobility of humanoid robots by including the walking mode and wheel mode. The technique enables robots to climb

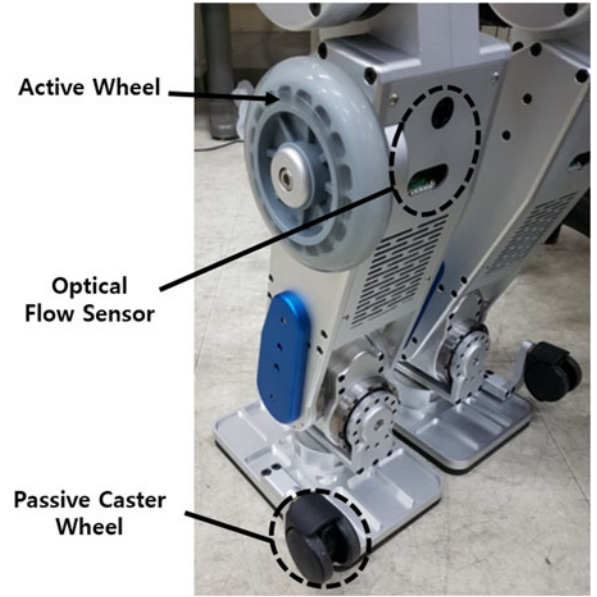


Fig. 7. Wheel attachment construction. A forward active wheel is attached on the calf of each leg, and a passive wheel is attached to the corresponding foot.

stairs and ladders, traverse extremely uneven terrain by walking, and move across even terrain more rapidly and stably using the wheel mode.

This dual-mode strategy includes the processes of changing from the walking mode to the wheel mode and vice versa. As shown in Fig. 7, an active wheel is attached to the side surface of the calf of the leg. When the robot is in a standing state (walking mode), its wheels are on the sides of the legs. However, after the transition, these wheels are in contact with the ground. As illustrated in Fig. 8, the active wheel system includes a 200 W brushless DC (BLDC) motor, a harmonic reduction gear, and urethane wheels. Using a harmonic drive with a ratio of 100 to 1 at a maximum motor speed of 16 000 r/min, the robot achieves a maximum speed of 3.5 km/h in the wheel mode, which is almost twice as fast as its walking speed of 1.8 km/h. A passive caster wheel is included on each foot to enable the robot to rotate in tight spaces (radius of rotation = 0 cm). To maximize the supporting plane of the wheel mode, each wheel is attached to the side of the corresponding foot, as depicted in Fig. 7.

The center of mass (COM) and support polygon of each mode are illustrated in Fig. 9. As shown in the lower part of the figure, the support polygon of the wheel mode is larger than that of the walking mode (support polygon sizes: walking mode = 660 cm², wheel mode = 980 cm²). Consequently, the robot stability during manipulation can be increased using the wheel mode. In addition, a lower COM height (COM height: walking mode = 780 mm, wheel mode = 480 mm) significantly enhances robot stability.

In the walking mode, the balance of the robot is maintained by the zero-moment point (ZMP) controller. However, in the wheel mode, it is possible to move stably without the ZMP controller since the support polygon is much wider than the walking mode. In actual calculations, the designed robot can

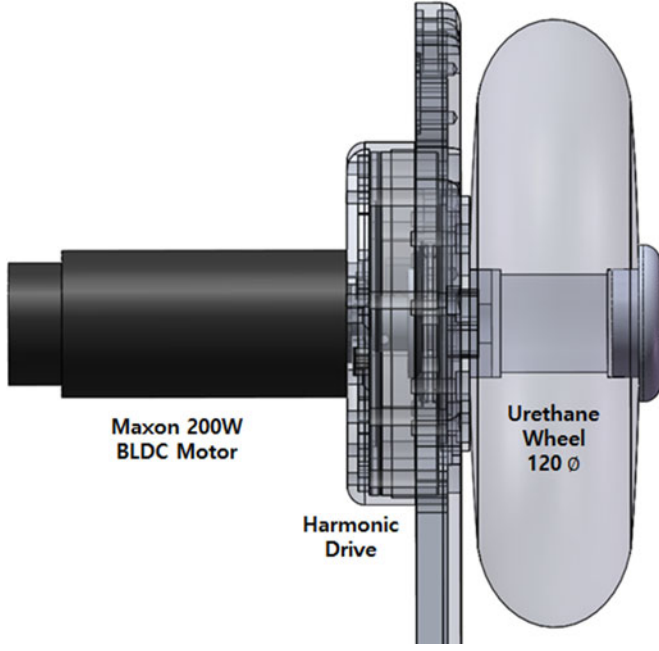


Fig. 8. Active wheel structure. It consists of a motor, a harmonic drive, and urethane wheels.

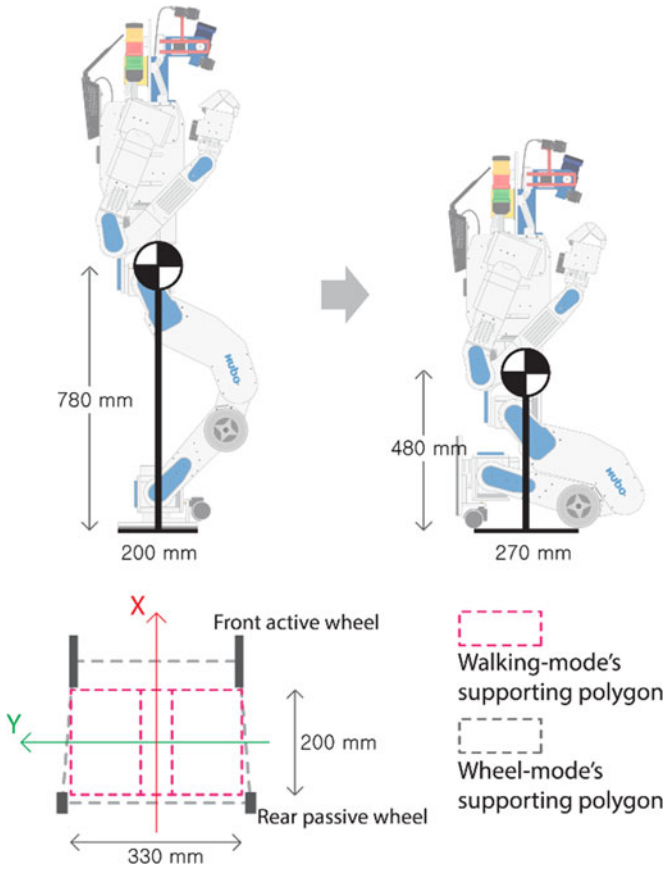


Fig. 9. Support polygons and COM comparison between walking mode and wheel mode. The stability is increased during wheel mode as compared to that during walking mode, because of the wide support polygon and low COM.

fall when its acceleration is more than 2.75 m/s^2 or 3.37 m/s^2 during straight or rotational motion, respectively. The maximum acceleration of the robot with a mass of 80 kg was 1.81 m/s^2 in straight motion and 1.64 m/s^2 in rotary motion owing to the rated torque of the motor, which confirms the robot to be sufficiently stable without ZMP control. Therefore, in the wheel mode, the controller for the balance is not used separately.

The overall transition process between the walking mode and wheel mode consists of three steps, as illustrated in Fig. 10. The transition from the wheel mode to the walking mode is implemented in reverse order. In the DRC Finals, the time required for completion was used as a tiebreaker between teams; thus, rapid switching between the walking mode and the wheel mode was important. Therefore, the motion that satisfied the ZMP condition in the minimum time was generated using the optimization process described in [16]. The optimization process uses the optimization method used in [17] to generate the mode change joint motion satisfying the ZMP condition in the minimum transition time. By using this optimization process, we could shorten the mode switching time compared to the static motion in which the mass center of the robot is located at the center of the two feet. The actual experimental results are presented in Section IV.

In addition, several problems that may occur in the wheel mode can be solved. For example, if the ground is uneven, the robot can become unstable if not all four wheels touch the ground simultaneously. The low gain position control with gain override on the knee and hip joints improves the stability of the robot by enabling all four wheels to touch the ground simultaneously, even when it is not flat.

Moreover, when the robot moves in the wheel mode, its exact posture is difficult to estimate owing to slippage according to ground conditions. To solve this problem, an IMU and a FOG are used on the pelvis of DRC-HUBO+ and optical flow sensors are used on the shins. The first sensor consists of the IMU sensor, ADIS-16480, which includes an accelerometer that measures the absolute posture, and a FOG, KVH-1750, which measures angular velocity. In addition, an optical flow sensor, ADNS-3080, is attached below each knee joint, as shown in Fig. 7. These sensors are directed vertically toward the ground and measure the linear velocity along the corresponding calf. The actual velocity of DRC-HUBO+ can be measured using these sensors. The linear velocity is obtained as the average value of the two optical flow sensors, and the angular velocity is obtained from the gyro and IMU. The desired velocity of each wheel is V_d , and V_r is the actual velocity, which is obtained from the gyro and optical flow sensors. Then, the final input velocity V_{input} of each wheel can then be calculated using

$$V_{input} = V_d + \left(K_p e_{vel} + K_i \int e_{vel} dt + K_d \frac{d}{dt} e_{vel} \right) \quad (1)$$

where K_p , K_d , and K_i are PID control gains, and e_{vel} is the velocity error ($e_{vel} = V_d - V_r$). A simple block diagram of the wheel velocity compensation process is presented in Fig. 11.

A stable walking pattern and controller are required to increase the stability of a position-controlled humanoid robot in

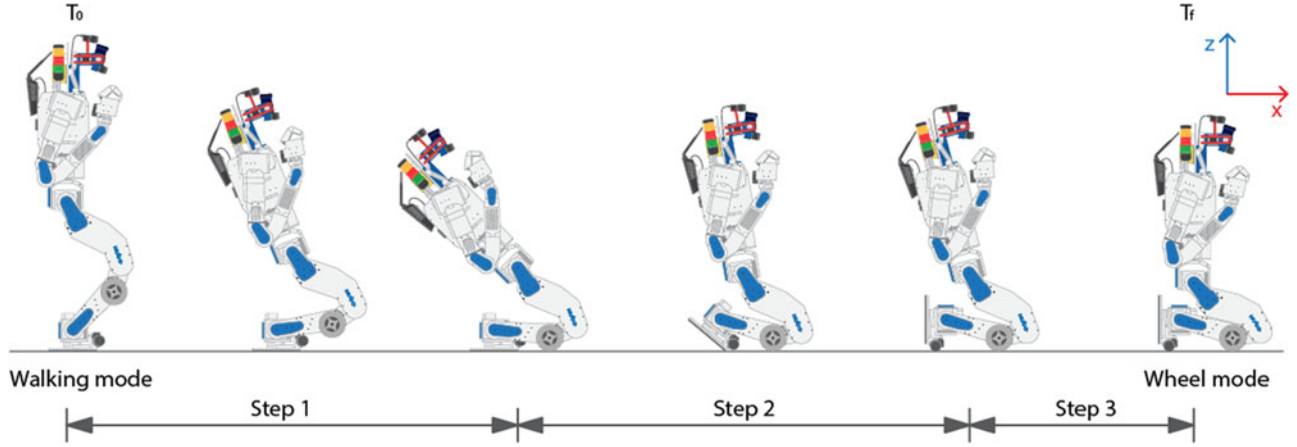


Fig. 10. Posture transition between walking mode and wheel mode. The process consists of three steps: 1) kneeling, 2) kick-off from the ankle, and 3) gain overriding.

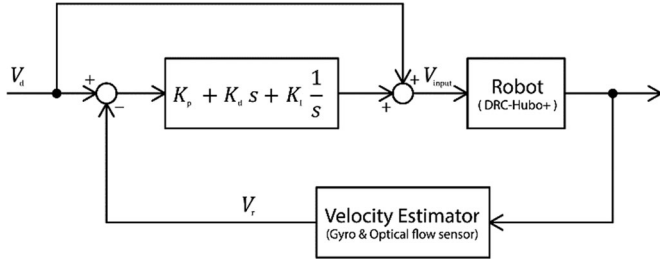


Fig. 11. Wheel velocity compensation. The real velocity is estimated by the gyro and optical flow sensors.

the walking mode. Preview control is utilized to generate the stable reference trajectory of the COM [18]. A linear inverted pendulum model of the robot is used in preview control. There is model error between the robot and model, in addition to disturbances from the ground. These factors make it difficult for the robot to follow the walking pattern.

Controllers are employed to stabilize the robot despite model error and disturbances. Specifically, we selected the previously developed ZMP [19] and upright posture [20] controllers. The ZMP controller generates the COM position control input to track the reference ZMP. The COM position, which is calculated by the ZMP controller, is added to the reference COM trajectory of the walking pattern in real time. To track the desired orientation of the reference frame, the upright posture controller generates the ankle joint angle control input. This input is added to the ankle reference joint angle. Section IV describes the walking experiment in which the walking algorithm described previously was implemented and the mode change experiment performed to minimize the required time.

G. Design for Power Enhancement

When a robot performs a difficult task such as removing debris, climbing a ladder, running, or jumping, its joints, including the knees, hips, and arm joints, may require more torque. As the torque of a hydraulic actuator is higher than that of an electric motor of the same size, robots that use hydraulic systems, such as Atlas, are relatively free from insufficient joint torque

problems. On the contrary, robots that use electric motors, such as DRC-HUBO+, encounter problems owing to insufficient joint torque.

These issues can be resolved using several motors simultaneously for each joint; however, this solution is not feasible for all joints because of mass and design constraints. Thus, improving motor efficiency is a potential solution. A typical method of improving motor efficiency is cooling the motor and preventing temperature from rising. A motor that produces a torque higher than the specified torque will have higher temperature, which lowers torque efficiency and can even damage the motor. A cooling device enables the motor to continuously produce a torque higher than that specified.

The Japanese robot Schaft [12], [13] provides improved motor efficiency using a water-cooling device. This is an extremely effective method of enhancing the performance of the electric motor. However, the water-cooling method has certain problems when it is applied to DRC-HUBO+. Critical problems could arise if water leakage were to occur, and the water-cooling circuit is too complex and bulky to pass through the inner body. Therefore, we used an air-cooling system instead of a water-cooling system. In particular, because cooling the motor surface directly using a fan is associated with lower cooling efficiency, an air-cooling radiation pin was attached to each joint subject to high torque, as shown in Fig. 12(a). Fig. 12(b) shows that a design that allows for direct heat dissipation by the frame was applied to the arms and other areas where fans could not be attached easily. The frame serves as a heat sink in this method. Thermal grease and a clamp are used to enhance the heat conductivity from the motor to the frame. DRC-HUBO+ can generate higher power through this simple but efficient air-cooling system. The heat dissipation efficiencies of the different methods are compared in Section IV.

H. Sensors Used to Perceive the Surrounding Environment or Its State

When a robot works independently in the field, the operator must be aware of its surroundings. Two types of information are required for the awareness of the surroundings, i.e., visual

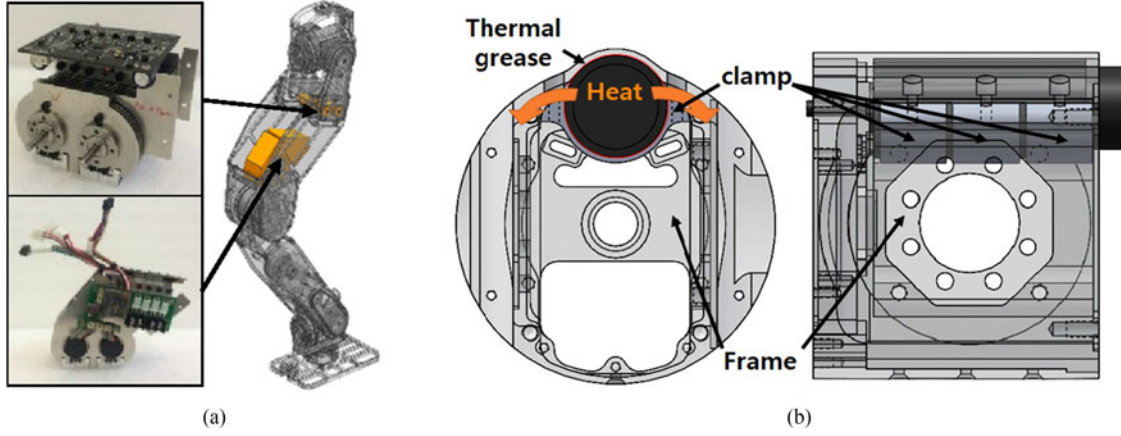


Fig. 12. Air-cooling system of DRC = HUBO+. (a) Air-cooling system using radiation pin. (b) Air-cooling system using direct heat dissipation by the frame.

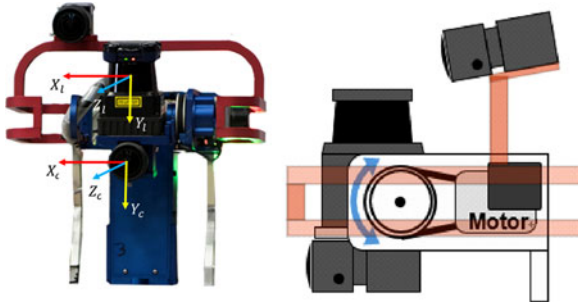


Fig. 13. Sensor system configuration for DRC-HUBO+ and its movement example. DRC-HUBO+ has two cameras and one 2-D LIDAR sensor installed on its head to obtain color images and depth information.

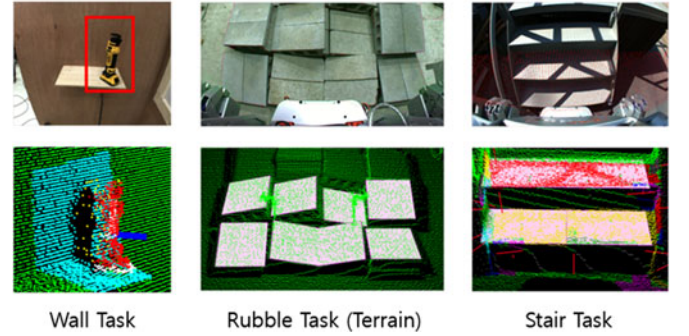


Fig. 14. Camera image and the 3-D point cloud data from the LIDAR and motor system.

information and 3-D mapping information. A camera (Point Grey Flea3-GigE with a 2.2 mm lens, 1280×960 pixels) is used to obtain visual information about the surroundings. To derive 3-D information based on the distance to the head of the robot, we constructed a 3-D LIDAR system using a 2-D LIDAR sensor (Hokuyo Automatic Co., Ltd.: UTM-30LX-EW, a scanning angle of 180° with an angular resolution of 0.25°) while synchronously controlling the head joint to move up and down.

As shown in Fig. 13, the main camera and 2-D LIDAR sensor are mounted on the same rig; this enables the robot to acquire 3-D point data and visual information by performing rotation using only a single motor. For the driving task, the streaming camera (Point Grey Flea3-GigE with a 2.2 mm lens, 1280×960 pixels) is placed on the head guard (red bar). Placing the streaming camera in an elevated position enables the robot to have a broad visual range, which is helpful for the operator when controlling the robot by steering the handle.

Fig. 14 shows the camera images and 3-D point cloud data obtained by the LIDAR system. For the wall task, a user points to the drill on the screen, and the robot automatically takes the drill based on the calculated 3-D position. For the rubble and stair tasks, the robot can recognize the ground conditions and generate walking patterns suitable for the recognized conditions to move accordingly.

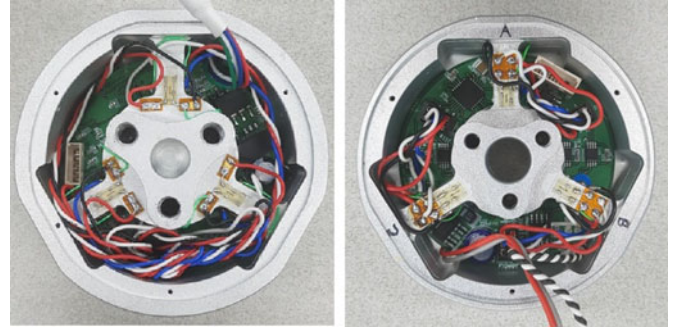


Fig. 15. Foot F/T sensor (left) and wrist F/T sensor (right).

In addition, an F/T sensor is required to measure the force between the robot and external environment when the robot is walking or working with its arms. We installed self-developed six-axis F/T sensors at the ends of the arms and legs for this purpose. Each sensor is comprised of a three-beam Maltese-cross-type structure, as shown in Fig. 15. Each foot F/T sensor can measure three-axis moments (M_x , M_y , and M_z) of up to $80 \text{ N}\cdot\text{m}$ and three axial forces (F_x , F_y , and F_z) of up to 2000 N , considering the mass of the robot. The wrist F/T sensors are designed to measure smaller loads, and each sensor can measure moments and forces of $20 \text{ N}\cdot\text{m}$ and 400 N , respectively. The structures of the wrist F/T sensors are similar to those of

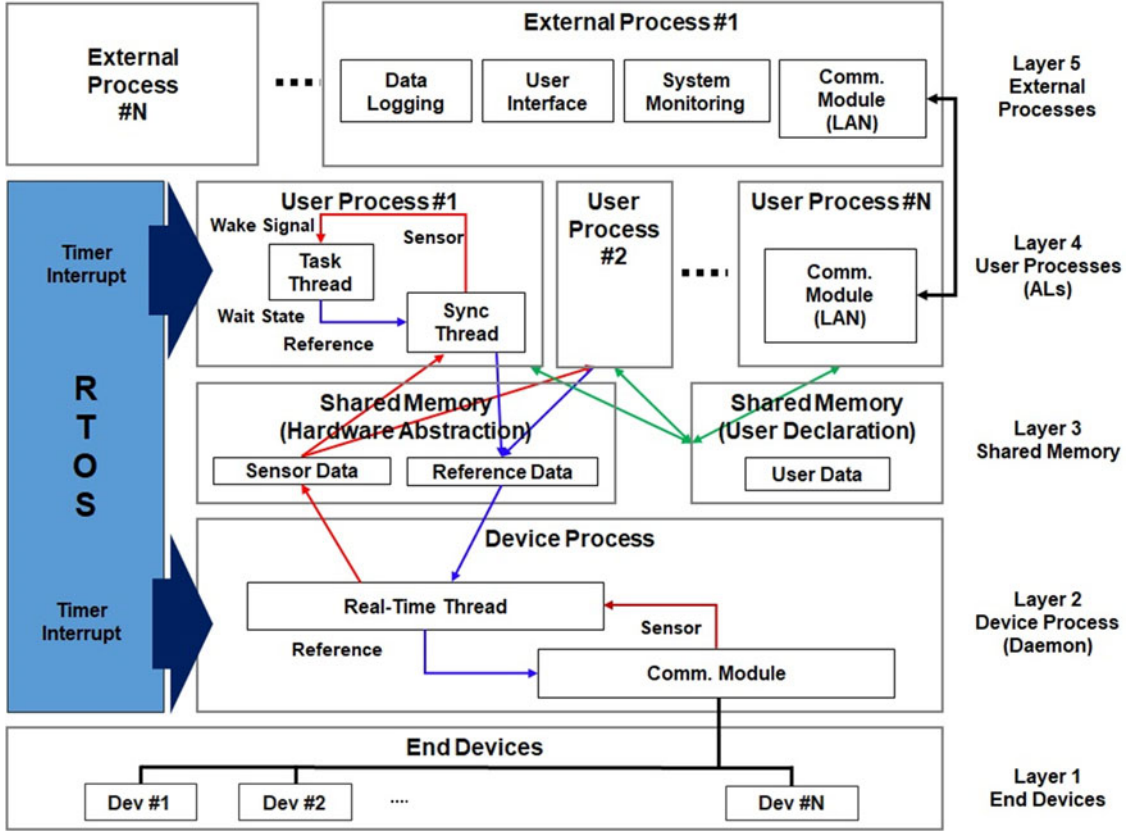


Fig. 16. Logical diagram of the architecture of the PODO software. Through the shared memory, the various processes can exchange their information. One special process (Daemon) has a real-time thread; it is the only process that can access end devices directly. The other processes (ALs) generate joint references and send those references to Daemon using the sensor data from the shared memory.

the foot F/T sensors, except that each sensors contains a hole in the middle for the hollow shaft. To increase the signal-to-noise ratio, we used an SS-080-050-345PB-S4 semiconductor gauge from Micron Instruments with a high gauge factor instead of the commonly used foil strain gauge. Six half-bridge circuits were constructed to reduce the negative effects due to temperature changes. The built-in board of each six-axis F/T sensor includes seven analog-to-digital converters (ADCs), analog filters, analog amplifiers, and three-axis digital acceleration sensors (only foot F/T sensor). The seven ADCs capture six sensor values and a board temperature measurement.

III. SOFTWARE FRAMEWORK DEVELOPMENT

PODO is the software framework that was developed for the DRC Trials and Finals [1]. It was designed considering the following four goals: Hard real-time functionality, usability in collaborative development, lightness and speed, and expandability. Among these four purposes, the essential goals for the DRC Finals are hard real-time functionality and usability in collaborative development. Hard real-time functionality is the most important feature for such robots, which demand fast reactivity and low latency. For the DRC Trials and Finals, only two robots are assigned to several developers that are responsible for the eight tasks. Therefore, robotic software must enable multiple researchers to codevelop and cooperate robots simultaneously. The previous paper [11] provides a brief description of PODO

and the overall system configuration; it does not describe the hierarchical architecture and practical testing of PODO in detail. In this section, we focus on the hard real-time functionality and usability in collaborative development, which are directly required for the DRC Finals, and explain how these two goals are achieved through the overall structure explanation of PODO. Then, we provide more details on how hard real-time functionality is improved over previous versions of PODO.

A. Hierarchical Architecture of PODO for Usability in Collaborative Development

PODO is a five-layered hierarchical software architecture. The adjacent layers can communicate with each other, and the total system can be stabilized through this managed system. Fig. 16 shows the overall architecture of PODO. The first layer comprises the robot simulator and end devices; the targets of control, e.g., motor controllers and sensor boards, exist in this layer. In second layer, a special process referred to as Daemon is the only process that communicates with end devices. In addition, only Daemon contains a real-time thread for controlling the robot hardware and simulator. We utilized Xenomai [21] on a Linux system (Ubuntu 12.04) to realize a real-time thread. The third layer is shared memory, which is an interprocess communication method. It is fast, easy to use, and does not increase in complexity, regardless of the number of connected processes [22]. We designed the shared memory to contain two types of

data, i.e., hardware abstraction data and user declaration data. Joint references and sensor data belong to hardware abstraction data. The command sets and any other parameters defined by a user belong to the user declaration data of the shared memory. The fourth layer consists of user processes, which are referred to as ALs. ALs are independent of each other, and they typically read sensor data from the shared memory and generate joint references. Then, they update the references to the shared memory. Not every AL is required to generate a motion reference. It can be used as a communication channel for the fifth layer or for debugging and safety. The fifth layer is external processes, and several different types of software can be used in this layer. In general, the GUI designed by a user can be utilized in the fifth layer to provide a command and to observe the system easily.

The key components that create environments that multiple users can develop simultaneously are ALs, shared memory, and Daemon. There is a motion owner for each joint, and only the ALs with the motion owner for that joint can output the corresponding joint reference. Owing to the use of motion owners, even if the operations of the two ALs collide with each other, an appropriate operation can be generated by assigning the motion owner to a more important mission. For example, when a user wants to create an AL to pick up an object, if an AL to prevent collisions is running, then the user simply creates an AL for the picking operation. Then, if a collision is anticipated, the motion owner of all joints of the robot is transferred to the AL for avoiding the collision. In addition, a new motion is created in the AL for avoiding collisions and the joint reference for the motion is output to Daemon. In this manner, users can develop ALs that suit their goals without having to consider situations such as balance control, collision detection, or the ALs of other users.

Daemon distinguishes between the hardware and the user's AL and acts as a linking medium. A user can only read the sensor data from hardware abstraction shared memories and provide a joint reference to hardware abstraction shared memories. The other shared memories declared by the user are only pathways for exchanging information between the ALs. These memories send and receive information to and from the ALs; however, they do not affect the behavior of the robot. By dividing the shared memory into two parts, the hardware abstraction shared memory and Daemon can be managed as an encapsulated and independent module so that the users who develop the robot motions and algorithms are not required to know how Daemon cooperates with the robot hardware. Consequently, a user can concentrate on studying the motion and algorithms of the robot without considering how to directly drive the hardware. A user can create a program even if he/she is not familiar with the entire system. As described previously, PODO enables users to develop ALs without conflicting with other users owing to the motion owner property, and it facilitates development of new programs through the shared memories and Daemon.

B. Synchronization Between Daemon and ALs for Hard Real-Time Functionality

Daemon consists of a real-time thread, and the tasks in the thread can be finished within the given control period every

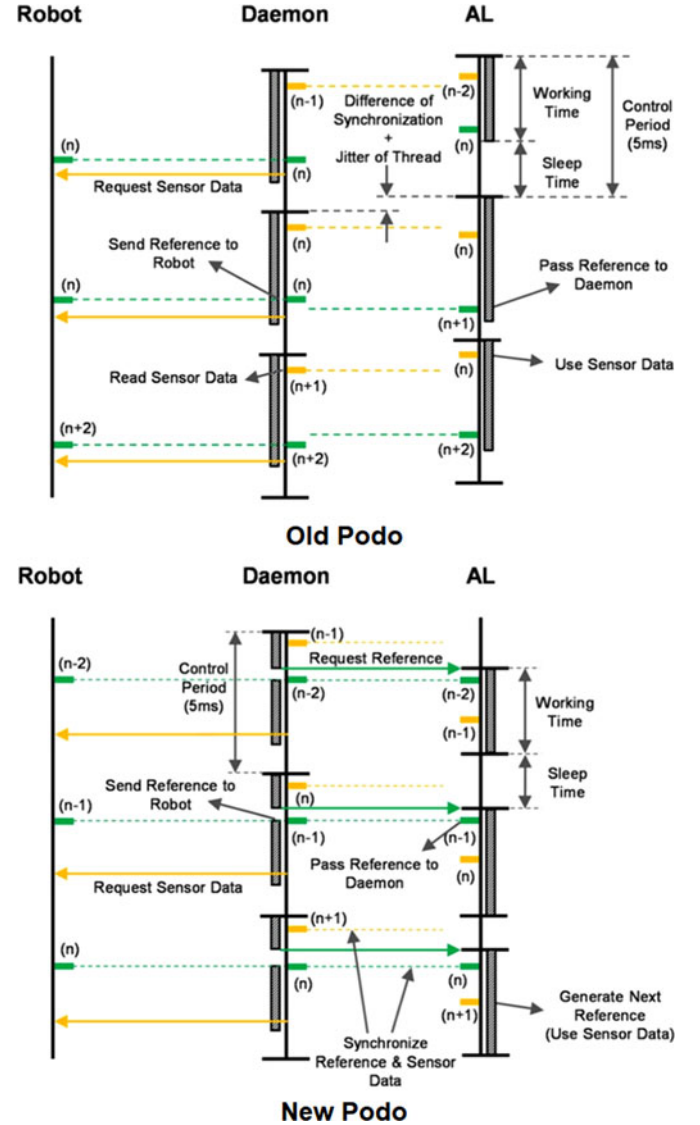


Fig. 17. Operating procedures of the Daemon and AL in the time line. The upper one is for the old PODO and the lower one is for the new PODO. The old one does not synchronize the AL process with the Daemon process, so it may use uneven sensor data and send irregular joint references. The new PODO has improved on that synchronizing issue.

time. Thus, it can be considered as a hard real-time process [23]. However, Daemon performed insufficiently in the previous version of PODO, i.e., the PODO used for the DRC Trials in 2013 [24]. In the earlier PODO, every process contained real-time threads with the same periods (5 ms). Thus, Daemon controlled the devices of the robot and ALs generated joint references to move the robot. Up to this point in this discussion, there is no difference between the old and new PODOs. However, there were time gaps between Daemon and the ALs in the old PODO, as shown in Fig. 17. The jitter of the Xenomai thread [25] and the starting time of the real-time thread cause a problem of dis-synchronization. Moreover, the calculation time of joint references may vary according to the algorithm, and there is no rule for when ALs write references to the shared memory or when Daemon reads these references and sends them to the robot. Then, from the viewpoint of the robot, the end devices

may receive the same references twice within twice the time of the control period while the system omits appropriate references (the robot column of the old PODO in Fig. 17 shows an explanation of this). This situation causes a jerking motion of the robot; this is more dangerous for faster motions such as dynamic walking and fast arm manipulation.

In the case of the new PODO, there is a synchronizing procedure between Daemon and the ALs. The first action of Daemon is to read sensor data from the buffer, which stores the incoming CAN messages of the sensor board and motor controller in a nonreal-time thread. Then, Daemon updates the sensor data of the shared memory with the read value. For the next sequence, Daemon changes two types of flags, i.e., the Ack and Sync flags, in the shared memory. From the change in the Sync flag, the ALs recognize that Daemon wants to synchronize; the ALs send a calculated joint reference and an Ack flag through the shared memory. During this sequence, Daemon waits until it receives an Ack signal from the ALs or until there is a time constant, which is the maximum waiting time for the Ack signal not to exceed the given real-time period. After Daemon receives the Ack signal and updates the joint references, it sends the references to the motor controllers of the robot. Then, the motor controllers receive these messages and move the joints. Finally, Daemon requests sensor data for the next step. The ALs perform their jobs while Daemon works as described previously. There are two nonreal-time threads in the AL, i.e., the task thread and the Sync watching thread. The task thread possesses the user algorithm and the generating code for the next joint references. This thread can access the shared memory, and thus developers can use the sensor or encoder information from the shared memory while developing their algorithms. A special characteristic of the task thread is that it suspends itself whenever it performs a thread once. The Sync watching thread observes the Sync flag of the shared memory; then, it sends the Ack signal and the joint references that were generated at the task thread to Daemon. The Sync watching thread resumes the task thread to calculate the next references. At this time, the sensor data in the shared memory are already updated. In this manner, PODO eliminates jerking motion and improves real-time performance compared to the old version. In Section IV, the experiments and the experimental results are explained to verify that PODO provides hard real-time functionality and usability in collaborative development, which are the proposed design goals.

IV. EXPERIMENTS

A. Experiment on Manipulation Improvement

A simulation experiment was performed to determine the degree to which the performance of DRC-HUBO+ is improved

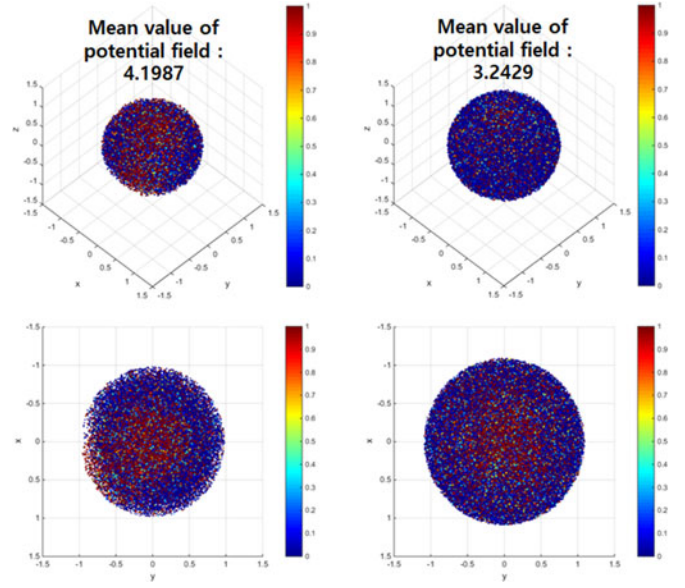


Fig. 18. Graphs comparing the potential fields of DRC-HUBO-1 (left) and DRC-HUBO+ (right). The position of each point represents the position of the left arm end effector, and the color of each point indicates the potential field at that position. The upper graphs are drawn in 3-D, and the lower graphs are those obtained by looking down vertically.

compared to that of DRC-HUBO-1. The potential field is a common indicator for joint limit avoidance; therefore, in this experiment, we used the potential field as a function of the angle close to the joint limit as a comparison index to determine the relation to the joint limit; see (2) at the bottom of this page. where $\theta_{lim\ i}$ is the joint limit and $\theta_{buff\ i}$ is the value at which the potential field begins to increase to prevent reaching the joint limit. The comparison graphs are provided in Fig. 18, which shows that as arm length increases, the working radius of DRC-HUBO+ increases compared to that of DRC-HUBO-1. In addition, because of waist rotation, the working radius of DRC-HUBO+ is constant for 360°, whereas DRC-HUBO-1 cannot perform work on its reverse side. Furthermore, the average value of the entire potential field of DRC-HUBO+ is lower than that of DRC-HUBO-1. From the result of the above-mentioned experiment, we can observe improved manipulability of the new robot compared to that of the existing robot.

B. Experiment to Verify Stable Mobility

The wheel mode was confirmed to be always stable except on steep slopes. In this section, we discuss the stability during the walking mode and when changing between modes, when the robot may experience lesser stability.

$$U_{Li}(\theta) = \begin{cases} \frac{1}{2}\eta_i \left(\frac{1}{\theta_i - \theta_{lim\ i}} - \frac{1}{\theta_{buff\ i}} \right)^2; & \theta_{lim\ i} < \theta_i \leq \theta_{lim\ i} + \theta_{buff\ i} \\ 0; & \theta_{lim\ i} + \theta_{buff\ i} < \theta_i \leq \theta_{lim\ 2\ i} - \theta_{buff\ 2\ i} \\ \frac{1}{2}\eta_i \left(-\frac{1}{\theta_i - \theta_{lim\ 2\ i}} - \frac{1}{\theta_{buff\ 2\ i}} \right)^2; & \theta_{lim\ 2\ i} - \theta_{buff\ 2\ i} < \theta_i \leq \theta_{lim\ 2\ i} \end{cases} \quad (2)$$

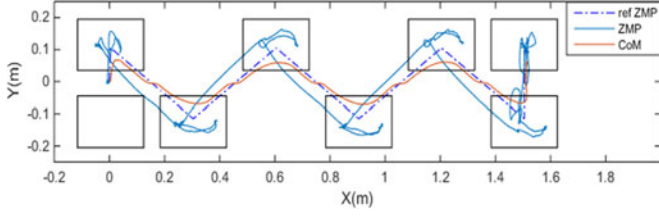


Fig. 19. ZMP and CoM trajectories when the robot walks forward with a step length of 0.3 m and a period of 0.9 s.

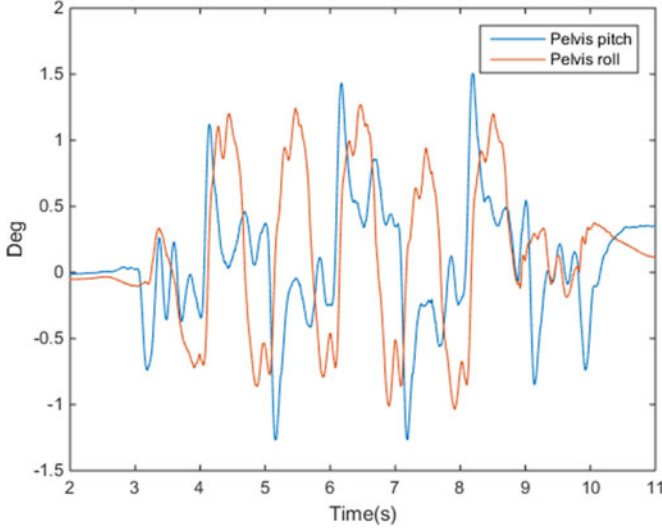


Fig. 20. Measured pelvis orientation when the robot walks forward with a step length of 0.3 m and a period of 0.9 s.

First, the stability of the robot while walking was analyzed. As described in Section II-F, the preview control algorithm was employed to generate the gait pattern, and the pattern was stabilized using the upright posture and ZMP controllers. Fig. 19 shows the reference ZMP, measured ZMP, and control input data obtained when the robot walked forward with a step length of 0.3 m and a step period of 0.9 s. The fact that the measured ZMP is always located in the support polygon confirms that the robot walked stably. Pelvis orientation data, which were measured by the gyro sensor, are presented in Fig. 20. The maximum tilt angle of the pelvis was measured to be 1.505° , and this result shows that the angle of the upper body was maintained as straight.

Next, we examined the stability and speed of kneeling motion while using the mode switching time minimization algorithm discussed in [16]. We measured the ZMP and confirmed the stability of the knees while bent by checking that the ZMP was within the support polygon. The ZMP measurements obtained during kneeling process are shown in Fig. 21. As the ZMP remains close to zero during this step, it can be confirmed that the kneeling process is stable. The results of these two experiments confirm that mobility is stable as the ZMP is always located inside the support polygon, even during walking and when the mode is changed.

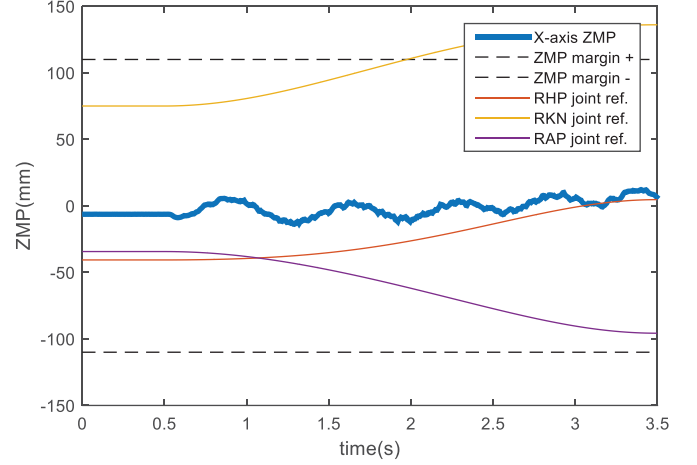


Fig. 21. ZMP variation during posture transformation process 1 (kneeling). The ZMP is held near zero by the ZMP-constrained optimal time control.

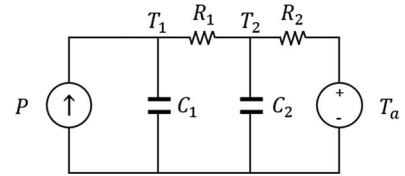


Fig. 22. Thermal model of motors.

C. Experiments on Heat Dissipation for Higher Joint Torques

In this section, we describe the walking simulation used to determine the heat continuously generated by the motor and the subsequent test of the effectiveness of the developed motor cooling device. In addition, a comparative experiment involving frame heat dissipation, an air-cooling device, and a water-cooling device was conducted.

DRC-HUBO+ uses EC-4pole 200 W motors from Maxon, except for its head and wrist joints. These motors are operated below 16 000 r/min. In this range, the Joule power loss P is a major cause of heat generation [26], and it is given by

$$P = I^2 R_e \quad (3)$$

where I is the current input into the motor and R_e is the phase-to-phase resistance of the motor, which can be linearized by applying (4)

$$R_e = R_0(1 + \alpha_{Cu}(T - T_0))\sqrt{b^2 - 4ac} \quad (4)$$

where the temperature resistance coefficient of the winding α is the same as that of copper ($\alpha_{Cu} = 0.0039$) and R_0 is the nominal resistance at nominal temperature T_0 . When the motor core temperature T reaches the thermally critical temperature of the winding ($T_{1\max} = 155^\circ\text{C}$), the phase-to-phase resistance increases by almost 50% [27].

We built the temperature model shown in Fig. 22 to estimate the temperature of the motor core. The equations of the model

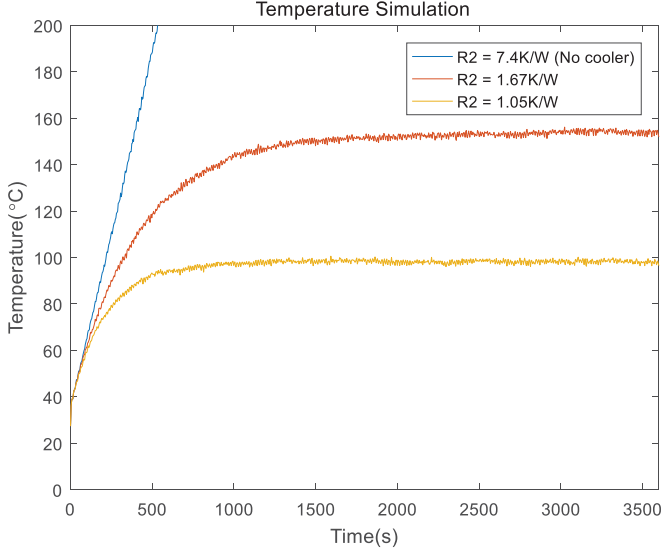


Fig. 23. Temperature simulation result.

are

$$\frac{dT_1}{dt} = \frac{1}{C_1} \left(P - \frac{T_1 - T_2}{R_1} \right) \quad (5)$$

and

$$\frac{dT_2}{dt} = \frac{1}{C_2} \left(\frac{T_1 - T_2}{R_1} - \frac{T_2 - T_a}{R_2} \right) \quad (6)$$

where T_1 , T_2 , and T_a are the motor core temperature, motor housing temperature, and ambient temperature, respectively, R_1 and R_2 are the thermal resistances between the housing and motor core and between the housing and ambient regions, respectively, and C_1 and C_2 are the heat capacities of the motor core and motor housing, respectively. According to (5) and (6), the core temperature can be estimated by measuring T_2 .

The knee joints of a humanoid robot are typically loaded with the highest torque during operation. To design an actuation system suitable for the knee joints, we designed a heat dissipation system using the torque required for the robot to walk for 1 h at 1.5 km/h as reference torque. Fig. 23 presents the motor core temperature simulation results based on the change in R_2 . Without a cooling device ($R_2 = 7.4$ K/W), T_1 exceeds the permissible core temperature $T_{1\max}$ (155 °C). To achieve continuous walking, R_2 should be less than 1.67 K/W. However, owing to various factors, R_2 must actually be lower. For example, the ambient temperature, T_a , may be higher than room temperature (25 °C). In addition, there may be situations in which more torque is required. Therefore, we designed the motor heat dissipation device so that the maximum temperature of the motor would not exceed 100 °C during continuous usage for safety. The simulation results show that the thermal resistance of the knee motor cooling system should not exceed 1.05 K. In addition, we tested whether our heat dissipation parts met the required criteria.

We compared the following three types of cooling solutions: Frame heat dissipation, an air-cooling system using a radiation

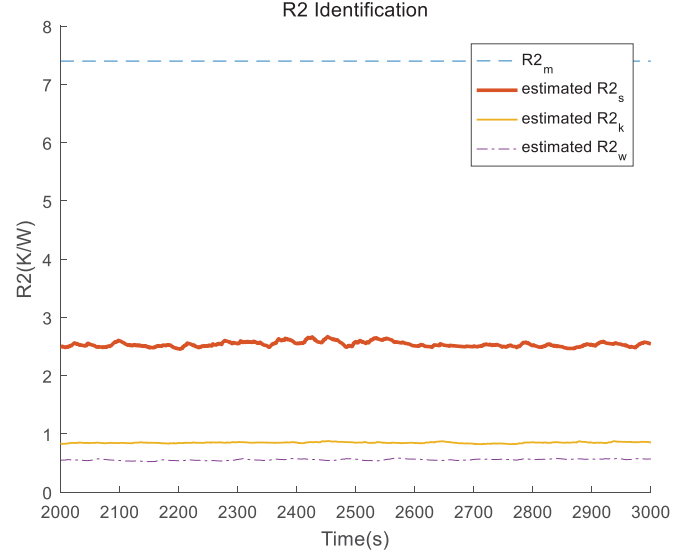




Fig. 24. R_2 identification result.

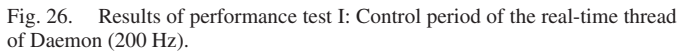
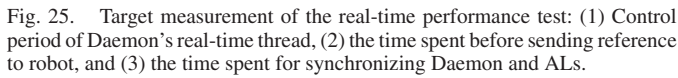
TABLE III
THERMAL RESISTANCE AND TORQUE-PER-WEIGHT COMPARISON
FOR EACH COOLING METHOD

	No cooler	Frame cooling	Air cooling	Water cooling
				
$R_2(\text{K/W})$	7.4K/W	2.53K/W	0.85K/W	0.56K/W
$\tau_{m\text{cont}}$	101.6mNm	249.9mNm	401.8mNm	470.5mNm
mass	300g	300g	687.5g	582.6~1006g

fin and fan, and a water-cooling system. The R_2 identification results obtained using the thermal model after allowing sufficient time to eliminate the effects of heat capacities are presented in Fig. 24. The R_{2f} of frame heat dissipation is 2.53 K/W, the R_{2k} of the air-cooling system is 0.85 K/W, and the R_{2w} of the water-cooling system is 0.56 K/W. The relationship between the continuous torque of the motor $\tau_{m\text{cont}}$ and thermal resistance is

$$\tau_{m\text{cont.}} = K_t \sqrt{\frac{T_{1\max} - T_a}{(R_1 + R_2)R_e}}. \quad (7)$$

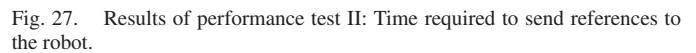
Table III shows the thermal resistance of each cooling unit and the continuous torque that the motor can produce. The water-cooling system yields the highest performance, and the air-cooling system performs sufficiently well. According to the obtained value, the motor with the air-cooling system produces four times as much continuous torque as a motor without any cooling device. This is only 15% less than water-cooling methods, and the performance of the air-cooling system is sufficient for this robot. Moreover, as mentioned earlier, water-cooling systems have various risks. Based on these factors, we used an air-cooling system through a heat sink for the knee and



hip-pitch joints in DRC-HUBO+. Furthermore, the joints that do not require high torque, such as the shoulder and ankle joints, were designed to maintain higher torques than those provided by conventional motors via heat dissipation through the frame.

D. PODO Performance Test

This section describes the real-time functionality of PODO and presents quantitative experimental results that verify its performance. The experiments were conducted using the same environment as that in the DRC Finals. A mini PC (Intel NUC Kit D54250WYK, i5-4250U 1.30 GHz \times 2, 4 GB RAM) was employed, and the control cycle was set to 200 Hz. Fig. 25 shows three time measurements obtained to test the performance of PODO. The first is the period of a real-time thread in Daemon, and it ensures the consistency of the Daemon thread period. The second is the offset time between the starting point of a new cycle in a real-time thread and the time at which the reference is transferred to the robot. This measurement shows whether the action delay is bounded. The last measurement indicates the time required to synchronize Daemon and the ALs; it is the period from just after the “Sync” signal is sent to immediately after the “Ack” signal is received. This measurement



shows how quickly PODO reacts and how much it influences the second measurement. Tests were performed with 1, 3, 5, and 10 working ALs, and 6000 samples were acquired from each test. Each task thread of an AL contained the same floating computations. We designed the computation load to require approximately 130 μs during independent operation. As shown in Fig. 26, the control periods in Daemon are concentrically distributed close to 5000 μs . The jitter minimization performance is remarkable because the maximum jitter is within the 5 μs bound. In other words, the real-time thread of Daemon starts a given job extremely precisely. Fig. 27 presents the temporal histograms of the second measurements. In Fig. 27, the peak value shifts from left to right as the number of running ALs increases; this indicates that the time required to send the references to the devices increases. However, the maximum values

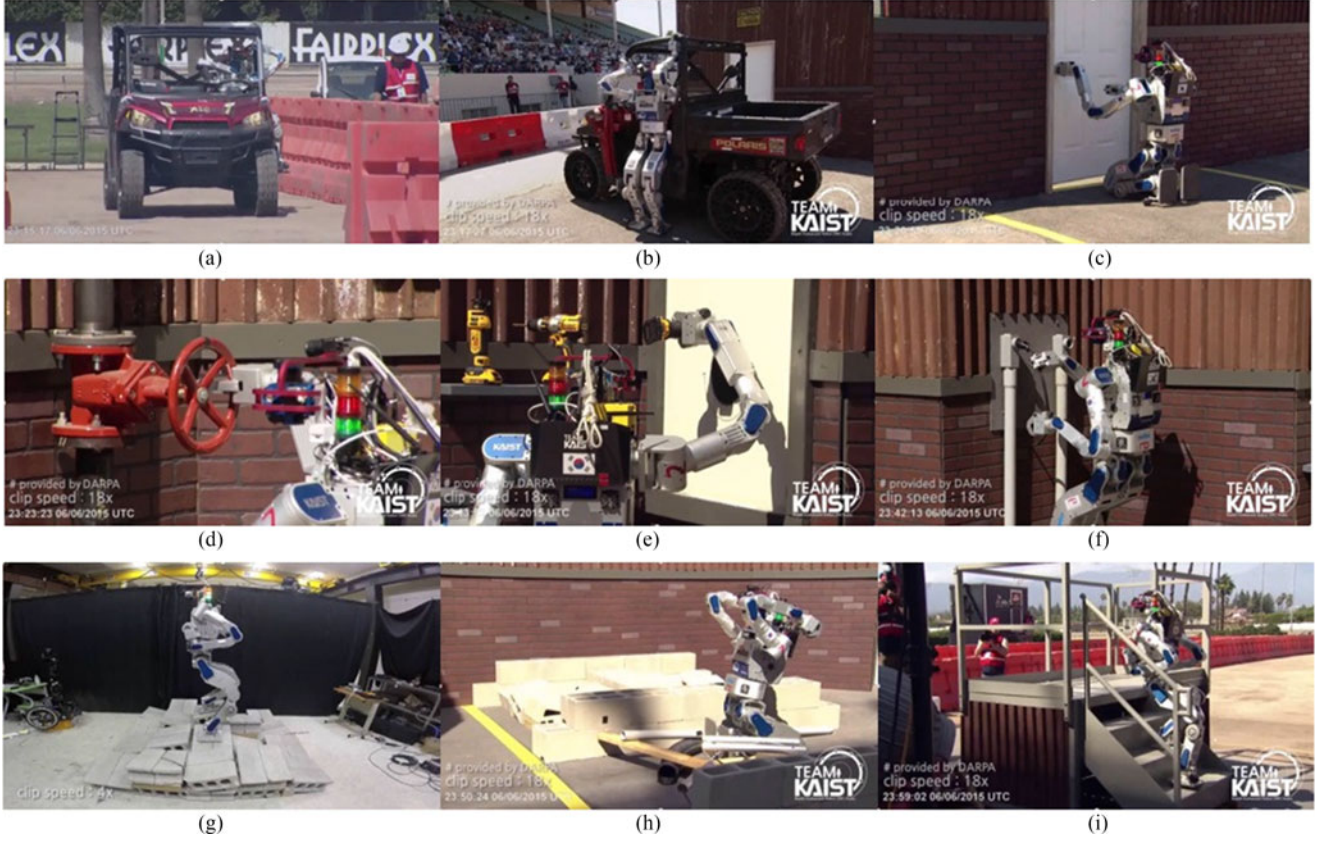


Fig. 29. Tasks of the DRC Final. (a) Drive task. (b) Egress task. (c) Door task. (d) Valve task. (e) Wall task. (f) Surprise task. (g) Terrain task. (h) Debris task. (i) Stair task.

are bounded between 10 and 60 μs ; thus, the time varies within a range of only 50 μs . This variation is only 1% of the duration of a given control period, which demonstrates that the motor controllers of the robot receive uniformly periodic joint references. Finally, PODO can be used to control a multirate system such as that of DRC-HUBO+. Fig. 28 presents four histograms that represent the time required to synchronize Daemon and the ALs. The graphs exhibit a tendency similar to that of the results shown in Fig. 27, with a variation of 10–60 μs . Thus, it can be concluded that the synchronization process is dominant when sending the joint references to the robot devices. This feature originates from the mechanism of the synchronization process. To catch Sync signals, a thread in the AL watches for these signals with the highest priority. Even though the checks are simple and few, their frequency may reduce system performance. As the computing power of the embedded PC is limited, there must be a tradeoff between reactivity and power consumption. We considered that a control period of less than 10% for the low-level controller would be reasonable based on the time variation of the actions, and thus we selected 50 μs as the period of the threads for observing Sync signals.

The results of these three experiments demonstrate the hard real-time functionality discussed in Section III. In addition, the results are almost constant regardless of the number of working ALs. This characteristic verifies the possibility of collaborative development because the structural independence and

computational performance support developers who want to program their own algorithms. Moreover, none of the logged data exceed 300 μs until Daemon ends its thread. Daemon was designed to be light and fast, and it only has the task of controlling low-level devices so that the ALs can use most of the computational resources.

As a result, PODO in the DRC-HUBO+ system exhibited Sufficient performance while 16 ALs were used simultaneously.

V. CONCLUSION

In this paper, we described the development of DRC-HUBO+. To satisfy the requirements mentioned in the first part of this paper, we designed DRC-HUBO+, which is 170 cm tall and has a mass of 80 kg, to have a height and mass similar to those of an average adult man. Owing to its exoskeletal structure and the avoidance of cantilever forms, DRC-HUBO+ has a rigid but light body. The hardware of DRC-HUBO+ has sufficient mobility, manipulability, power, and autonomy for it to perform human tasks by teleoperation. Moreover, the modularized joints and user-friendly software architecture of this robot facilitate research related to the robot tasks. All DRC Final tasks were executed successfully, as shown in Fig. 29. The enhanced manipulability facilitated all tasks that were performed using the robot arms. For the drive task, the infinite rotation ranges of the wrist joints made it easy for DRC-HUBO+ to steer

using the handle. It grabbed the center of the handle and could steer the car simply by rotating its wrist joints. In addition, it could turn off the valve easily in this manner. Both hands pulled on the roll cage to reduce the impact during landing when getting out of the car. This task requires each hand to support up to 15 kg, and the new robot gripper, which is lightweight but provides strong gripping force, allows for the use of the egress method. By applying this method, we finished the egress task in the shortest time among the teams at the DRC Finals. In addition, the new gripper was used for multiple tasks such as opening a door, boring a hole in a wall using a drill, and clearing debris. Moreover, the ability of the waist joint to rotate up to $\pm 350^\circ$ enabled the robot to perform work regardless of direction and was advantageous for forward and backward motion. The dual mode strategy, which included the walking and wheel modes, improved mobility and stability. The wheel mode was used for moving on a flat surface or performing tasks that require the application of strong external force, such as the door, wall, and debris tasks. The robot could move and perform the tasks stably and rapidly when supporting area was increased. In contrast, biped walking was used in the terrain and stair tasks because the robot was required to consider the variations in the height of the terrain. The software architecture, PODO, controlled DRC-HUBO+ without any failure and enabled multiple researchers to successfully prepare each task. Finally, by obtaining full scores at the end of the DRC Finals, we proved that DRC-HUBO+ can replace humans in disaster situations.

REFERENCES

- [1] Defense Advanced Research Projects Agency, "DARPA DRC," Defense Advanced Research Projects Agency, Arlington, VA, USA, 2015. [Online]. Available: www.theroboticschallenge.org
- [2] Atlas, [Online]. Available: http://www.bostondynamics.com/robot_Atlas.html
- [3] K. Kaneko *et al.*, "Humanoid robot HRP-2Kai—Improvement of HRP-2 towards disaster response tasks," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, 2015, pp. 132–139.
- [4] S. J. Yi *et al.*, "Team THOR's entry in the DARPA robotics challenge trials 2013," *J. Field Robot.*, vol. 32, no. 3, pp. 315–335, 2015.
- [5] N. Paine *et al.*, "Actuator control for the NASA-JSC Valkyrie humanoid robot: A decoupled dynamics approach for torque control of series elastic robots," *J. Field Robot.*, vol. 32, no. 3, pp. 378–396, 2015.
- [6] K. Kojima *et al.*, "Development of life-sized high-power humanoid robot JAXON for real-world use," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, Nov. 2015, pp. 838–843.
- [7] M. Schwarz *et al.*, "NimbRo rescue: Solving disaster-response tasks with the mobile manipulation robot momaro," *J. Field Robot.*, vol. 34, no. 2, pp. 400–425, 2017.
- [8] P. Hebert *et al.*, "Mobile manipulation and mobility as manipulation—Design and algorithms of RoboSimian" *J. Field Robot.*, vol. 32, no. 2, pp. 255–274, 2015.
- [9] A. Stentz *et al.*, "Chimp, the CMU highly intelligent mobile platform," *J. Field Robot.*, vol. 32, no. 2, pp. 209–228, 2015.
- [10] I. W. Park *et al.*, "Online free walking trajectory generation for biped humanoid robot KHR-3 (HUBO)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006.
- [11] J. Lim *et al.*, "Robot system of DRC-HUBO+ and control strategy of team KAIST in DARPA robotics challenge finals," *J. Field Robot.*, vol. 34, no. 4, pp. 802–829, 2017.
- [12] Y. Ito *et al.*, "Development and verification of life-size humanoid with high-output actuation system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 3433–3438.
- [13] J. Urata, Y. Nakanishi, K. Okada, and M. Inaba, "Design of high torque and high speed leg module for high power humanoid," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4497–4502.
- [14] Adaptive Robot Gripper. [Online]. Available: <http://robotiq.com/products/industrial-robot-hand/>
- [15] L. Birglen and C. M. Gosselin, "Kinetostatic analysis of underactuated fingers," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 211–221, Apr. 2004.
- [16] H. Bae, I. Lee, T. Jung, and J. H. Oh, "Walking-wheeling dual mode strategy for humanoid robot, DRC-HUBO+," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 1342–1348.
- [17] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. Automat. Control*, vol. AC-30, no. 6, pp. 531–541, Jun. 1985.
- [18] S. Kajita *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, vol. 2, pp. 1620–1626.
- [19] J. Y. Kim, I. W. Park, and J. H. Oh, "Experimental realization of dynamic walking of the biped humanoid robot KHR-2 using zero moment point feedback and inertial measurement," *Adv. Robot.*, vol. 20, no. 6, pp. 707–736, 2006.
- [20] M. S. Kim and J. H. Oh, "Posture control of a humanoid robot with a compliant ankle joint," *Int. J. Humanoid Robot.*, vol. 7, no. 1, pp. 5–29, 2010.
- [21] "Xenomai—Real-time framework for Linux," Xenomai, Web. 8, Jan. 2014.
- [22] N. Dantam and M. Stilman, "Robust and efficient communication for real-time multi-process robot software" in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2012, pp. 316–322.
- [23] K. G. Shin and P. Ramanathan, "Real-time computing: A new discipline of computer science and engineering," *Proc. IEEE*, vol. 82, no. 1, pp. 6–24, Jan. 1994.
- [24] J. S. Lim and J. H. Oh, "Backward ladder climbing locomotion of humanoid robot with gain overriding method on position control," *J. Field Robot.*, vol. 33, no. 5, pp. 687–705, Aug. 2016.
- [25] J. H. Brown and B. Martin, "How fast is fast enough. Choosing between Xenomai and Linux for real-time applications," 2010. [Online]. Available: <https://www.osadl.org/fileadmin/dam/rtlws/12/Brown.pdf>
- [26] O. Stemme and P. Wolf, *Principles and Properties of Highly Dynamic DC Miniature Motors*. Sachseln, Switzerland: Maxon Motor, 1994.
- [27] Maxon Motor, Sachseln, Switzerland. [Online]. Available: <http://www.maxonmotor.com>



Taejin Jung received the B.S. degree in mechanical engineering from Hanyang University, Seoul, South Korea, in 2009 and the M.S. degree in robotics program from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012. Since 2012, he has been working toward the Ph.D. degree at the Department of Mechanical Engineering, KAIST.

His research interests include humanoid robot, robot design, robot hand, gripper, and quadruped robot. He designed DRC-HUBO-1 and DRC-HUBO+. He was a member of TEAM KAIST that won the first place in DARPA Robotics Challenge Final 2015.



Jeongsoo Lim received the B.S., M.S., and Ph.D. degrees from the School of Mechanical, Aerospace and Systems Engineering, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2010, 2012, and 2016, respectively.

Since 2016, he has been conducting research on humanoid and cooperative robot with Rainbow Robotics, Daejeon, South Korea, as a Principal Engineer and a Board Member. His research interests include robotic system, robot software framework, disaster robot and system, and designing motions for humanoid robots.



Hyoin Bae received the B.S. and M.S. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2012 and 2014, respectively. Since 2014, he has been working toward the Ph.D. degree in mechanical engineering at KAIST and working on the project of development for humanoid robots HUBO2 and DRC-HUBO.

His research interests include state estimation for multidegree of freedom system, disturbance estimation, humanoid robot design, and sensor fusion algorithm.



Kang Kyu Lee received the B.S. and M.S. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2013 and 2015, respectively. Since 2015, he has been working toward the Ph.D. degree in mechanical engineering at KAIST and working on the project of development for humanoid robots DRC-HUBO.

His research interests include whole body control, floating-base dynamics, motor thermal control, and humanoid design for a humanoid robot. He is a member of TEAM KAIST, which won first place in the DARPA Robotics Challenge Final 2015.



Hyun-Min Joe received the B.S. degree in mechanical engineering from Kyungpook National University, Daegu, South Korea, in 2011 and the M.S. degree in robotics program from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2013. He is currently working toward the Ph.D. degree at the Department of Mechanical Engineering, KAIST.

His research interests include walking control, motion planning, and applications of robotics and control. He was a member of TEAM KAIST, which won

first place in DARPA Robotics Challenge Final 2015 and performed terrain and stair missions.



Jun-Ho Oh (M'13) received the B.S. and M.S. degrees in mechanical engineering from Yonsei University, Seoul, South Korea, in 1977 and 1979, respectively, and the Ph.D. degree in mechanical engineering from University of California, Berkeley, Berkeley, CA, USA, in 1985.

From 1979 to 1981, he was a Researcher with Korea Atomic Energy Research Institute. Since 1985, he has been with the Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is currently a significant Professor and the Director of Humanoid Robot Research Center.

From 1996 to 1997, he was a Visiting Research Scientist with University of Texas Austin. His research interests include humanoid robots, adaptive control, intelligent control, nonlinear control, biomechanics, sensors, actuators, and application of microprocessor.

Dr. Oh is a member of the KSME, KSPE, and ICASE.