Humanoid Robotics Platforms developed in HRP

Hirohisa Hirukawa[1], Fumio Kanehiro[1], Kenji Kaneko[1], Shuuji Kajita[1], Kiyoshi Fujiwara[1],
Yoshihiro Kawai[1], Fumiaki Tomita[1], Shigeoki Hirai[1], Kazuo Tanie[1], Takakatsu Isozumi[2],
Kazuhiko Akachi[2], Toshikazu Kawasaki[2], Shigehiko Ota[2], Kazuhiko Yokoyama[3], Hiroyuki Handa[3],
Yutaro Fukase[4], Jun-ichiro Maeda[4], Yoshihiko Nakamura[5], Susumu Tachi[5], and Hirochika Inoue[5]

[1] National Institute of Advanced Industrial Science and Technology(AIST), AIST Tsukuba Central 2, 1-1-1
Umezono, Tsukuba, 305-8568 Japan
`hiro.hirukawa@aist.go.jp`
[2] Kawada Industries, Inc., 122-1 Hagadai, Hagamachi, Haga-gun, 321-3325 Japan
`taka.isozumi@kawada.co.jp`
[3] Yaskawa Electric Co., 12-1 Ohtemachi, Kokura-kita-ku, Kitakyusyu, 803-8530 Japan
`yoko@yaskawa.co.jp`
[4] Shimizu Co., No.4-17, Ecyujima 3-chome, Koto-ku, Tokyo, 105-8007 Japan
`fukase@shimz.co.jp`
[5] The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
`inoue@jsk.t.u-tokyo.ac.jp`

Abstract. This paper presents humanoid robotics platform that consists of a humanoid robot and an open architecture software platform developed in METI's Humanoid Robotics Project (HRP). The final version of the robot, called HRP-2, has 1540mm height, 58kg weight and 30 degrees of the freedom. The software platform includes a dynamics simulator and motion controllers of the robot for biped locomotion, falling and getting up motions. The platform has been used to develop various applications and is expected to initiate more humanoid robotics research.

1 Introduction

The objectives of Humanoid Robotics Project sponsored by METI/NEDO include the development of a humanoid robotics platform and the exploration of humanoid robot applications. The platform consists of humanoid robots, teleoperation systems and a collection of fundamental software for them. The first version of the humanoid robots is HRP-1 developed by Honda R&D [2,3]. The controller of HRP-1 for its biped locomotion is also developed by Honda R&D. HRP-1 has been used for investigating the applications of a humanoid robot for the maintenance tasks of industrial plants and security services of home and office [4].

As the next step, AIST has replaced the control software to have more flexibility for motions[17]. We call the combination of HRP-1 hardware and AIST controller HRP-1S. We can control the arms and legs at the same time on HRP-1S, which is not possible on HRP-1 due to a restriction of the Honda controller. The applications of a humanoid robot to the teleoperation of industrial vehicles and human care services have been carried out on HRP-1S [4]. The dynamics simulator of humanoid robots has also been developed in HRP [7, 16]. The software developed on the simulator can be applied to the hardware as it is. We call the package of the simulator and the motion control software OpenHRP (Open Architecture Humanoid Robotics Platform).

HRP has also been developing a new humanoid hardware. The leg module was developed at first to examine the required specifications of the legs for biped locomotion[9]. We call the leg module HRP-2L. The arm module was also made to evaluate its performance for cooperative task with a human [4]. It is called HRP-2A. Then the prototype of the whole body humanoid has been developed[10]. The prototype is called HRP-2P. The final version of the new humanoid robot, HRP-2, was developed by refining HRP-2P at various points. Note that OpenHRP is commonly used to develop the software for HRP-2L, HRP-2A, HRP-2P and HRP-2 as well as HRP-1S. The application of a humanoid robot for cooperative task with a human has been investigated on HRP-2.

HRP-2 and OpenHRP are expected to be humanoid robotics platform after HRP is completed. This paper overviews the specifications of HRP-2 and OpenHRP with taking a look for the above mile stones. The rest of the paper is organized as follows. Section 2 visits the mile stones. Section 3 describes the detailed specifications of HRP-2P and how HRP-2P was refined to complete HRP-2. Section 4 overviews OpenHRP. Section 5 concludes the paper.

2 Mile stones of the platform hardware

HRP-1 has 1600mm height, 120kg weight and 30 degrees of the freedom consisting of 2 joints for head, 7 joints for each arm, 1 joint for each hand and 6 joints for each leg. HRP-1 is an enhanced version of Honda P3. The main difference is its interface for teleoperations. Figure 1 shows the front and side views of HRP-1. Figure2 shows the teleoperation cockpit that consists of two master arms, head mounted display, surrounding display and sound display. The cockpit can be connected to HRP-1 via an optical fiber or Ethernet. Though many applications can be investigated on HRP-1, but some applications must require different specifications to the platform. This observation has motivated us to develop HRP-2. As the first step, the leg module and the arm module had been independently developed. Figure 3(a) shows HRP-2L and (b) HRP-2A respectively. HRP-2P was developed after the examinations on them.

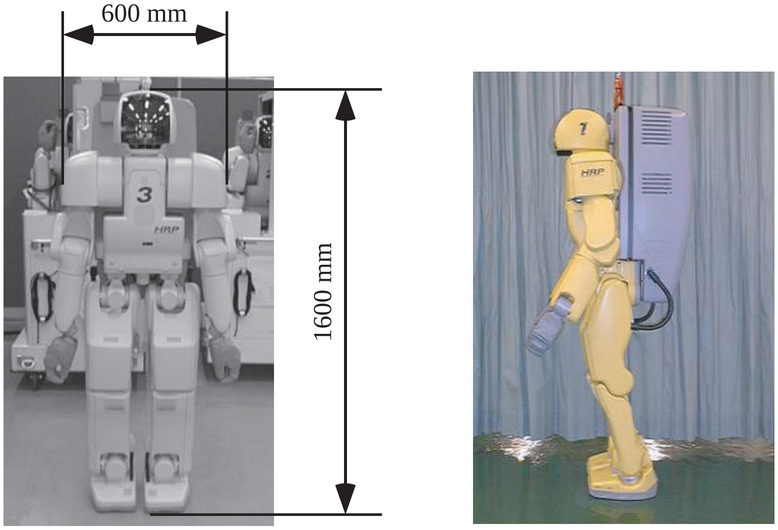

Fig. 1. Humanoid robot HRP-1

Fig. 2. Teleoperation cockpit

3 HRP-2P and HRP-2

3.1 Specifications of HRP-2 prototype model

A snapshot of HRP-2P is shown in Fig.4(a), and the mechanical configuration of HRP-2 prototype, HRP-2P for short, is illustrated in Fig.4(b). As shown in Fig.4, HRP-2P has unique configurations. One is that

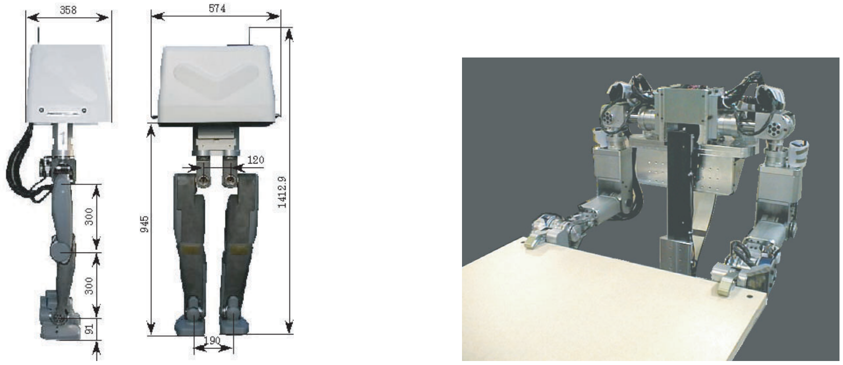

Fig. 3. (a)HRP-2L (b)HRP-2A

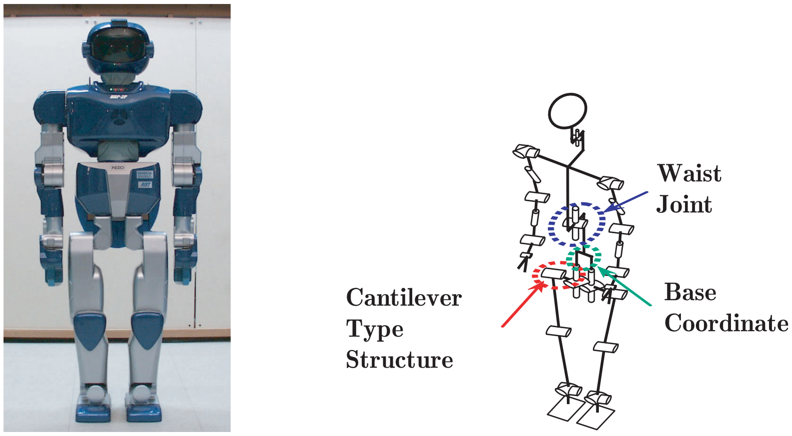

Fig. 4. (a) HRP-2P (b) Configuration of HRP-2P

the hip joint of HRP-2P has a cantilever type structure. The other is that HRP-2P has a waist joint. Table 1 shows the specifications of HRP-2P.

Table 1 Specifications of HRP-2P

D.O.F.	Head: 2 D.O.F (Pitch, Yaw) = 2 D.O.F.
	Arms: 6 D.O.F. (Shoulder:3, Elbow:2, Wrist:1 ×2 = 12 D.O.F.
	Hands: 1 D.O.F. (Open/Close) ×2 = 2 D.O.F.
	Waist: 2 D.O.F. (Pitch, Yaw) = 2 D.O.F.
	Legs: 6 D.O.F. (Hip:3, Knee:1, Ankle:2) = 12 D.O.F.
	Total: 30 D.O.F.
Dimension	Height: 1,549.6 [mm]
	Width: 654.0 [mm]
	Depth: 337.7 [mm]
	Upper arm length: 250.0 [mm]
	Lower arm length: 250.0 [mm]
	Upper leg length: 300.0 [mm]
	Lower leg length: 300.0 [mm]
Weight	Head: 1.2 [kg] = 1.2 [kg]
	Arms: 5.4 [kg/arm]×2 = 10.8 [kg]
	Bodies: 26.9 [kg] = 26.9 [kg]
	Legs: 7.6 [kg/leg]×2 = 15.2 [kg]
	Total: 54.1 [kg]

3.2 Electrical design of HRP-2P

In the electrical design for HRP-2P, several efforts for light weight and realization of compact body were adopted. In this section, the details of electrical design are introduced.

I/O control system Recently, PCI bus becomes the most popular bus in industrial field. However, employing PCI bus brings an issue. That is, PCI bus accepts only four or less PCI boards without a bus-bridge. This causes a serious problem for constructing a humanoid robot. Because, several kinds of function such as DA, AD, counter, and Digital Input/Output, and multi-channels are necessary for control the humanoid robot. In the case of HRP-2P with 30 D.O.F., 30 DC motors, 30 encoders, 3 gyro sensors, 3 acceleration sensors, and 4 force/torque sensors should be controlled using PCI boards occupied within four slots of PCI bus. To overcome this issue, the function boards on the market are not sufficient. We then developed a HRP interface board and a quad force sensors interface board [9].

Using these developed boards, we could complete a compact I/O control system as shown in Fig.5.

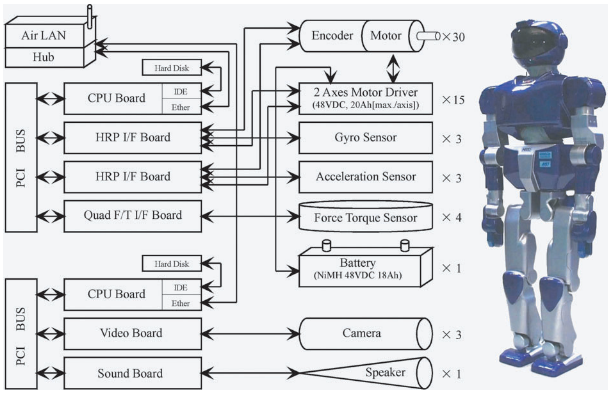

Fig. 5. Electronic system of HRP-2P

Servo driver module To realize humanoid robot using electrical actuators, servo drivers are necessary. Since the humanoid robot has high D.O.F., the volume of servo drivers is significant issue to realize a compact humanoid. To overcome this issue, we have developed a compact servo driver module for HRP-2L [9]. Its volume and weight were almost 15 [%] and 33[%] of smaller servo drivers on the market, respectively. For making it smaller and more efficient, we re-developed it by surface mount technology (SMT). Since one module enables to control two DC motors independently, 15 modules are employed for control 30 DC motors of HRP-2P.

Stereo camera system HRP-2P has three camera units as its vision system which is used to generate a 3D terrain map of the working environment and to find the position and orientation of the target objects. VVV is used as stereo vision software for the purpose[15].

Computer system As shown in Fig.6, HRP-2P has two CPU boards (Pentium III, 1 [GHz]) in the body. One of them is utilized for the realtime controller of whole body motion, while the other is utilized for VVV. The operating system is ART-Linux[5]. ART-Linux enables the execution of realtime processes at the user level so that users can implement realtime applications as if they are non-realtime ones. This feature of ART-Linux is essential for realizing the identical controller for the simulation and the real robot[7].

3.3 HRP-2

The final version of HRP-2 was completed by improving HRP-2P in the following points.

HRP-2 can walk longer than HRP-2P HRP-2 can walk for 60 minutes at the speed of 1.25 km/h by the battery on the robot, whereas HRP-2P could do for 20 minutes at 0.675 km/h. The improvement was realized by using more powerful motors for the legs, introducing a cooling system for them and employing a better battery.

HRP-2 can walk more stably than HRP-2P HRP-2 can walk more stably than HRP-2P, since the rigidity of the mechanism of HRP-2 was improved from that of HRP-2P.

HRP-2 looks better than HRP-2P The chest of HRP-2 has become more compact than that of HRP-2P, and its appearance was designed by a professional designer for humanoid heros in animation whose name is Yutaka Izubuchi. See Fig.6 to see the design picture of HRP-2, and a snapshot of real HRP-2.

Fig. 6. Design picture of HRP-2 and its real counterpart

HRP-2 is expected to be a platform robot for the research and development of humanoid robotics after HRP is completed as well as it is used to investigate the cooperative tasks between a human and a humanoid robot.

4 OpenHRP

4.1 Overview of OpenHRP

Since humanoid robots demand various kinds of software, the architecture of the software must have modular structure. Stasse et al. developed PredN that has a modular structure and can operate with multi-thread on ATM and Ethernet[14]. Oka et al. investigated an asynchronous parallel system[13].

OpenHRP is implemented as a distributed object system on CORBA (Common Object Request Broker Architecture)[18]. A user can implement a controller using an arbitrary language on an arbitrary operating system if it has a CORBA binding. A lot of implementation of ORB exist, Orbix/E is adopted in OpenHRP. Because Orbix/E has C++ and Java binding and supports several operating systems. OpenHRP consists of several CORBA servers. Each server can be replaced with another implementation if it has the same interface defined by IDL (Interface Definition Language). The users of OpenHRP can replace a server of OpenHRP or add a new server to examine their contributions while using the remaining part of OpenHRP. In the sense, the users can concentrate their effort to the point where they have their interests. Using the language independence feature of CORBA, some of servers are implemented using Java and Java3D, the others are implemented using C and C++. Currently, OpenHRP supports Windows 2000/XP/98/Me and Linux. Each server consists of a CORBA interface part, a native language interface part and a core logic part. A real-time routine uses native interface and a non real-time routine does CORBA interface. As a result, lots of codes are shared between the simulator and the controller, and this sharing let the development more efficient.

4.2 Configuration of OpenHRP

A simulation is controlled by a CORBA client which has a graphical interface shown in Fig.7 which is called ISE (Integrated Simulation Environment). ISE uses services provided by four CORBA servers, i.e.

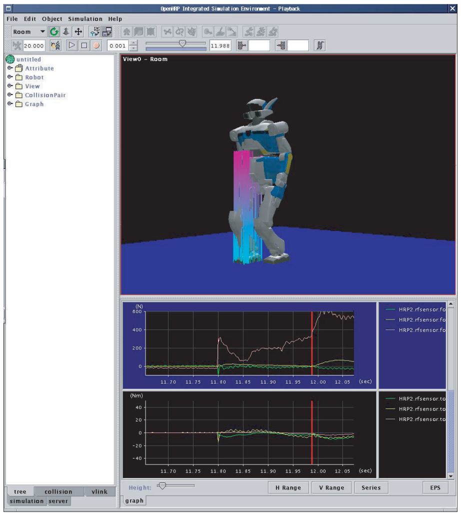

Fig. 7. Integrated Simulation Environment of OpenHRP

collision checker, model parser, dynamics and view simulator. A user can implement a real-time control algorithm which is to be embedded in a CORBA skeleton prepared beforehand.

The configuration of OpenHRP is shown in Fig.8. The functions of each server are as follows.

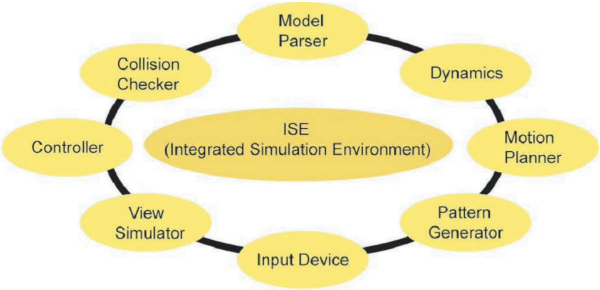

Fig. 8. CORBA objects of OpenHRP

ModelParser This server loads a VRML file describing the geometric models and dynamics parameters of robots and their working environment, and provides these data to other servers.

CollisionChecker The interference between two sets of triangles is inspected, and the position, normal vector and the depth of each intersecting point are found.

Dynamics The forward dynamics of the robots is computed.

Controller This server is the controller of a robot, which is usually developed by the users of OpenHRP.

View Simulator A field of view from cameras on a humanoid is generated.

Pattern Generator A dynamically stable walking motion is calculated and trajectories of joint angles and ZMP (Zero Moment Point) are generated.

Motion Planner A motion path which is collision free and dynamically stable is computed.

Input Device A status of an input device such as a joystick is provided.

4.3 Dynamics simulation

Using the servers, the forward dynamics of the robots are computed in the following procedure. Setting up of the simulation environment (1) ModelParser reads a VRML file via HTTP protocol. The kinematics and dynamics parameters are sent to Dynamics and the geometric model is to CollisionChecker. Execution of the dynamics simulation (2) Controller reads the outputs of the simulated sensors while communicating with Dynamics. (3) Controller and Dynamics execute the computations. Note that these computations can be run in parallel. The outputs of Controller are the torques of the actuators, and those of Dynamics are the updated states of the robot. (4) While the forward dynamics is computed, CollisionChecker is called to find the position, normal vector, and the depth of each intersecting point. (5) After these computations, Controller sends the control outputs to Dynamics Visualization and recording (6) ISE acquires the current states of the world from Dynamics, visualizes the simulated world and records it. While the dynamics simulation, the view simulator can generate the field of the view from the cameras of the robot.

4.4 Controllers

Unification of the applications The applications developed on the simulator can be ported as is to the real counterpart. This concept is illustrated in Fig.9. The portability of the applications on OpenHRP between the simulator and the real robot is realized by three tricks; the employment of ART-Linux, hardware abstraction and the sychronization mechanism.

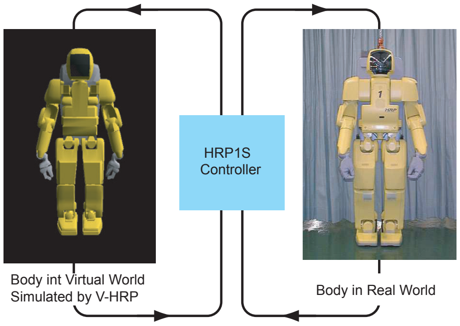

Fig. 9. Unification of the applications on OpenHRP

It is not possible to realize the portability on realtime operating system like VxWorks on which applications are compiled by a cross-compiler on a different operating system. Though it may be possible to realize the compatibility at the source level, but the development process is not efficient then. Besides, rich development tools are not available on VxWorks. RT-Linux[19] is another candidate for our purpose, but realtime tasks can be implemented as kernel modules on RT-Linux and then a limited library can be used from the realtime tasks. Besides, a special mechanism called RT-FIFO is used for the communication between the kernel and user space, therefore the mechanism must be virtualized as well to realize the portability.

ART-Linux is a realtime extension of Linux on which realtime tasks can be executed in the user space. Thanks to the feature, realtime tasks can utilize most library on Linux, and the portability can be realized without the virtualization of RT-FIFO. The application programming interface (API for short) of the virtual robot on the simulator is not identical with that of the real robot in general. This is the reason why some porting task is demanded. OpenHRP introduces an abstracted API, and uses an emulation adaptor on the API of the virtual robot and a hardware adaptor on that of the real robot to absorb the difference of APIs. This mechanism is illustrated in Fig.10.

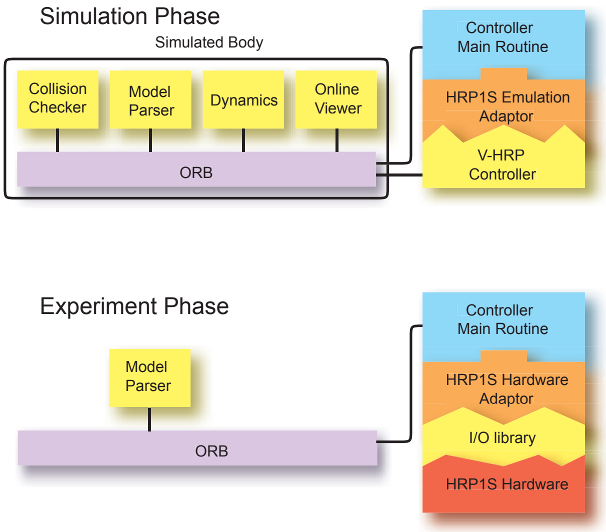

Fig. 10. Abstracted API and the adaptors

The abstracted interface consists of four APIs; `open()`, `read()`, `write()` and `close()`. `open()` establishes a connection to the robot and initialize it. `read()` reads the outputs of the sensors periodically. `write()` outputs the commands to the robot. `close()` terminates the connection to the robot and cleans up it. These APIs are implemented as a C++ class, and the adaptors inherit this class and must be implemented according the specifications of the robot. The definition of the abstracted interface is as follows.

```
class robot_adaptor
{
public:
  virtual bool open(int argc, char *argv[]);
  virtual bool close();
  virtual bool read(robot_state *rs);
  virtual bool write(motor_command *mc);
};
```

The hardware adaptor is implemented by *art_enter*, *art_wait* and *art_exit* which are system calls of ART-Linux for realtime tasks. `open()` calls *art_enter* to make a process into a realtime one. `read()` calls *art_wait* to get the outputs of the sensors periodically, and `write()` to send the commands to the robot. `close()` calls *art_exit* to restore the realtime process to non-realtime one.

We can switch between the simulation and the experiment by replacing the adaptor. The adaptors are given by shared objects which can be linked dynamically according to the simulation or the experiment.

The time passes at a different speed in the simulated world and real one. In general, the forward dynamics simulation takes longer time than the corresponding real time. So it must be synchronized to realize the binary compatibility. To this end, the time in the simulation is updated when an integration interval for the forward dynamics computation is terminated, and that in the real world proceeds when `read()` calls *art_wait*.

We have developed several motion controllers for humanoid robots that realize biped locomotion, lying down and getting up. The biped locomotion controller consists of a walking pattern generator and a feedback stabilizer described as follows.

Walking pattern generator A walking pattern generator is a part of the controller which manages a dynamically stable biped walking. During walking, the dynamics of a biped robot can be approximated by a single inverted pendulum which connects the supporting foot and the center of mass of the whole robot. However, even with this approximation, the inverted pendulum has the vast possibilities of moving pattern which are not good for walking. To pick up the suitable motion for walking we constrain the center of mass to move on a plane specified by equation,

$$z = k_x x + k_y y + z_c, \tag{1}$$

where (x, y, z) is the position of the mass with respect to the supporting point, $(k_x, k_y, -1)$ specifies the normal vector of the constraint plane and z_c is the z intersection. In the case of the walk on a flat floor, the constraint plane is horizontal and the height of the center of mass is kept constant.

We obtain the following dynamics of the pendulum under the constraints

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} u_p, , \tag{2}$$

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} u_r.. \tag{3}$$

We call this dynamics as the 3D Linear Inverted Pendulum Mode (3D-LIPM)[6]. The only parameter which governs 3D-LIPM is z_c, i.e., the z intersection of the constraint plane and the inclination of the plane never affects the horizontal motion. Since equations (2) and (3) are linear and independent, we can easily obtain the closed form solutions which can be directly used for a dynamic biped walking. Particularly, when the input torques of the supporting point are zero ($u_r = u_p = 0$), we obtain hyperbolic trajectories on the constraint plane.

The reference joint angles and speeds are calculated by the inverse kinematics so that the position and the velocity of the feet with respect to the center of mass follow the 3D-LIPM.

Stabilizer for walking The walking pattern generator can give walking motions that should be dynamically stable, but a feedback stabilizer for the biped walking is still necessary to realize the walking to cope with possible disturbances. Especially, the stabilizer is essential for the robot that has a soft spring-damper mechanism on its feet like HRP-1S and HRP-2P. We have developed a stabilizer that consists of a body inclination control, ZMP dumping control and foot adjusting control for the purpose[17].

4.5 Experiments

When the software has been examined on the simulator, it can be applied to a hardware platform as it is. That is, the binary compatibility between the software on the simulator and that on the robot is guaranteed. After the required software has been downloaded to a robot, the behavior of the robot can be initiated by sending a script to the robot. The internal states of the robot can be observed on the operator's console via a wireless LAN.

Figure11 shows the snapshots of walking HRP-2P. HRP-2P can also get up from the floor[8]. HRP-2P is the first human-size humanoid robot that can get up from the floor. See Fig.12. HRP-2P can also lie down to the floor[8].

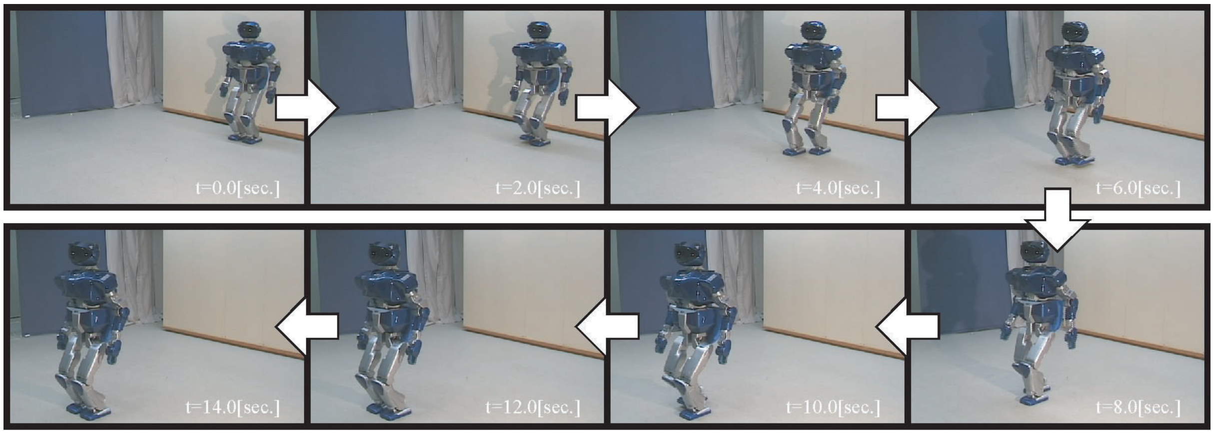

Fig. 11. HRP-2P walking

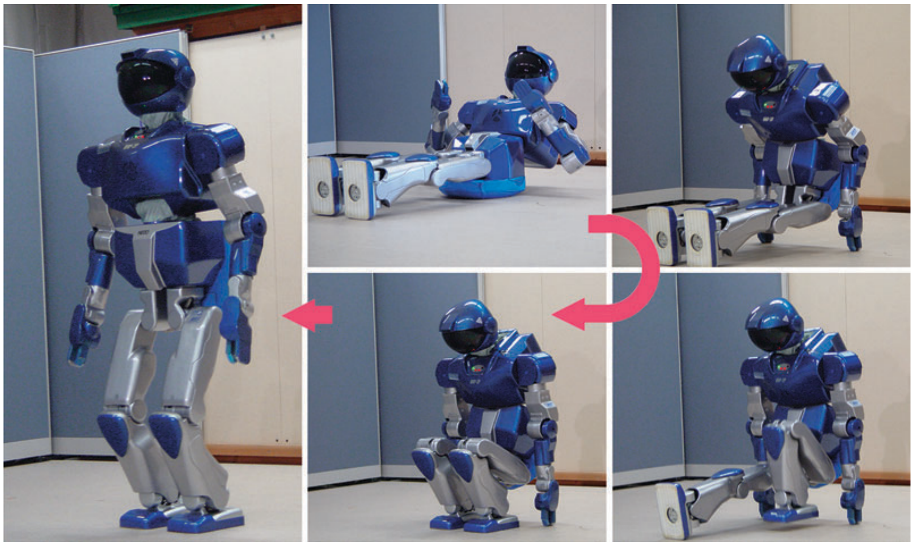

Fig. 12. HRP-2P getting up from the floor

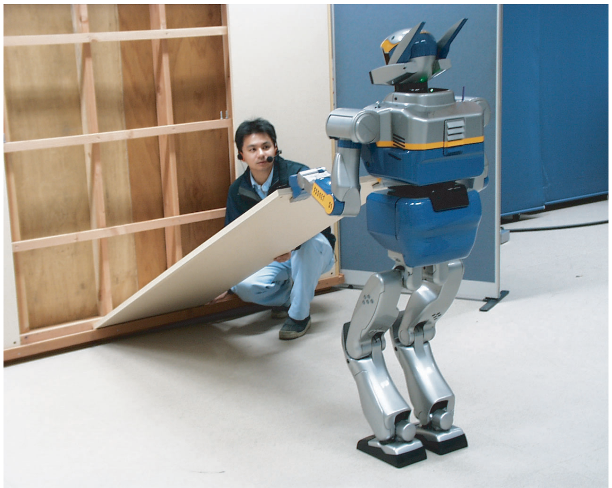

Fig. 13. HRP-2 assemblies a cottage with a human.

An example of applications of HRP-2 driven by a controller of OpenHRP is the cooperative tasks with a human. Figure 13 shows an assembly of a panel onto a cottage by a human and HRP-2. Besides these motions, falling motion control[1], running [11] and dancing motions[12] have been investigated on the simulator of OpenHRP, and their experimental study is going on.

5 Conclusions

HRP has developed the humanoid robotics platform consisting of the hardware and the software. We expect that HRP-2 and OpenHRP will be able to serve as the research platform for humanoid robotics in the near future thanks to their open architecture feature.

Though various robotic technologies are available in the academic community, but a very few of them have been implemented on humanoid robots so far. The main reason for the situation is that there has been a little chance to use a humanoid robot with the fundamental software. We believe that HRP-2 and OpenHRP can provide us more chance to integrate robotic technologies on humanoid robots.

Acknowledgements

This research was supported by the Humanoid Robotics Project of the Ministry of Economy, Trade and Industry (METI) and New Energy and Industrial Technology Development Organization (NEDO). The authors thank them for their support. The authors also thank Manufacturing Science and Technology Center (MSTC) for their help.

References

1. K.Fujiwara, F.Kanehiro, S.Kajita, K.Kaneko, K.Yokoi, and H.Hirukawa, UKEMI: Falling Motion Control to Minimize Damage to Biped Humanoid Robot, Proc. of IROS 2002.
2. H.Inoue, S.Tachi, K.Tanie, K.Yokoi, S.Hirai, H.Hirukawa, K.Hirai, S.Nakayama, K.Sawada, T.Nishiyama, O.Miki, T.Itoko, H.Inaba, and M.Sudo, HRP:Humanoid Robotics Project of MITI., In Proc. of the First IEEE-RAS Int. Conf. on Humanoid Robots, 2000.
3. H.Inoue, S.Tachi, Y.Nakamura, N.Ohyu, S.Hirai, K.Tanie, K.Yokoi and H.Hirukawa, Overview of Humanoid Robotics Project of METI, In Proc. of the 32nd ISR, 2001.
4. H.Inoue and H.Hirukawa, Explorations of Humanoid Robot Applications, Proc. Second IEEE-RAS Int. Conf. on Humanoid Robots, 2001.
5. Y.Ishiwata and T.Matsui, Development of Linux which has Advanced Real-Time Processing Function, In Proc. 16th Annual Conference of Robotics Society of Japan, pp. 355-356, 1998.
6. S.Kajita, F.Kanehiro, K.Kaneko, K.Yokoi, and H.Hirukawa, The 3D Linear Inverted Pendulum Model : A simple smodeling for a biped walking pattern generation, Proc. of the 2001 IROS, pp.239-246, 2001.
7. F.Kanehiro, K.Fujiwara, S.Kajita, K.Yokoi, K.Kaneko, and H.Hirukawa, Open Architecture Humanoid Robot Platform, Proc. IEEE Int. Conf. on Robotics and Automation, 2002.
8. F.Kanehiro, K.Kaneko, K.Fujiwara, K.Harada, S.Kajita, K.Yokoi, H.Hirukawa, K.Akachi and T.Isozumi, The First Humanoid Robot that has the Same Size as a Human and that can Lie down and Get up, IEEE ICRA 2003, to appear.
9. K.Kaneko, S.Kajita, F.Kanehiro, K.Yokoi, K.Fujiwara, H.Hirukawa, T.Kawasaki, M.Hirata, and T.Isozumi, Design of Advanced Leg Module for Humanoid Robotics Project of METI, Proc. IEEE Int. Conference on Robotics and Automation, 2002.
10. K.Kaneko, F.Kanehiro, S.Kajita, K.Yokoyama, K.Akachi, T.Kawasaki, S.Ota, T.Isozumi, Design of Prototype Humanoid Robotics Platform for HRP, Proc. IEEE IROS, 2002.
11. T.Nagasaki, S.Kajita, K.Yokoi, K.Kaneko and K.Tanie, Running Pattern Generation and Its Evaluation Using A Realistic Humanoid Model, Proc. IEEE ICRA 2003, to appear.
12. S.Nakaoka, A.Nakazawa, K.Yokoi, H.Hirukawa and K.Ikeuchi, Generating Whole Body Motions for a Biped Humanoid Robot from Captured Human Dances, Proc. IEEE ICRA 2003, to appear.
13. T.Oka, K.Takeda, M.Inaba and H.INOUE, Designing Asynchronous Parallel Process Networks for Desirable Autonomous Robot Behavior, Proc. IEEE IROS'96 pp.178–185, 1996.
14. O.Stasse and Y.Kuniyoshi, PredN: Achieving Efficiency and Code Re-Usability in a Programming System for Complex Robotic Applications, pp.81–87, Proc. IEEE ICRA, 2000.
15. Y. Sumi, Y. Kawai, T. Yoshimi, and F. Tomita, 3D Object Recognition in Cluttered Environments by Segment-Based Stereo Vision, International Journal of Computer Vision, 46, 1, pp.5-23, 2002.
16. K.Yamane and Y.Nakamura, Dynamics Computation of Structure-Varying Kinematic Chains for Motion Synthesis of Humanoid, In Proc. of the 1999 IEEE International Conference on Robotics and Automation, pp. 714-721, 1999.
17. K. Yokoi, F. Kanehiro, K. Kaneko, K. Fujiwara S. Kajita, and H. Hirukawa, A Honda Humanoid Robot Controlled by AIST Software, In Proc. of the IEEE-RAS Int. Conf. Humanoid Robots, 2001.
18. http://www.omg.org. Object Management Group.
19. http://luz.cs.nmt.edu/˜rtlinux, *RT-Linux*, V. Yodaiken and M.Barabanov.