We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

6

Development of a CORBA-based Humanoid Robot and its Applications

Yasuo Nasu[1], Genci Capi[2], Hanafiah Yussof[3]
Mitsuhiro Yamano[1] and Masahiro Ohka[3]
[1]*Faculty of Engineering, Yamagata University,* [2] *Faculty of Engineering, Toyama University,* [3]*Graduate School of Information Science, Nagoya University*
Japan

1. Introduction

Recently, the research on humanoid robots has attracted many researchers. The research spans from stability and optimal control, gait generation, human-robot and robot-robot communication. In addition, humanoid robots have been also used to understand better human motion. Among humanoid robot prototypes the most well known is Honda humanoid robot (Hirai et. al., 1998). This robot has the ability to move forward and backward, sideways to the right or the left, as well as diagonally. In addition, the robot can turn in any direction, walk up and down stairs continuously. Other example is the 35 dof (degrees of freedom) Saika humanoid robot (Inaba et al. 1998). This robot can perform a reach-and-grasp motion through coordinating legs and arms. The key idea of the system architecture of this robot is a remote brained approach. In addition, the Waseda humanoid robot group has also developed an anthropomorphic dynamic biped walking robot adapting to the humans' living floor (Takanishi et. al., 1990). Fujitsu also has developed a commercial 25 dof miniature humanoid robot, named HOAP-1, for research purposes. Weighing 6 kg and standing 0.48 m tall, the light and compact HOAP-1 and accompanying simulation software can be used for developing motion control algorithms in such areas as two-legged walking, as well as in research on human-to-robot communication interfaces.
In our robotics laboratory at Yamagata University, we initialized the humanoid robot project. The goal of this project is to contribute to the research on humanoid robots. For this reason, we developed an anthropomorphic biped humanoid robot called Bonten-Maru. During the humanoid robot design, we tried to mimic as much as possible the human characteristics, from the viewpoints of links dimensions, body structure, as well as the number and configuration of the degrees of freedom. The high number of dofs helps the humanoid robot to realize complex motions in even and uneven terrains, like walking, going up and down stairs, crawling, etc. In this chapter, we present the development of Common Object Request Broker Architecture (CORBA) based humanoid robot control systems. Consequently, this chapter explains the application of real time generation of humanoid robot optimal gait by using soft computing techniques, and also teleoperation systems and its applications. Simulation and experimental results of the proposed system in

Source: Humanoid Robots: Human-like Machines, Book edited by: Matthias Hackel
ISBN 978-3-902613-07-3, pp. 642, Itech, Vienna, Austria, June 2007

each research theme utilizing the humanoid robot Bonten-Maru are presented which reveals good performance of the robot control system.
This chapter is organized in the following sequence: Section 2 explains the motivations of this research. In Section 3, we briefly explain the development of research prototype humanoid robot *Bonten-Maru I* and *Bonten-Maru II* in term of hardware configurations and control systems. Next, Section 4 explains the CORBA-based Humanoid Robot Control Architecture (HRCA). This section includes concept, algorithm and experimental results of the HRCA. Next in Section 5, we present an optimal locomotion strategy using Genetic Algorithm (GA) and Neural Networks (NN) for bipedal humanoid robot. In this section we presents a GA gait synthesis method for biped robots during walking based on Consumed Energy (CE) and Torque Change (TC), and also application of Radial Basis Function Neural Networks (RBFNN). In Section 6, we explain development of teleoperation systems via internet and user-friendly Virtual Reality (VR) user interface. Consequently, experimental results are presented in each section. Finally, summary and conclusions are presented in Section 7, followed by future work in Section 8.

2. Motivations

At present, many robots are developed for particular industrial, entertainment and service applications. However, these robots cannot be applied for other application. Especially when different programming languages are used for other applications, the cooperation between different systems is needed and the programs must be converted to be the same. Furthermore, the application of tele-operated system in robotics field is highly demanded like nowadays telecommunication technology. Therefore, in order to build open robot control platforms in humanoid robot control system, CORBA has been proposed.
The used of CORBA for the humanoid robots has open new dimension in robotic research, for example in teleoperation operations via internet. The system can be apply not only for the humanoid robots systems, but also for other fields like tele-medical, industrial robots, service and security, and also in aerospace project. However, the accuracy issue and time delay problem are the main factor to be consider in order to make the project successful in common architecture applications. Therefore, we considered programming language built in network program like Java and Perl in the robot control programming which commonly used C or C++. The management in cooperation between network programs and robot control programs are expected to reduce the time delay and increase the accuracy of certain motion in the robot task. In addition, the design of robot hardware and control systems is also considered to obtain reliable and accurate motions in real time applications.
Eventually, we were one of the first humanoid robot groups, that proposed a humanoid robot control platform based on CORBA (Takeda et al., 2001). One of our main objectives is to make the control platform open for other researchers to test their results and also to conduct collaborative research. By using a CORBA based control platform, it is easy to add modules developed in different programming languages. In addition, the control of the humanoid robot is made in a distributed environment. Therefore, various humanoid robots in the world can share their own modules with each other via the internet.
Another direction of our research is the real time generation of humanoid robot optimal gait by using soft computing techniques (Capi et al. 2003). Genetic Algorithms (GA) was employed to minimize the energy for humanoid robot gait. For a real time gait generation, we used Radial Basis Function Neural Networks (RBFNN), which are trained based on

Genetic Algorithm (GA) data. Until now the walking and going up-stairs modules are created. The main goal is to create an autonomous humanoid robot that can operate in different environments. Based on the information received by the eye system, the appropriate module will be simulated to generate the optimal motion.
Control of humanoid robot in a long distance was also another research issue. In our research, the main objective is to create various sophisticated motions and new application fields for humanoid robots. For this reason, we considered accident site operations which are often unknown environments. In our work, we used a teleoperation system to control the humanoid robot via the internet. We carried out long distance experiments on the teleoperation system of the humanoid robot between Deakin University (Australia) and Yamagata University (Japan) (Nasu et al., 2003). Additionally, we have developed a user-friendly Virtual Reality (VR) user interface that is composed of ultrasonic 3D mouse system and a Head Mounted Display (HMD) (Kaneko et al., 2003). The real time experiments were conducted using the Bonten-Maru humanoid robot.

3. Development of Research Prototype Humanoid Robot

We have developed a research prototype humanoid robot system known as "Bonten-Maru". The earliest model was the 23 dof Bonten-Maru I. Next, we developed an advanced version called Bonten-Maru II which consists of 21 dof. The Bonten-Maru humanoid robot series are one of few research humanoid robots in the world that can be utilized in various aspects of studies.

3.1 Humanoid Robot Bonten-Maru I

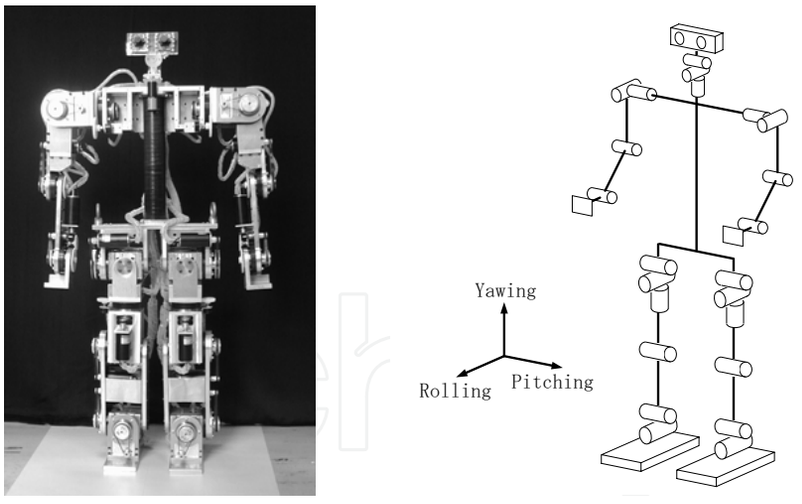


Figure 1. The Bonten-Maru I humanoid robot and its distribution of dofs

The Bonten-Maru I humanoid robot is consists of 23 dof. The appearance and distribution of dofs are shown in Fig. 1. Bonten-Maru I is 1.2 m high, weight about 35 kg. The properties of Bonten-Maru I is shown in Table 1, meanwhile Table 2 shows the configurations of dofs. The robot's each leg has six degree of freedom and is composed by three segments: upper leg,

lower leg and the foot. The hip is a ball-joint, permitting three dof; the knee joint one dof; the ankle is a double-axis design, permitting two. The shoulder has two dof, the elbow and wrist one dof. The distribution of dof is similar with the dof in human limbs. A DC servometer actuates each joint. The DC servomotors act across the three joints of the head, where is mounted the eye system, enabling a total of three dof.
Potentiometers are used to obtain the position and velocity of every link, interfaced to the computer via RIF-01 (ADC). The power is supplied to each joint b timing belt and harmonic drive reduction system. The gyro unit is connected to the PC by RS-232C interface. The head unit has 2 CCD cameras (542 x 492 pixels, Monochrome). The CCD camera units are connected to PC by video capture board. The control platform is based on CORBA. This allows an easy updating and addition of new modules. A Caleron based microcomputer (PC/AT compatible) is used to control the system. The OS are Linux 2.0.36 and FreeBSD2.2.8R.

	Body	*Lower leg*	*Upper leg*	*Lower leg + foot*
Moment of inertia [kg m^{2}]	0.19	0.014	0.002	0.017
CoM from lower joint [m]	0.3	0.09	0.1	0.136
CoM from upper joint [m]	0.0	0.11	0.104	0.136
Length [m]	0.3	0.2	0.204	0.284
Mass of the link [kg]	12	2.93	3.89	4.09

CoM: Center of Mass

Table 1. Properties of Bonten-Maru I

JOINT		DIRECTION	DOF
Leg	Hip	Rolling, Pitching	2
	Thigh	Yawing	1
	Knee	Pitching	1
	Ankle	Rolling, Pitching ,	2
Arm	Shoulder	Rolling, Pitching	2
	Elbow	Yawing	1
	Wrist	Pitching	1
Head	Neck	Rolling, Pitching, Yawing	3

Table 2. Configurations of dofs in Bonten-Maru I

3.2 Humanoid Robot Bonten-Maru II

Bonten-Maru II is an anthropomorphic prototype humanoid robot. This robot is 1.25 m tall, weighting about 32.5 kg. Fig. 2 shows the appearance of Bonten-Maru II and distribution of dofs. The robot has a total of 21 dofs: six for each leg, three for each arm, one for the waist, and two for the head. Configurations of dofs at each joint and joint rotation range are shown

in Table 3. The link dimensions are determined such that to mimic as much as possible the humans. The high number of dofs and configuration of joints that closely resemble those of humans provide Bonten-Maru II with the possibility of realizing complex trajectories to attain human-like motion. Each joint is driven by a DC servomotor with a rotary encoder and a harmonic drive-reduction system, and is controlled by a PC with the Linux OS. Under each foot are four force sensors, and the arms are equipped with six-axis force sensor. The head unit has two CCD cameras (542 x 492 pixels, Monochrome), which are connected to PC by video capture board. A Celeron based microcomputer (PC/AT compatible) is used to control the system.

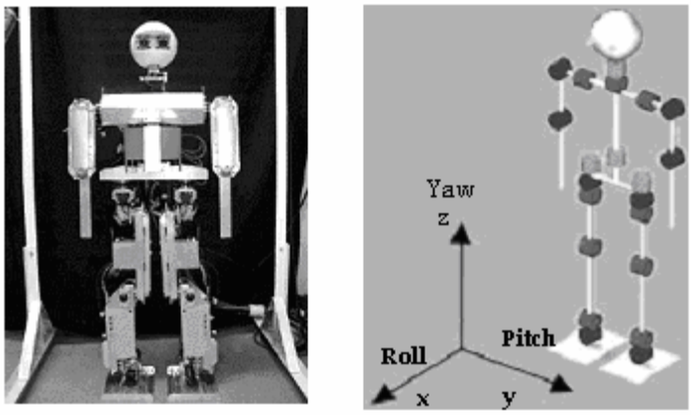


Figure 2. The Bonten-Maru II humanoid robot and its distribution of dofs

Axis	Quantity of dofs	Range of joint rotation angle (deg)
Neck (roll and pitch)	2	-90 ~ 90
Shoulder (pitch) right & left	2	-180 ~ 120
Shoulder (roll) right /left	2	-135 ~ 30/-30 ~ 135
Elbow (roll) right /left	2	0 ~ 90/0 ~ -90
Waist (yaw)	1	-45 ~ 45
Hip (yaw) right /left	2	-90 ~ 60/-60 ~ 90
Hip (roll) right/left	2	-90 ~ 25/-25 ~ 90
Hip (pitch) right & left	2	-130 ~ 45
Knee (pitch) right & left	2	-20 ~150
Ankle (pitch) right & left	2	-90 ~ 60
Ankle (roll) right/left	2	-20 ~ 90/-90 ~ 20

Table 3. Configurations of dofs and joint rotation range in Bonten-Maru II

4. CORBA-Based Humanoid Robot Control Architecture (HRCA)

The development of an efficient humanoid robot control system required the control modules to be developed independently and able to integrate easily to the system. Commonly, the control modules developed by many researchers are apart from OS and

programming languages must be connected to the internet directly for the common use in the worldwide. For this reason, CORBA (Moubray et al., 1997) is a good platform for humanoid control system architecture. CORBA is a specification of message exchange among objects, which is specified by Open Management Group (OMG). CORBA has attracted many researchers. Vinoski (Vinoski, 1997) suggested the effectiveness of CORBA for heterogeneous systems. Pancerella (Pancerella et al., 1996) and Whiteside (Whiteside et al., 1997) have implemented a CORBA based distributed object software system for the Sandia Agile Manufacturing Testbed. Harrison (Harrison et al., 1997) has developed a real-time CORBA Event Service as a part of the TAO project at Washington University. CORBA is a useful distributed application platform, which can make the cooperation among distributed applications very easily. Also, CORBA is independent on a specific programming language or OS. Thus, it is able to communicate among objects developed in different programming languages and OS.

4.1 HRCA Concept

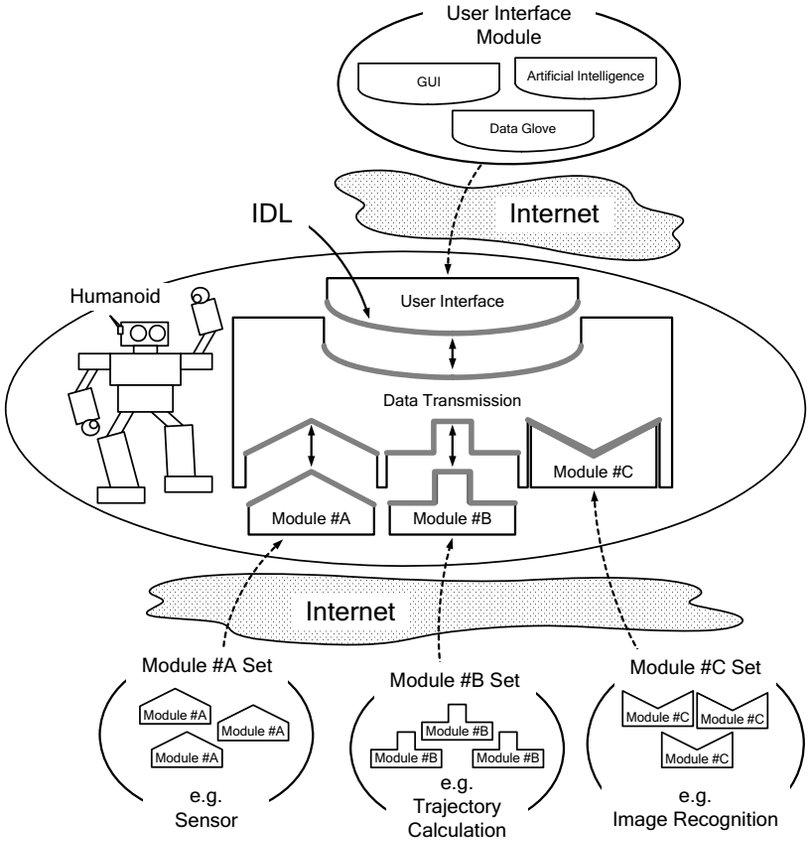


Figure 3. Concept of the proposed HRCA

We have proposed the HRCA to integrate the desired modules which developed by many researchers individually to control the motion of humanoid robot via the internet. The HRCA can share many modules among many users or researchers from remote distance through any computer by the internet communication. Figure 3 shows a basic concept of the proposed HRCA. The HRCA design is based on the Unified Modeling Language (UML), which includes the Use Case Diagram (UCD) and Class Diagram (CD).

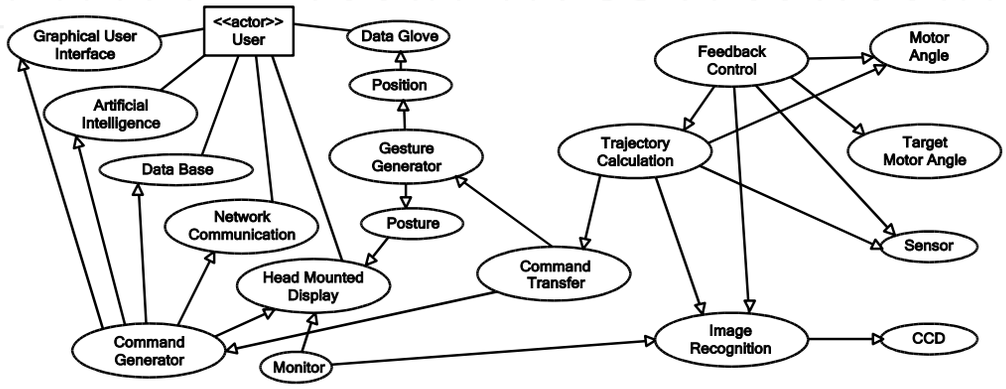


Figure 4. Relationships among the HRCA modules

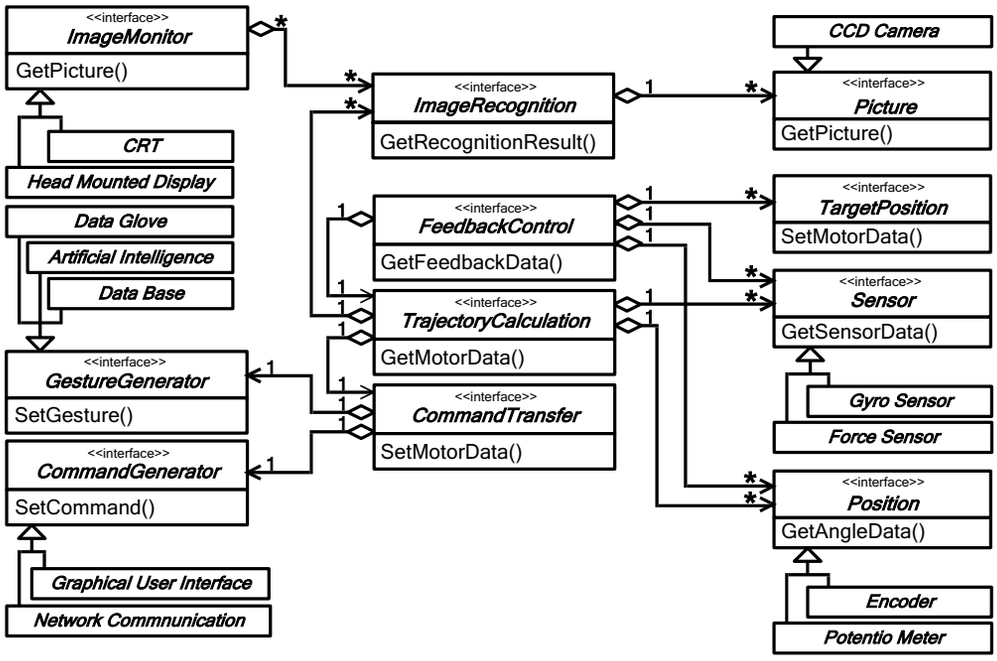
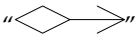


Figure 5. Class diagram of HRCA

The UML is used to define CORBA servers, clients, and it's IDL. Booch (Booch et al., 1999), Fowler (Fowler et al., 1997), and Open Manegement Group (OMG) proposed the UML for

the design of the object-oriented programming. The HRCA modules are designed by using the UCD. The relationships among the HRCA modules are shown in Fig. 4. The HRCA is very complex, but in this figure we only show the highest level of the system. Each circle presents a Use Case (UC) and each arrow shows relationships among them. Eventually, when we design the CD, each UC is defined as a class.
The CD is shown in Fig. 5. There are many classes in each square, but in this study, we use only the interface class, because the IDL defines only the object's interface. In the CD, each square presents a class icon. Inside the square the stereotype, name, and method of the class are written. The symbol "  " presents an association among classes.
The number written on the both ends of the symbols show how many classes are used. The symbol "*" shows that the class number is not limited. Finally, each class in Fig. 5 is implemented as HRCA modules, which correspond to CORBA servers and client. The IDL of each HRCA modules are obtained from CD, and convert to a programming language source code by IDL compiler.

4.2 Proposed HRCA Modules

The HRCA model is shown in Fig. 6. This figure presents some algorithms and devices, which can be implemented as HRCA modules. The HRCA is able to use these algorithms and devices by selecting the appropriate CORBA servers. Until now, we have implemented the following modules: DTCM, MCM, JTM, GSM, JAM, FCM, and UIM, which are shown in Fig. 7. Each module corresponds to "Data Transmission", "Target Position", "Angle Trajectory Calculation", "Sensor", "Position", "Feedback Control", "Command Generator", respectively, which are shown in Fig. 6. To implement CORBA servers and client, the Inter-Language Unification (ILU) is used, which has proposed by XEROX PARC. ILU supports many programming languages, such as C++, ANSI C, Python, Java, Common Lisp, Modula-3, Guile Scheme, and Perl 5. In our research, we used only C language for implement HRCA. But in the future, we would like to implement some HRCA modules using other languages.
In our HRCA, the DTCM controls the data flow of the modules. The DTCM communicates with MCM, JTM, GSM, JAM, and FCM by using their functions. But DTCM communicates with UIM by its own function. The data flow model is also shown in Fig. 7. Until now, the UIM is very simple, which is able to command "WALK", "OBJECT_OVERCOMING", and "GYRO_TEST" only.
The MCM controls the joint motors of the humanoid robot. The model of MCM and the IDL between MCM and DTCM, and are shown in Fig. 8 and Fig. 9, respectively. In Fig. 9, the MCM provides two functions, "SetMotorData()" and "SetMotorFeedbackData()". "SetMotorData()" is a function for the data input of the joint trajectories. "ROBOT_DEGREE_DATA" data type includes the time unit data and the joint trajectory data, which are named as "time_unit" and "degree_data", respectively. "SetMotorFeedbackData()" is a function for the feedback data input from FCM. "MOTOR_FEEDBACK_DEGREE_DATA" data type includes the joint feedback data, which is named as "feedback_degree_data". Using these arguments of the "SetMotorData()" and "SetMotorFeedbackData()", the MCM controls each joint motor. In addition, in this study, we used multi-threaded implementation for motor control routine because of the time delay, which is caused by controlling each joint motor, sequentially. Using multi-threaded implementation, the motors are controlled in parallel and the time delay is reduced.

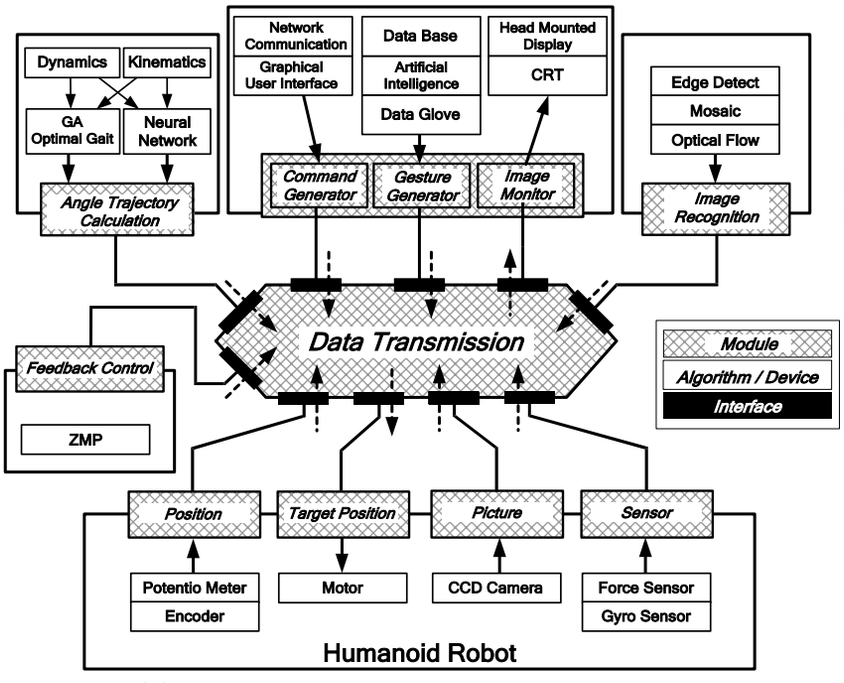


Figure 6. HRCA model

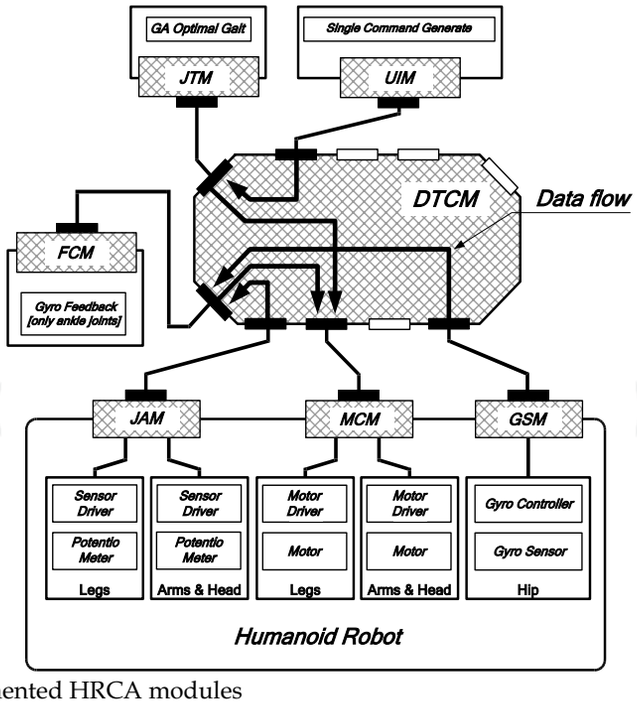


Figure 7. Implemented HRCA modules

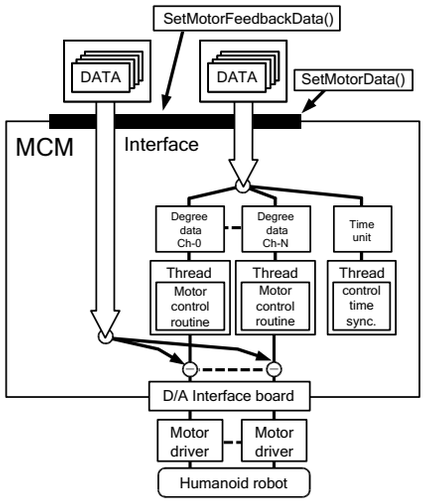


Figure 8. The model of Motor Control Module (MCM)

```
module MCM{

   // Define Array
   const long MOTOR = 25;                  //max 25 motors
   const long TIMES = 100;                 //max 100 set data

   typedef double             DEGREE[TIMES];

   typedef struct MotorDegreeData{
      DEGREE degrees;
   }MOTOR_DEGREE_DATA[MOTOR];

   typedef struct RobotDegreeData{
      long time_unit;
      MOTOR_DEGREE_DATA degree_data;
   }ROBOT_DEGREE_DATA;

   // For Robot Feedback
   typedef double    FEEDBACK_DEGREE[MOTOR];

   typedef struct MotorFeedbackDegreeData{
      FEEDBACK_DEGREE feedback_degree_data;
   }MOTOR_FEEDBACK_DEGREE_DATA;

   //Motor Interface
   interface MotorInterface{

      // Motor Angle Values Setting
      short
      SetMotorData(
            in ROBOT_DEGREE_DATA degree_order,
            in long data_size);

      short
      SetMotorFeedbackData(
            in MOTOR_FEEDBACK_DEGREE_DATA
                  degree_feedback_order );

   };
};
```

Figure 9. MCM IDL and MCM module

The IDL of other modules are developed and implemented same as MCM. The JTM provides the joint trajectories data to DTCM. The joint trajectories data is defined same as the input data type of MCM. The joint trajectory data are calculated by a genetic algorithm program and are used in a data base. These data are provided from JTM to DTCM. The GSM provides the angle, angular velocity, and angular acceleration data of gyro sensor to DTCM. The JAM provides the joint angle data of humanoid robot to DTCM. The JAM is used for reading and recording the actual joint trajectory data of the humanoid robot by using multi-threaded implementation. The FCM provides the feedback joint angle data to MCM via DTCM. The FCM obtains the joint angle data from JAM and GSM via DTCM, which balance the humanoid robot. We use only gyro sensor data for ankle joints control, but in the future, we would like to add another sensor data for precise feedback control.

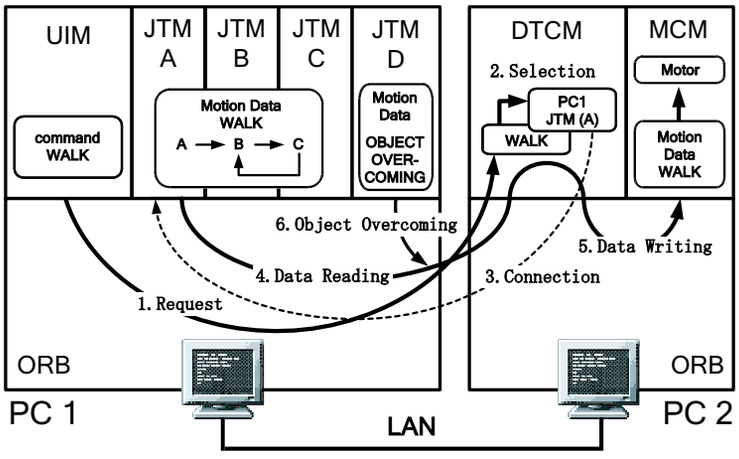


Figure 10. Data transfer flow in experimental system

4.3 Experiment and Result

Using the humanoid robot, we have carried out an experiment to show the effectiveness of the proposed HRCA. In this test, we are concentrated in the development of the humanoid robot control architecture, not in the control scheme itself or the robot response.

In order to measure the utilization of the proposed HRCA, the motion mode change tests are conducted on the Bonten-Maru I humanoid robot. The HRCA is applied to change the motion related to JTM as shown in Table 4. The static walking motion is divided into 3 parts in order to reuse the motion modules efficiently. The JTM (A, B, C, and D) and the UIM are running in PC1, the MCM and DTCM are running in PC2. The PC1 and PC2 are connected via LAN. The data transfer flow is shown in Fig. 10. The module changing procedure to control the motion of humanoid robot is explained as follows:

1. Request: The UIM sends an order sequence to DTCM (in this experiment it sends the "WALK" request);
2. JTM Selection: After receiving the "WALK" request from the UIM, the DTCM selects a JTM;
3. Connection: The DTCM is connected with JTM;
4. Data Reading: The DTCM reads the "WALK" data from JTM(A);

5. Data Writing: The DTCM transfers the data of JTM(A) to MCM and the MCM executes the data. When the humanoid robot is walking, the walking movement starts by JTM(A) and the normal walking is carried out by repeating in the round robin order JTM(B) and JTM(C);
6. Object Overcoming: The DTCM changes the JTM from "WALK" to "OBJECT_OVERCOMING", connects to JTM(D), and reads the "OBJECT_OVERCOMING" data from JTM(D). Then, the data is transferred to MCM, which moves the motor.

Figure 11 shows the joint angle trajectories, for right joint angles $(\theta_{R2}, \theta_{R3}, \text{and}\, \theta_{R6})$ and for left joint angles $(\theta_{L2}, \theta_{L3}, \text{and}\, \theta_{L6})$, during 4 steps JTM(A, B, C, and D). Figure 14 shows the continuous walking motion of humanoid robot by different modules of our proposed HRCA. The humanoid robot walks in smooth and static manner. Ideally, the time lag should be as short as possible at every step change. However, during the experiment, we measured that the time lag at every step change is about 200 milliseconds. But this time lag did not influence on the walking motion of the humanoid robot during every step because the humanoid robot walks in static condition. This experimental result shows that the proposed HRCA is able to control the static motion of humanoid robot accurately by anging the response of JTM. Figure 12 shows the continuous walking motion of humanoid robot by different modules of the proposed HRCA. The humanoid robot walks in smooth and static manner. This experiment result shows that the proposed HRCA is able to control the static motion of humanoid robot accurately by changing the respond of JTM.

1	JTM(A)	From stand position to right leg swing
2	JTM(B)	From right leg swing to left leg swing
3	JTM(C)	From left leg swing to right leg swing
4	JTM(D)	Object overcoming by using left leg

Table 4. Joint Trajectory Module (JTM) motions

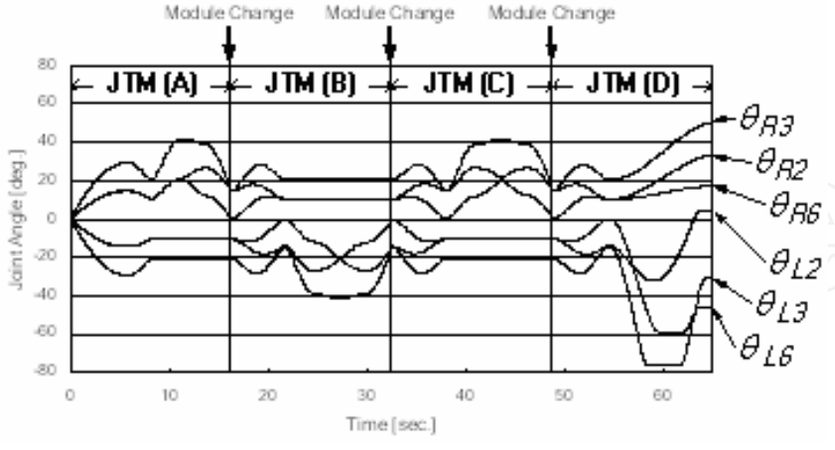


Figure 11. Joint angle trajectories

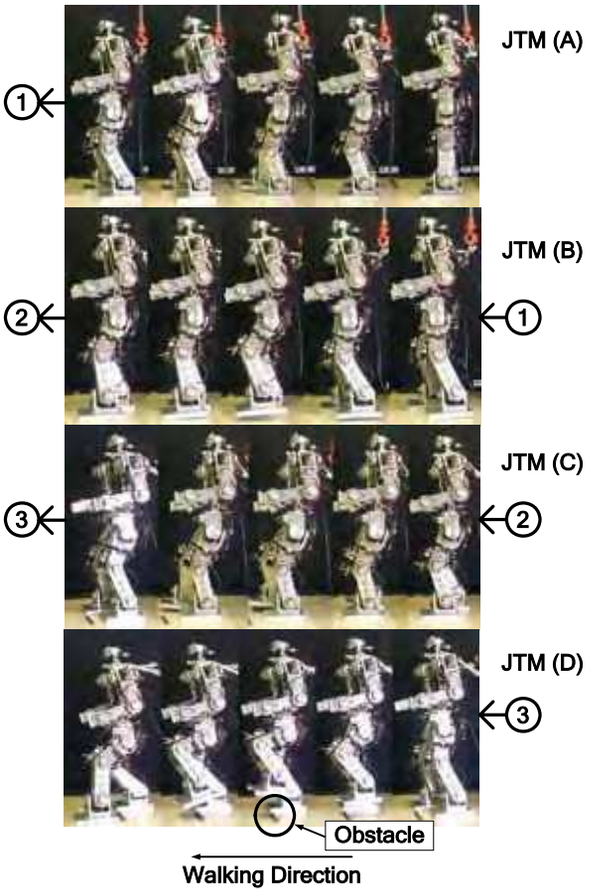


Figure 12. Continuous walking motion

5. Optimal gait strategy

We considered minimum Consumed Energy (CE) as criterion for humanoid robot gait generation, because autonomous humanoid robots make difficult to use external power supply. In order to have a long operation time when a battery actuates the motors, the energy must be minimized. From the viewpoint of energy consumption, one factor that has a great influence is the gait synthesis. In most of the previous papers related to biped robots (Vukobratovic et al., 1990) (Takanishi et al. 1990), the angle trajectories of the leg part are prescribed based on data taken from humans. The motion of upper body is calculated in order to have the ZMP inside the sole region. Some effort has been placed to analyze the effect of gait synthesis on consumed energy. In (Roussel et al., 1998) and (Silva et al., 1999) the minimum consumed energy gait synthesis during walking is treated. The body mass is considered concentrated on the hip of the biped robot (Roussel et al., 1998). In (Silva et al., 1999), the body link is restricted to the vertical position, the body forward velocity is considered to be constant and the tip motion of the swing leg is constrained to follow

sinusoidal functions. The effect of the walking velocity and step length on the consumed energy is treated in (Channon et al., 1996). A variation technique is used to minimize the cost function. However, in all these approaches related with optimal gait of biped robots, the stability and the real time implementation are not considered.

In this section, we describe a Genetic Algorithm (GA) gait synthesis method for biped robots during walking based on Consumed Energy (CE) and Torque Change (TC). The proposed method can easily be applied for other tasks like overcoming obstacles, going down-stairs, etc. The final aim of this research is to create modules for many tasks considered to be performed by Bonten-Maru I humanoid robot. Based on the information received by eye system, the appropriate module will be simulated to generate the optimal motion.

When solving for optimal gaits, some constrains must be considered. In our work, the most important constraint is the stability, which is verified through the ZMP concept. GA makes easy handling the constraints by using the penalty function vector, which transforms a constrained problem to an unconstrained one. GA has also been known to be robust for search and optimization problems (Channon et al., 1996), It has been used to solve difficult problems with objective functions that do not possess properties such as continuity, differentiability, etc. For a real time implementation, in (Roussel et al., 1998), the authors suggest to create in the future a database of pre-computed optimal gaits. This can generate the angle trajectories only for the step lengths and step times, which are included in the database. In order to cover all the interval of the pre-computed optimal gaits, we consider teaching a Radial Basis Function Neural Network (RBFNN) based on the GA results. In this section we present the results where input variables are the step length and step time. Simulations show good results generated by RBFNN in a very short time.

5.1 Genetic Algorithm (GA)

GA is a search algorithm based on the mechanics of natural selection and population genetics. The search mechanism is based on the interaction between individuals and the natural environment. GA comprises a set of individuals (the population) and a set of biologically inspired operators (the genetic operators). The individuals have genes, which are the potential solutions for the problem. The genetic operators are crossover and mutation. GA generates a sequence of populations by using genetic operators among individuals. Only the most suited individuals in a population can survive and generate offspring, thus transmitting their biological heredity to the new generation. The main steps are shown below:

1. Supply a population P_0 of N individuals and respective function values;
2. $i \leftarrow 1$;
3. $P_i' \leftarrow$ selection_function ($P_i - 1$);
4. $P_i \leftarrow$ reproduction_function (P_i');
5. Evaluate (P_i);
6. $i \leftarrow i+1$;
7. Repeat step 3 until termination.

5.2 Biped Model

The arms of the humanoid robot will be fixed on the chest during performing motion. Therefore, it can be considered as a five-link biped robot in the saggital plane, as shown in Fig. 13. The motion of the biped robot is considered to be composed from a single support

phase and an instantaneous double support phase. The friction force between the robot's feet and the ground is considered to be great enough to prevent sliding. During the single support phase, the ZMP must be within the sole length, so the contact between the foot and the ground will remain. To have a stable periodic motion, when the swing foot touches the ground, the ZMP must jump in its sole. This is realized by accelerating the body link. To have an easier relative motion of the body, the coordinate system from the ankle joint of the supporting leg is moved transitionally to the waist of the robot ($O_1X_1Z_1$). Referring to the new coordinate system, the ZMP position is written as follows:

$$\overline{X}_{ZMP} = \frac{\sum_{i=1}^{5} m_i(\ddot{\overline{z}}_i + \ddot{z}_w + g_z)\overline{x}_i - \sum_{i=1}^{5} m_i(\ddot{\overline{x}}_i + \ddot{x}_w)(\overline{z}_i + z_w)}{\sum_{i=1}^{5} m_i(\ddot{\overline{z}}_i + \ddot{z}_w + g_z)}, \quad (1)$$

where m_i is mass of the particle "*i*", x_w and z_w are the coordinates of the waist with respect to the coordinate system at the ankle joint of supporting leg, $\overline{x}_i$ and $\overline{z}_i$ are the coordinates of the mass particle "*i*" with respect to the $O_1X_1Z_1$, $\ddot{\overline{x}}_i$ and $\ddot{\overline{z}}_i$ are the acceleration of the mass particle "*i*" with respect to the $O_1X_1Z_1$ coordinate system. Based on the formula (1), if the position, $\overline{x}_i, \overline{z}_i$, and acceleration, $\ddot{\overline{x}}_i, \ddot{\overline{z}}_i$, of the leg part (i=1,2,4,5), the body angle, θ_3, and body angular velocity, $\dot{\theta}_3$, are known, then because $\ddot{\overline{x}}_3, \ddot{\overline{z}}_3$ are functions of $l_3, \theta_3, \dot{\theta}_3, \ddot{\theta}_3$, it is easy to calculate the body angular acceleration based on the ZMP position. Let $_{(0)}$ and $_{(f)}$ be the indexes at the beginning and at the end of the step, respectively. At the beginning of the step, $\ddot{\theta}_{30}$, causes the ZMP to be in the position ZMP_{jump}. At the end of the step, the angular acceleration, $\ddot{\theta}_{3f}$, is calculated in order to have the ZMP at the position ZMP_f, so that the difference between $\ddot{\theta}_{3f}$ and $\ddot{\theta}_{30}$ is minimal. Therefore, the torque necessary to change the acceleration of the body link will also be minimal.

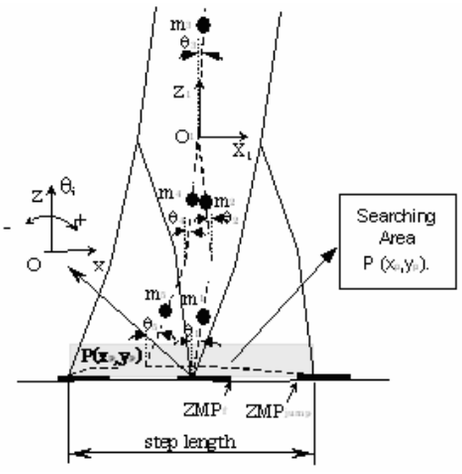


Figure 13. Five link biped robot

5.3 Problem Formulation

The problem consists of finding the joint angle trajectories, to connect the first and last posture of the biped robot for which the CE or TC is minimal. It can be assumed that the energy to control the position of the robot is proportional to the integration of the square of the torque with respect to the time. Because the joints of the manipulator are driven by torque, then the unit of torque, Nm, is equal to the unit of energy, joule. So, the cost function, J, can be defined as the following expression:

$$J = \frac{1}{2}\left(\int_0^{t_f} \tau^T \tau dt + \Delta\tau_{jump}^2 \Delta t + \int_0^{t_f} C dt\right) \tag{2}$$

where: t_f is the step time, τ is th e torque vector, $\Delta\tau_{jump}$ and Δt are the addition torque applied to the body link to cause the ZMP to jump and its duration time, and C is the constraint function, given as follows:

$$C = \begin{cases} 0 \text{ - if the constraints are satisfied,} \\ c_i \text{ - if the constraints are not satisfied,} \end{cases}$$

c denotes the penalty function vector. We consider the following constraints for our system.

- The walking to be stable or the ZMP to be within the sole length.
- The distance between the hip and ankle joint of the swing leg must not be longer then the length of the extended leg.
- The swing foot must not touch the ground prematurely.

The results generated for minimum CE cost function are compared with the angle trajectories that minimize the rate of change of the torque (Uno et al., 1989).
The cost function is as follows:

$$J_{\text{torque change}} = \frac{1}{2}\left(\int_0^{t_f} \left(\frac{d\tau}{dt}\right)^T \left(\frac{d\tau}{dt}\right) dt + \left(\frac{\Delta\tau}{\Delta t}\right)^2 + \int_0^{t_f} C dt\right). \tag{3}$$

5.4 Proposed Method

The block diagram of the proposed method is presented in Fig. 14. Based on the initial conditions and the range of searching variables, an initial population is generated. Every angle trajectory is presented as a polynomial of time. Its range is determined based on the number of angle trajectory constraints and the coefficients are calculated to satisfy these constraints. The torque vector is calculated from the inverse dynamics of the five-link biped robot (Mita et al., 1984) as follows:

$$J(\theta)\ddot{\theta} + X(\theta)\dot{\theta}^2 + Y\dot{\theta} + Z(\theta) = \tau \tag{4}$$

According to the formula (2) and (3), the cost function is calculated for minimum CE and minimum TC, respectively. The value of the cost function is attached to every individual in the population. The GA moves from generation to generation, selecting parents and producing offspring until the termination criterion (maximum number of generations

GN_{max}) is met. Based on the GA results, the gait synthesis is generated for minimum CE and minimum TC, respectively.

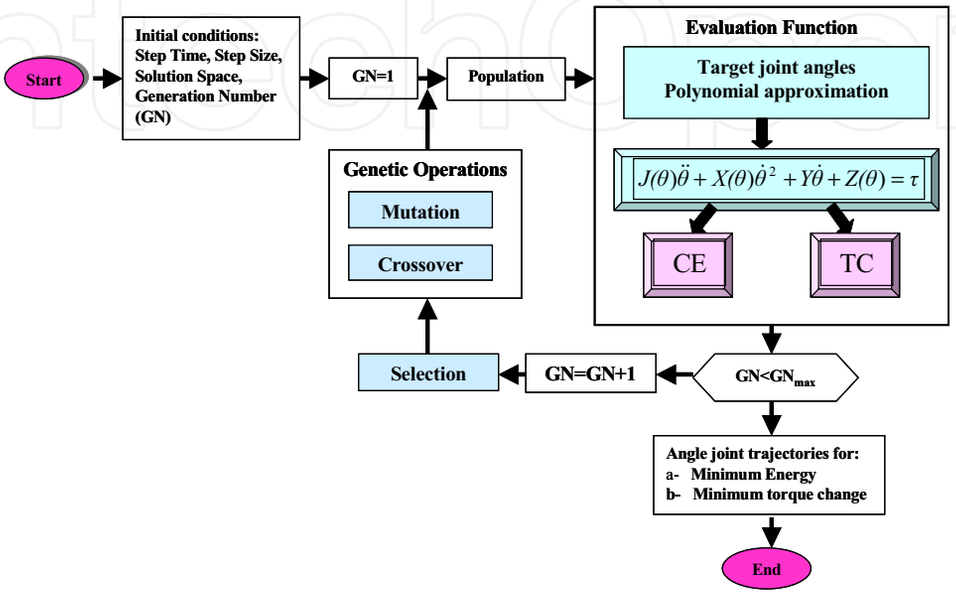


Figure 14. Proposed method

5.5 Boundary Conditions and CA Variables

To have a continuous periodic motion, the posture of the biped robot is considered to be the same at the beginning and at the end of the step. It must satisfy the following:

$$\theta_{10} = \theta_{5f},\ \theta_{20} = \theta_{4f},\ \theta_{1f} = \theta_{50},\ \theta_{2f} = \theta_{40},\ \theta_{30} = \theta_{3f}. \tag{5}$$

In order to find the best posture for walking, the optimum value of θ_{10}, θ_{20} and θ_{30} must be determined by GA. For a given step length during walking it is easy to calculate θ_{40} and θ_{50}. When referring to Fig. 13, it is clear that links 1, 2, 4 at the beginning of the step and links 2, 4, 5 at the end of the step, change the direction of rotation. Therefore, we can write:

$$\dot{\theta}_{10} = \dot{\theta}_{20} = \dot{\theta}_{40} = \dot{\theta}_{2f} = \dot{\theta}_{4f} = \dot{\theta}_{5f} = 0. \tag{6}$$

The angular velocity of link 1 at the end of the step and link 5 at the beginning of the step is considered to be the same. This can be written in the form $\dot{\theta}_{1f} = \dot{\theta}_{50}$. In order to find the best value of angular velocity, we consider it as one variable of GA, because the rotation direction of these links does not change. GA will determine the optimal value of the angular velocity of the body link, which is considered to be the same at the beginning and at the end of the step. The following relations are considered for the angular acceleration:

$$\ddot{\theta}_{10} = \ddot{\theta}_{5f},\ \ddot{\theta}_{20} = \ddot{\theta}_{4f},\ \ddot{\theta}_{1f} = \ddot{\theta}_{50},\ \ddot{\theta}_{2f} = \ddot{\theta}_{40}. \tag{7}$$

In this way, during the instantaneous double support phase, we don't need to apply an extra torque to change the angular acceleration of the links. To find the upper body angle trajectory, an intermediate angle θ_{3p} and its passing time t_3 are considered as GA variables.

To determine the angle trajectories of the swing leg, the coordinates of an intermediate point $P(x_p,z_p)$ and their passing time t_p, are also considered as GA variables. The searching area for this point is shown in Fig. 13. Based on the number of constraints, the range of the time polynomial for θ_1, θ_2, θ_3, θ_4 and θ_5 are 3, 3, 7,6 and 6, respectively.

5.6 Simulation

In the simulations, we use the parameters of the Bonten-Maru I humanoid robot. The parameter values are presented in Table 1. We must produce many locomotion modules for humanoid robot, i.e. walking, going up-stairs, going down-stairs, obstacle overcoming, etc. In this chapter, we show the simulation results for a normal gait generation.

For the optimization of the cost function, a real-value GA was employed in conjunction with the selection, mutation and crossover operators. Many experiments comparing real value and binary GA has proven that the real value GA generates better results in terms of the solution quality and CPU time (Michalewich, 1994). To ensure a good result of the optimization problem, the best GA parameters are determined by extensive simulation that we have performed, as shown in Table 5. The maximum number of generations is used as the termination function. The GA converges within 40 generations (see Fig. 5). The average of the cost function En against the number of generations is shown in Fig. 6. The 33-th generation has the lowest value, which agrees with Fig. 5 results.

Function Name	Parameters
Arithmetic Crossover	2
Heuristic Crossover	[2 3]
Simple Crossover	2
Uniform Mutation	4
Non-Uniform Mutation	[4 GNmax 3]
Multi-Non-Uniform Mutation	[6 GNmax 3]
Boundary Mutation	4
Normalized Geometric Selection	0.08

Table 5. GA functions and parameters

Based on the Bonten-Maru I parameters, the step length can vary up to 0.5m. If the step length is smaller then 0.36 m, the ZMP can smoothly pass from one foot to the other during the instantaneous double support phase. The problem becomes more complex when the step length is larger then 0.36 m because the ZMP must jump to the new supporting foot. In the following, the optimal motion for the step length 0.42 m and step time 1.2 s is analyzed. The GA results are shown in Table 6. The joint angle trajectories (θ_i), the torque vector (τ_i) and the optimal motion are shown in Fig. 17 and Fig. 18, for minimum CE and minimum TC, respectively. Based on the simulation results, we see that the biped robot posture is straighter, like the human walking, when the minimum CE is used as cost function. Comparing Figs 17(b) and 18(b), the torques change more smoothly when minimum TC is used as a cost function. The swing foot does not touch the ground prematurely, and the

ZMP is always inside the sole length, as presented in Fig. 19. At the end of the step, the ZMP is at the position ZMP_f, as shown in Fig. 13. At the beginning of the step, the ZMP is not exactly at the position ZMP_{jump} because the foot's mass is not neglected. It should be noted that the mass of the lower leg is different when it is in supporting leg or swing leg. The values of CE, calculated by the equation (2) for minimum CE and minimum TC gait synthesis, are presented in Fig. 20. The minimum CE gait synthesis reduces the energy by about 30% compared with minimum TC.

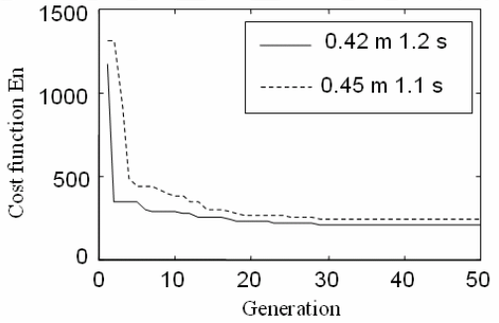


Figure 15. Cost function En vs generations

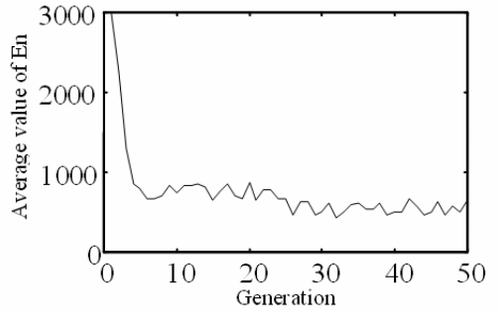


Figure 16. Average of the cost function En vs. generations

GA variables	1.1.1. Limits	1.1.2. GA Results	
		1.1.3. CE	TC
θ_{10}	-0.3~0	-0.1370	-0.0006
θ_{20}	-0.7~-0.3	-0.4374	-0.5199
θ_{30}	0 ~ 0.3	0.1309	0.1100
θ_{3p}	-0.1~0.2	0.0997	0.1001
t_3	0.3~0.7	0.5670	0.5900
x_{p1}	-0.2~0.2	-0.0861	-0.0585
y_{p1}	0.01~0.04	0.0142	0.0118
t_{p1}	0.1~0.9	0.4516	0.4471
$\dot{\theta}_{1f}$	0 ~ 2	0.6613	0.3935
$\dot{\theta}_{30}$	-1 ~ 1	0.0198	0.2514

Table 6. Variable space and GA results

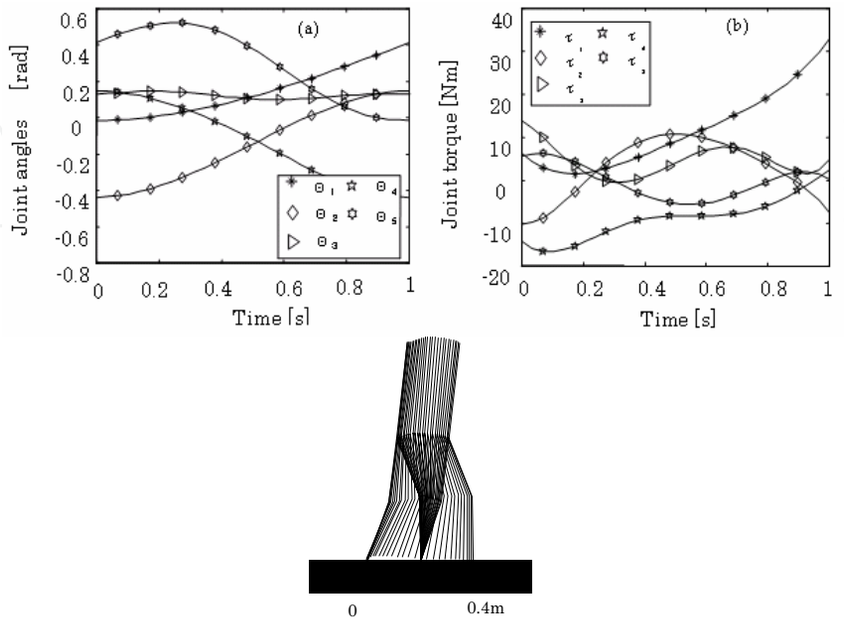


Figure 17. GA results and walking pattern of Joint angle (a) and Joint torque (b) for minimum CE cost function

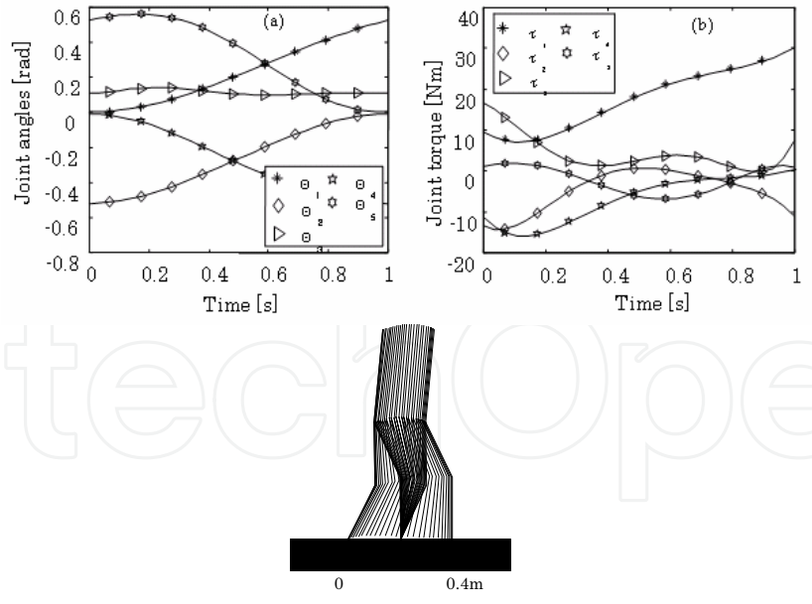


Figure 18. GA results and walking pattern of Joint angle (a) and Joint torque (b) for minimum TC cost function

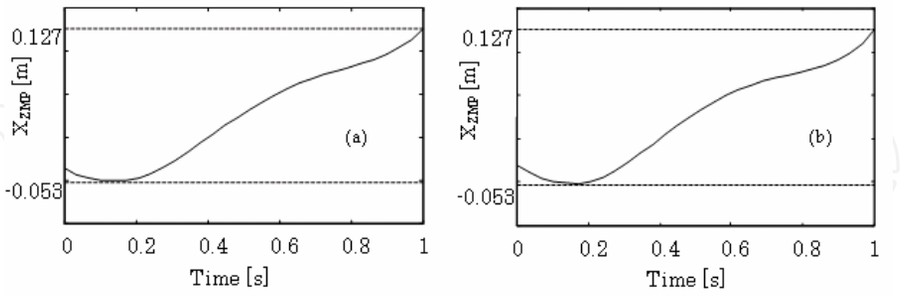


Figure 19. ZMP position for CE (a) and TC (b)

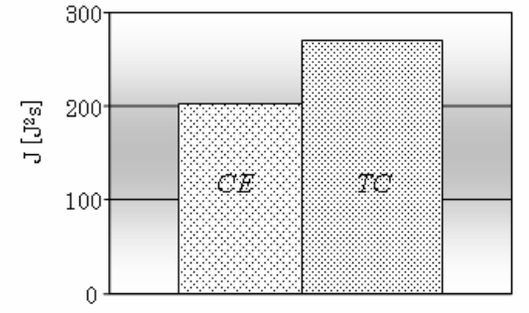


Figure 20. Energy comparison for normal step

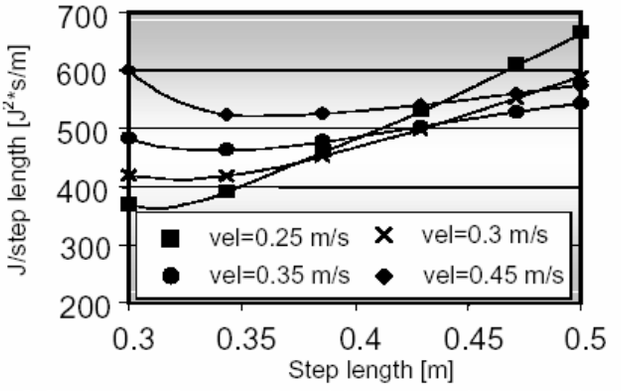


Figure 21. CE vs. the walking velocity

The energy required for 1 meter walking against the step length is presented in Fig. 21 for several walking velocities. In this case, the cost function is divided by step length. One result, which comes out from this figure, is that as the walking velocity gets higher, the optimal step length gets larger. The curves become more tended and don't intersect each other.

The energy required when the biped is moving slowly with a large step length is high. This makes that the curves of slow velocities to intersect the others. This suggests that low walking velocity doesn't mean low CE. In addition of walking velocity the step length must be also considered.

5.7 NN Implementation

In contrast to other optimization methods, GA needs more time to get the optimal solution. In our simulations, it needs about 10 minutes. However, in real time situations, based on the step length and step time, the angle trajectories must be generated in a very short time. In order to apply our method in real time, we considered teaching a NN based on the data generated by GA.

Our method employs the approximation abilities of a Radial Basis Function Neural Network (RBFNN) (Haykin, 1999). When the biped robot has to walk with a determined velocity and step length, the NN input variables would be the step length and step time. The output variables of the NN are the same as the variables generated by GA. From previous section, for a given step length it is a step time for which the CE is minimal. For this reason, when the walking velocity is not important the NN output will be the GA variables and the best step time. The NN input will be only the step length. In this chapter, we only show the NN simulation results, where the step length and step time are adapted as input variables.

RBFNN

The RBFNN involves three layers with entirely different roles, as shown in Fig. 22. The input layer connects the network to the environment. The second layer (hidden layer) applies a nonlinear transformation from the input space to the hidden space, which is of high dimensionality. We use as nonlinear transfer function the Gaussian function, because this is the most widely used function. The Gaussian function is expressed as follows:

$$h_i(x) = \exp(-\frac{\| x_i - c_i \|}{\sigma_i}) , \tag{8}$$

where: h_i is the i-th output of the neuron, x_i is the input vector, c_i and σ_i are the center and the width of the i-th RBF neuron.

The width of Gaussian function is a positive constant that represents the standard deviation of the function. The output layer is linear, supplying the response of the network to the activation pattern (the signal applied to the input layer). Based on the number of nodes in the hidden layer the RBFNN are divided in generalized and regularization RBF networks. In our simulations we use a regularization RBF network.

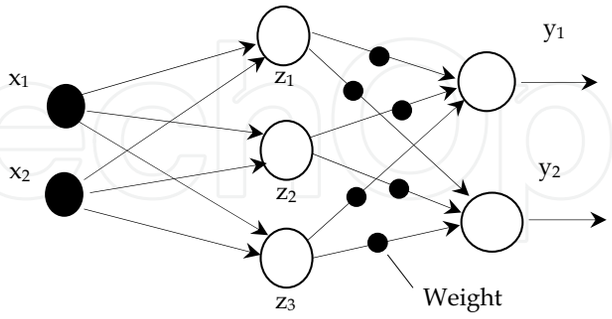


Figure 22. The structure of RBFNN

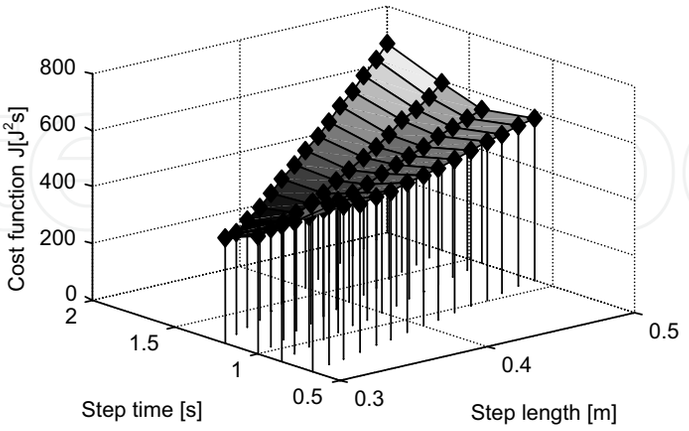


Figure 23. Cost function J vs step time and step length

5.8 RBFNN Results

The step length used to teach the NN varies from 0.3 to 0.5 and step time from 0.7s to 2s. Relation between cost Function J against the step length and step time is presented in Fig. 23. Because in our RBFNN the centers are the same with training data, determining the best value of the width σ is important in order to minimize the overall training error. The goal is to select the width in such way to minimize the overlapping of nearest neighbors, as well as maximize the generalization ability. The width selection depends on the distance between the two neighbor vectors. In our case the width σ is the same for all neurons, because the distances between the centers are the same. In Fig. 24 is shown the mean square error (mse) versus the width σ. The minimal value of mse is for $\sigma = 0.73$. We present the results generated by the NN for a set of step length and step time. It differs from training examples, for which the RBFNN output are the same with GA outputs. The input data of the NN have been [0.45 m 1.2 s]. The results generated by GA and NN are presented in Table 7. Based on these results, the angle trajectories are shown in Fig. 25. The difference between the NN and GA angle trajectories is very small. The time to generate the solution by the NN is 100ms, which is good for the real time implementation. The value of CE for NN gait is only 3.2 % more compared with GA one, as shown in Fig. 26.

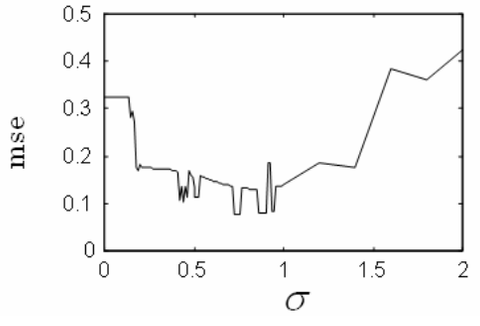


Figure 24. Mse vs the width σ

GA Var	Step length 0.42m, Step time 1.2s			St. length 0.45m, Step time 1.2s	
	Limits	CE	TC	GA	NN
θ_{10}	-.3~.0	-0.122	-0.0004	-0.124	-0.086
θ_{20}	-.7~-.3	-0.455	-0.57	-0.423	-0.447
θ_{30}	0~.3	0.107	0.370	0.258	0.264
$\dot{\theta}_{1f}$	0~2	0.523	0.399	0.395	0.259
$\dot{\theta}_{30}$	-1~1	-0.031	-0.11	-0.191	-0.167
θ_{3p}	-.1~.2	0.084	0.370	0.099	0.094
t_3	.2~.8	0.518	0.761	0.567	0.591
x_p	-.2~.2	-0.135	-0.132	-0.023	-0.006
y_p	.1~.4	0.016	0.017	0.018	0.014
t_p	0~1	0.441	0.432	0.613	0.632

Table 7. GA and NN results

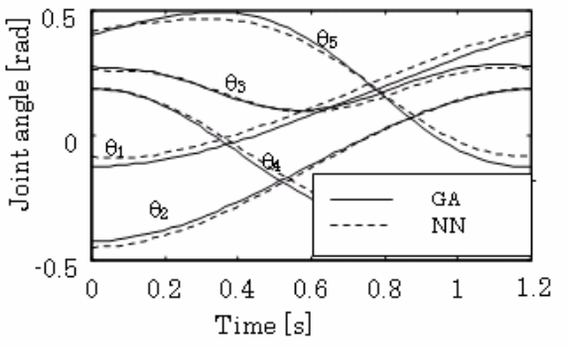


Figure 25. Comparison of GA and NN angle trajectories (step length 0.45m and step time 1.2s)

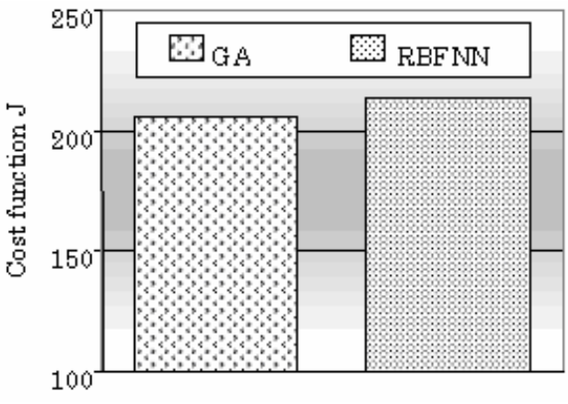


Figure 26. Comparison of values of J cost functions by GA and NN

6. Teleoperation Systems and its Applications

It is so effective to replace human being with humanoid robot for operation in disaster site and/or extreme environment (ex. Atomic power plants). These environments might change at every moment, thus, an operator must remotely control the robot. In order to remotely control the humanoid robot, several user interfaces have been developed. For example, remote control cockpit system is presented in (Hasunuma et al., 2002). In (Yokoi et al., 2003) a portable remote control device system with force feedback master arms was introduced. By using it, the operator can give his/her desired arm motion to the robot viscerally. But such device was very complex and expensive.

On the other hand, simple master device system (Neo et al., 2002) has two joysticks. Although the cost is reduced, because of a small number of degrees of freedom the system can not realize complex motions. In addition, it is hard to deal with environmental variations like sudden accidents. In order to overcome the shortcomings of previous systems, our objectives have been to: (1) develop a humanoid robot teleoperation system with a simple user interface; (2) able to reflect the operator's order; and (3) the cost for development and running to be low. Therefore, we first developed an on-line remote control of humanoid robot's arms. To carry out an easy operation, we developed an ultrasonic 3D mouse system as a master device for teleoperation system. In addition, we built a simple VR interface and a HMD equipped with a gyro sensor. In this chapter, we show the details of our teleoperation system and its user interface.

In this section, we present a new humanoid robot teleoperation system using the Internet/LAN and an easy user interface and a long distance teleoperation experiments.

6.1 Online Remote Control of Humanoid Robot using a Teleoperation System and User Interface

The schema of our teleoperation system is shown in Fig. 27. Our teleoperation system is a server-client system through the internet/LAN based on CORBA. There are two server PCs one to communicate and control the robot motion and the other for live streaming of CCD camera image. In addition, there are two client PCs for user interfaces, and for receiving live streaming vision images. While the operator sends the server his/her command including task commands and/or planned motion based on robot vision, the robot implements the order and returns results to the client, that is, current robot status (standing or crouching and so on) and robot vision. The communication between the operator and the humanoid robot is realized through TCP/IP with CORBA for motion operation and UDP for live streaming.

6.1.1 Operation Assist User Interface

Application of a joystick for giving commands to the robot based on images collected by robot vision system is a difficult task because of some troubles to manipulate the input device and robot camera at once. In addition, a joystick is not always suitable for quick 3D motion operation and to manipulate the input device and camera, separately. In order to overcome these difficulties, we decided to design the user interface as follow; 1) receive the operator's hand tip trajectory as order motion by a master device, 2) compose a VR interface with a HMD equipped with a gyro sensor to share the robot vision. The former needs to determine the space coordinates of operator's hand tip. Considering the environment, the operator's working area, the precision of measurement, and the cost of the system, we

developed an ultrasonic 3D mouse system applying ultrasonic positioning system (Yu et al., 2001). The ultrasonic positioning system is applied to reduce the workload of manipulating the robot camera. Based on the visual information, the operator can synchronize the robot head motion.

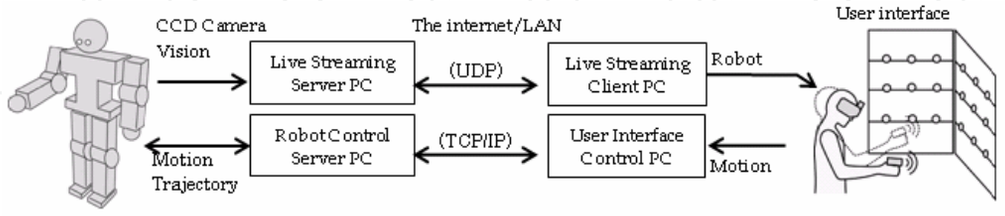


Figure 27. Schema of humanoid robot teleoperation system

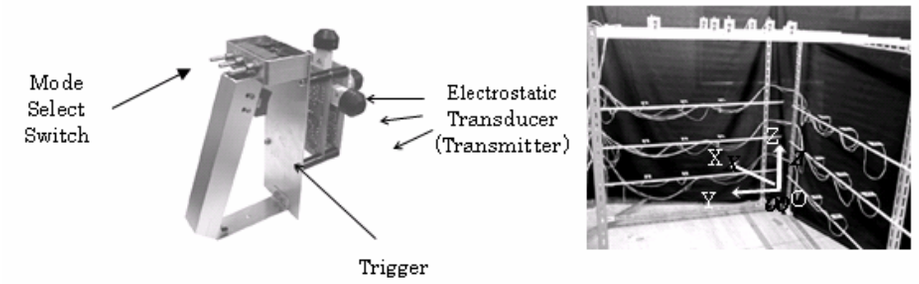


Figure 28. Ultrasonic 3D mouse and ultrasonic receiver net

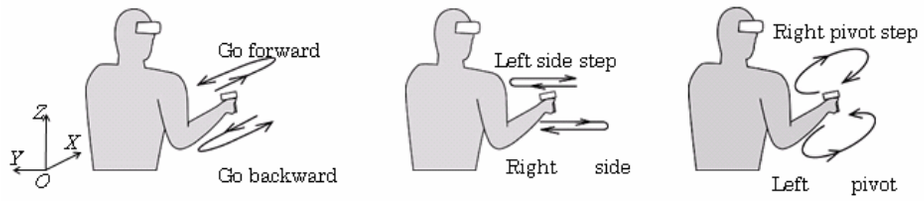


Figure 29. Gesture patterns corresponding to locomotion commands

6.1.2 Ultrasonic 3D Mouse System

This is a system to extract the operator's hand tip trajectory. The configuration is as follow; an ultrasonic 3D mouse; an ultrasonic receiver net cage and the system control PC (see Fig. 28). The 3D mouse has three electrostatic transducers (transmitter) and one trigger switch and three mode select switches. The receiver net cage has three planes that ultrasonic receivers are allocated by 300×300 mm regular interval on the frame of each plane. The origin of coordinate system is also shown in Fig. 28. The electrostatic transducers used in this study are MA40E7S/R made by Murata Manufacturing Co. Ltd, Japan. The specifications are shown in Table 8 . This system has two operating modes for manipulation of robot arms: the direct mode, which control the arm motion in real time, and the command mode to operate the locomotion by preset commands. The select switches on the 3D mouse are used to select the desired operating mode. Direct mode is used to operate one arm (right / left mode), or both arms (symmetrical / synchronized mode).

While the operator pulls the trigger, the system estimates 3D mouse position and extract the displacement vector of 3D mouse at every sampling. The vector is given to the robot as a reference motion data (reference motion vector). By using this system, the operator can generate in real time the robot's hand tip trajectory viscerally by dragging and dropping an icon on GUI desktop. In our system there is no need to consider the initial positioning between the 3D mouse and the robot hand tip at the start of operation, making easier to operate. On the other hand, the command mode is used to realize pre-designed motions like gesture input mode for walking motion. Here, gesture means an identification of the 3D mouse trajectory pattern. Preset commands for locomotion correspond with gesture patterns as shown in Fig. 29.

	Transmitter	Receiver
	MA40E7S	MA40E7R
Nominal Frequency	40 [kHz]	
Minimum Receiving Sensitivity	-74 [dB]	--
Minimum Transmitting Sensitivity	--	106 [dB]
Beam Angle	100 [deg]	
Capacitance	2200±20% [pF]	
Maximum Voltage	85 [$V_{p\text{-}p}$]	
Operating Conditions Temperature	-30 ~ 85 [^{o}C]	
Measuring Range	0.2 ~ 3 [m]	
Resolving Power	9 [mm]	
Cover Size	18(D) x 12(H) [mm]	
Weight	4.5 [g]	
Character	Waterproof	

Table 8. Technical specifications for MA40E7$^{R}/_{S}$ electrostatic transducer

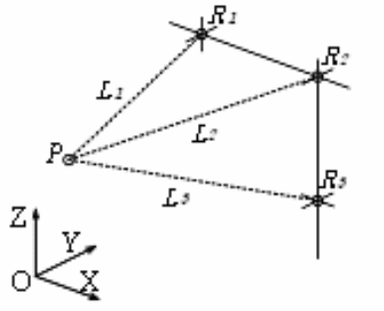


Figure 30. Position estimation

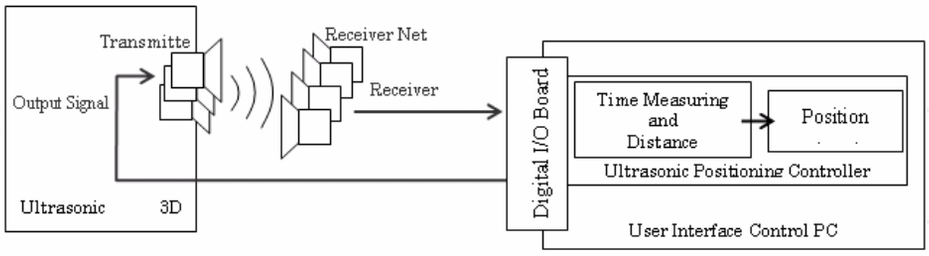


Figure 31. Diagram for position estimation

6.1.3 Ultrasonic Positioning Estimation

In our system, we know the speed of sonic wave at the air temperature and the propagation distance by measuring the wave propagation time. At least by knowing three distances between the 3D mouse and receivers, we can estimate the position of the 3D mouse in the sensor net by principle of triangulation. When the wave propagation time T_i [s] (i =1,2,3) is measured, the propagation distance L_i [m](i =1,2,3) is estimated as follow:

$$L_i = (331.5+0.6t)\ T_i \tag{9}$$

Here, t is room temperature [^{o}C]. Assuming that receiver positions R_i (x_i, y_i, z_i), (i =1,2,3) are known and exist on same plane (not on a straight line) in an arbitrary Cartesian coordinate as shown in Fig. 30, the position of the 3D mouse P (x, y, z) is estimated by the following formulations:

$$\left.\begin{aligned} (x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2 &= L_1^2 \\ (x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2 &= L_3^2 \\ (x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2 &= L_2^2 \end{aligned}\right\} \tag{10}$$

Figure 31 shows the diagram for the position estimation. When the ultrasonic positioning controller sends output signal to transmitter on the 3D mouse, it begins measuring the wave propagation time. In addition, a receiver detects ultrasonic waves and returns a receiver signal to the controller making possible to determine the time between the 3D mouse and the receiver. After sampling for 4 ms, the controller will calculate the distance between the 3D mouse and receivers, and estimate the position of 3D mouse.

The control PC used in this study is a DOS/V compatible PC with a Pentium III CPU (733MHz) and Linux OS (Red Hat 9). The time measuring process is executed by using CPU internal clock counter on the control PC. The precision for time measurement (the CPU frequency) depends on its operative conditions (power supply voltage, internal temperature and so on). But the sampling performance is about 250 kHz on the average, and the theoretical resolution for distance measurement is about 1.3mm at room temperature 20 ^{o}C. The total processing time is set 10 ms.

6.1.4 Live Streaming System

A live streaming system is applied to transmit robot camera vision to the operator. The robot camera vision is captured and it is encoded in real time to mpeg4 format data (QVGA (320x240 pixels)) on the live streaming server PC. Then it is transmitted to the client PC by UDP (Fig. 32). For the server and client application, we applied multicast application

"FocusShare", which is distributed at OpenNIME web site. The server PC used in this system is the DOS/V compatible PC with a Pentium IV CPU (2.53GHz) and Windows OS (Windows XP SP2). The live streaming data is decoded on the client PC (Notebook PC with Pentium M (900MHz) and Windows 2000 SP4), and projected on HMD. HMD used is i-Visor DH-4400VP made by Personal Display Systems, Inc., USA, and it has two 0.49inch, 1.44 million pixels LCD, and supports SVGA graphic mode. The gyro sensor used is InterTrax² is made by InterSense Inc. of USA, which can track roll, pitch, yaw direction angles (except for angular speed and acceleration), and its minimum resolution is 0.02deg.

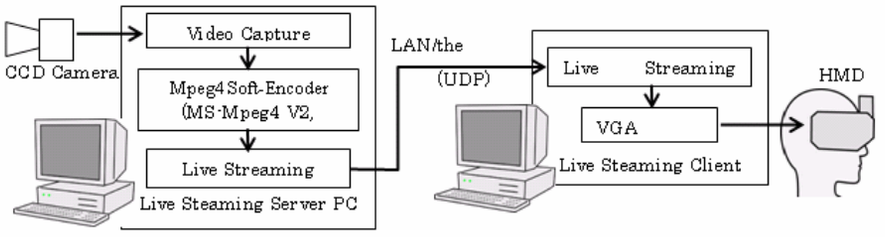


Figure 32. Live streaming system

6.1.5 Motion Trajectory Generation

For the motion trajectory generation we first added a reference motion vector given by the 3D mouse to current robot hand tip position. Therefore, the reference robot hand tip position is set. By linear interpolating the position and current robot hand, the reference hand tip trajectory is pre-released based on a given reference motion time (here, 10ms). At this moment, the trajectory is checked about collision and workspace of hand tip.

If there is any error, a new reference hand tip position will be set again, and a new reference hand tip trajectory will be released. Finally, it will be converted to reference arm joint angle trajectory by inverse kinematics. In Direct Mode, the reference motion vector is essentially handled as data for the right arm. Both reference hand tip positions are determined by adding same reference motion vector to each current robot hand. But in symmetrical mode, left reference hand tip position is determined by adding a reference motion vector that its Y direction element is reversed.

6.1.6 Experiments and Results

In order to evaluate the performance of the developed system, we completed experiments with Bonten-Maru II humanoid robot. In the following, we give the results of these experiments. First we discuss the results of right arm motion using the teleoperation system in a LAN environment. In this experiment the operator drew a simple quadrilateral hand tip trajectory on Y-Z plane in the ultrasonic receiver net with the 3D mouse. Fig. 33 (a) and (b) show an order trajectory given by 3D mouse and a motion trajectory of right robot hand tip. Note that in this experiment, the room temperature was 24 ◦C, and Fig. 33 (b) is viewed from the origin of right arm coordinate system located in the right shoulder. Although there is a difference in scaling that it is caused by feedback errors, each motion pattern matches well. And also, in Fig. 34 is shown the operation time in every communication. The horizontal axis is the number of communication times. There are some data spreads due to network traffics, but the operator could carry out the experiment in real time without serious time delay error.

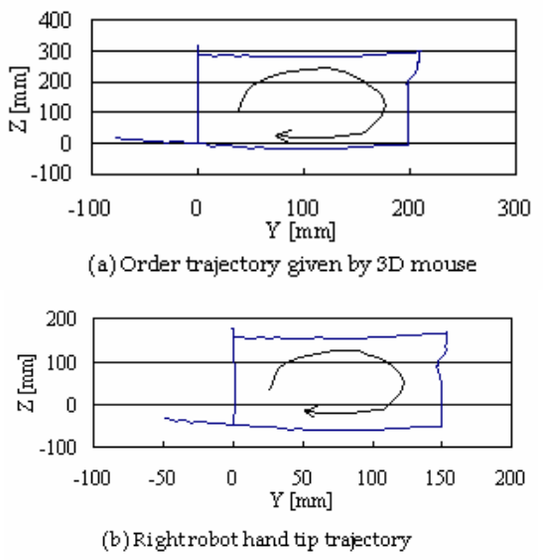


Figure 33. Results of teleoperation experiment

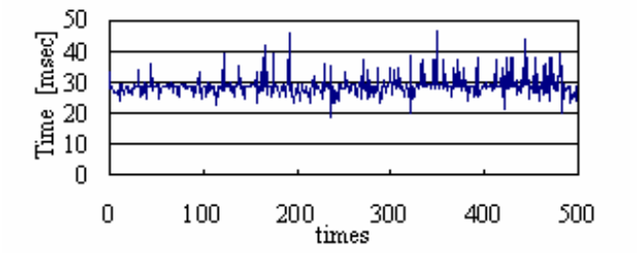


Figure 34. Operation time

In order to further verify the system performance, we performed an experiment to evaluate the ability to replicate the hand tip motion generated by the operator in *Y*-*Z* plane. In this experiment, the operator draws a quadrilateral hand tip trajectory on *Y*-*Z* plane. The operator cannot look his/her own hand because of the HMD. A stroboscopic photograph of the robot motion during the experiment is shown in Fig. 35. Fig. 36 (a) and (b) show an experimental measured operator's hand tip trajectory in the coordinate of receiver net and the right robot hand tip position viewed from the origin of right arm coordinates. Also in the Fig.11, the direction indicated by arrow shows the direction of motion. Each dot indicates the measured positions during the operation. The interval of each dot means one-operation cycle, which is about 1.5sec, including the sensing time in the receiver net, the robot motion time and the time-delay by the network traffics. The difference between Fig. 36 (a) and (b) originates in the decreasing reference data scale to 70%. In addition, this difference is exist because the robot hand tip trajectory is sometimes restricted due to the limitation of the workspace, the range of joint angles and change in trajectory to avoid the collision with the body. But both trajectory patterns are similar.

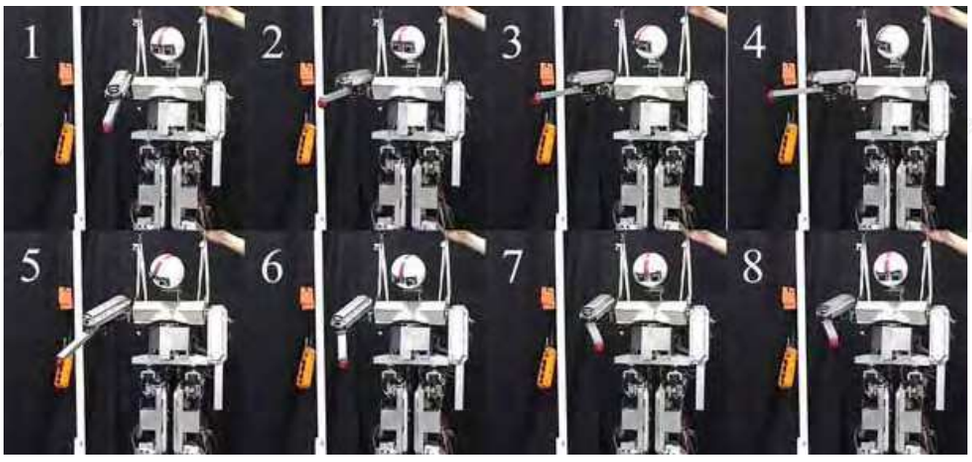
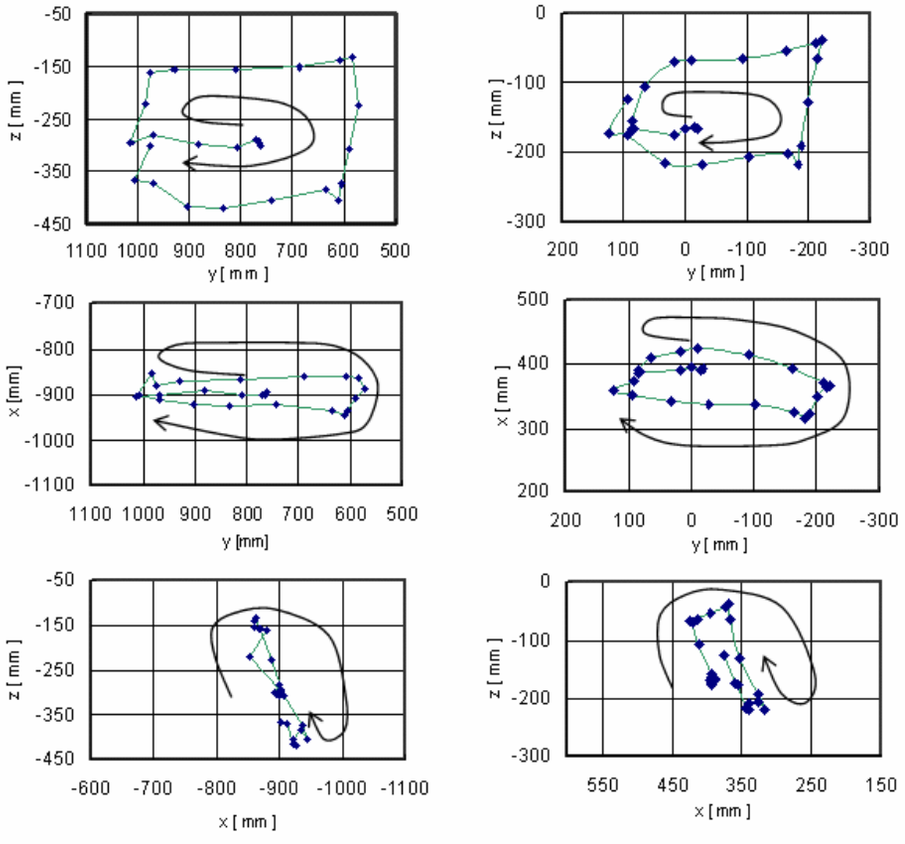


Figure 35. The robot motion during the experiment



(a) Measured operator hand tip (b) Robot right hand tip

Figure 36. Results of the experiment

As previously mentioned, the operator cannot check on his/her own hand tip position. These mean that, the operator could correct his/her own hand tip position using the HMD vision and generate his/her planned motion. In other words, our user interface can function as a VR interface to share data with the robot. As the matter of fact, the communicating interval between the CORBA client and the CORBA server must be considered in order to minimize as much as possible.

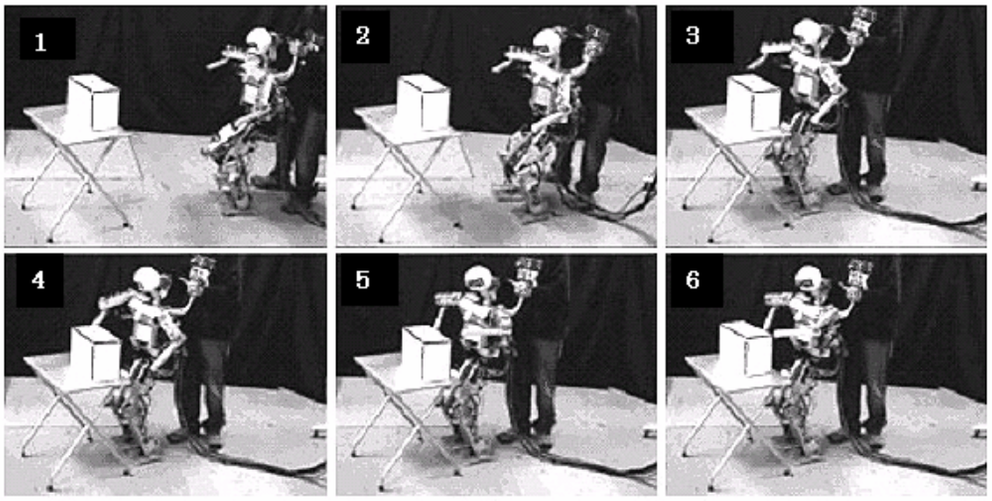
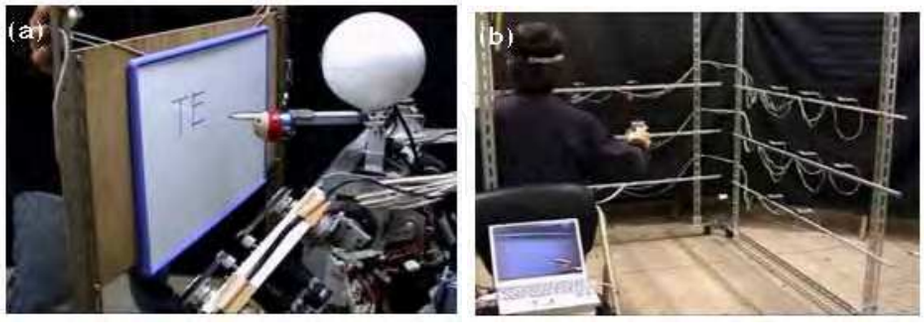


Figure 37. Video capture of teleoperation experiment

Next, we performed experiments using all the system. In this experiment, the operator gives locomotion commands by gesture input, in order to move the robot to a target box. Then the robot receives the command to touch the box. In Fig. 37 is shown a video capture of the robot. This experiment indicates that by using the developed teleoperation system we are able to communicate with the humanoid robot and realize complex motions. Fig. 38 shows a teleoperation demonstration to draw simple characters using the 3D mouse. The operator could draw simple characters easily.



(a) Drawing simple characters (b) Operator with the 3D mouse

Figure 38. Demonstration test of the 3D mouse

6.2 Long Distance Teleoperation via the Internet

In this section, we explain a teleoperation system to control the humanoid robot through the internet. We carried out experiments on the teleoperation of the humanoid robot between Deakin University (Australia) and Yamagata University (Japan) (Nasu et al., 2003). The experimental results verified the good performance of the proposed system and control.

6.2.1 Teloperation system

Figure 39 shows the teleoperation schematic diagram. The operator uses this system as a CORBA client and commands several kinds of motions, i.e. walking, crouching, crawling, standing up, etc. Figure 40 shows the HRCA for Bonten-Maru II humanoid robot. We have implemented the following main modules: DTCM, MCM, JTM, GSM, JAM, FCM, CCM VCM and UIM in this figure. Each module corresponds to "Data Transmission", "Target Position", "Angle Trajectory Calculation", "Sensor", "Position", "Feedback Control", "CCD Camera", "Video Capture Control" and "Command Generator", respectively. Up to now, the operator can command the number of steps and humanoid robot walking direction.

The operator receives the camera image mounted in humanoid robot's head and based on the data displayed in PC1, measures the distance between the robot and objects. PC2 is used to read and manipulate the sensor data and send output commands to the actuators. PC3 is used to capture the CCD camera image. A notebook type computer with a Pentium III, 700 MHz processor running Red Hat Cygwin on the Windows XP was used as the client computer (PC1). Two different type computers were used as server computers: PC2 (Celeron, 433MHz), PC3 (Pentium II, 200 MHz) running Red Hat Linux 7.3.

6.2.2 Data Stream

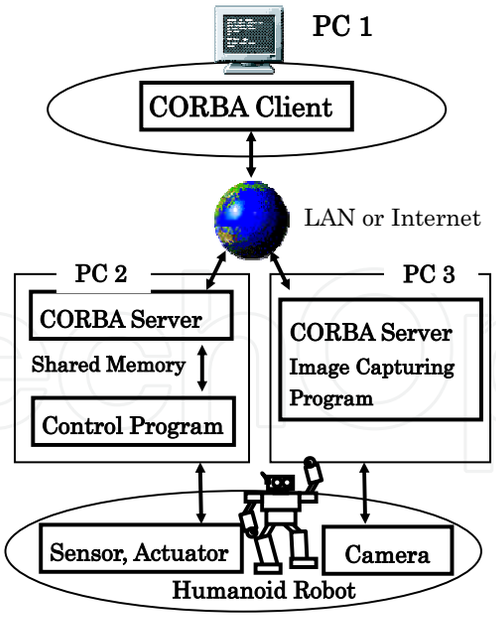


Figure 39. Teleoperation concept

CORBA server program receives a motion command from CORBA client and writes it on the shared memory of PC2. Sending and receiving the data between CORBA server program and control program are executed by using shared memory feature of UNIX OS. Among all programs on the LINUX, the control program OS implemented in accordance to highest-priority due to keep the control execution period. CORBA server program is implemented at default value. When the operator watches the camera image, PC1 and PC2 are used. When the operator executes CORBA client program of PC1, the image data, which is captured in PC3, is imported to PC1. The operator can use it to measure the object distance, to recognize the environment condition and make decision of the optimal motion.

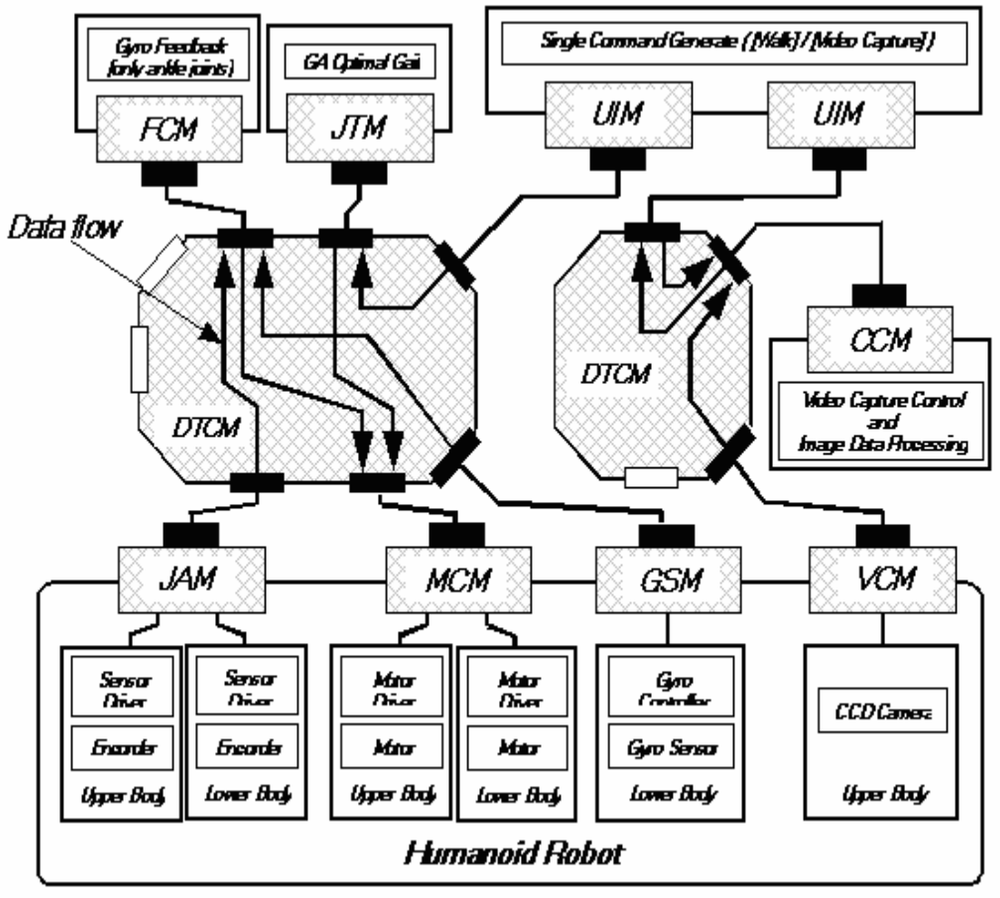


Figure 40. The HRCA for Bonten-Maru II humanoid robot

6.2.3 Experiments and Results

First, we measured the image capturing job time through the internet. The typical job time averaged about 13 second to a few minutes, because there are many communication traffic loads in the both universities LANs.

Second, using the humanoid robot, we have carried out two types of teleoperation obstacle avoidance experiments between Australia and Japan. The operator executed teleoperation program from Deakin University (Australia) through the internet.

Experiment 1: Obstacle avoidance by walk

At first, we set a box on the floor in front of humanoid robot. The operator recognized it in the image data from the humanoid robot. Fig. 41 shows a series of the obstacle avoidance walking motions and image data of the humanoid robot eyes. The humanoid robot received the following motion commands:

- Walk front (or back)
- Side step to left (or right)
- Spin left (or right)

The operator measures the distance between the robot and the obstacle, and plans a walk trajectory to avoid the obstacle. Because the measured obstacle data is not precious, the motion command is not always the best. But the operator can correct the walking trajectory by using the image information easily.

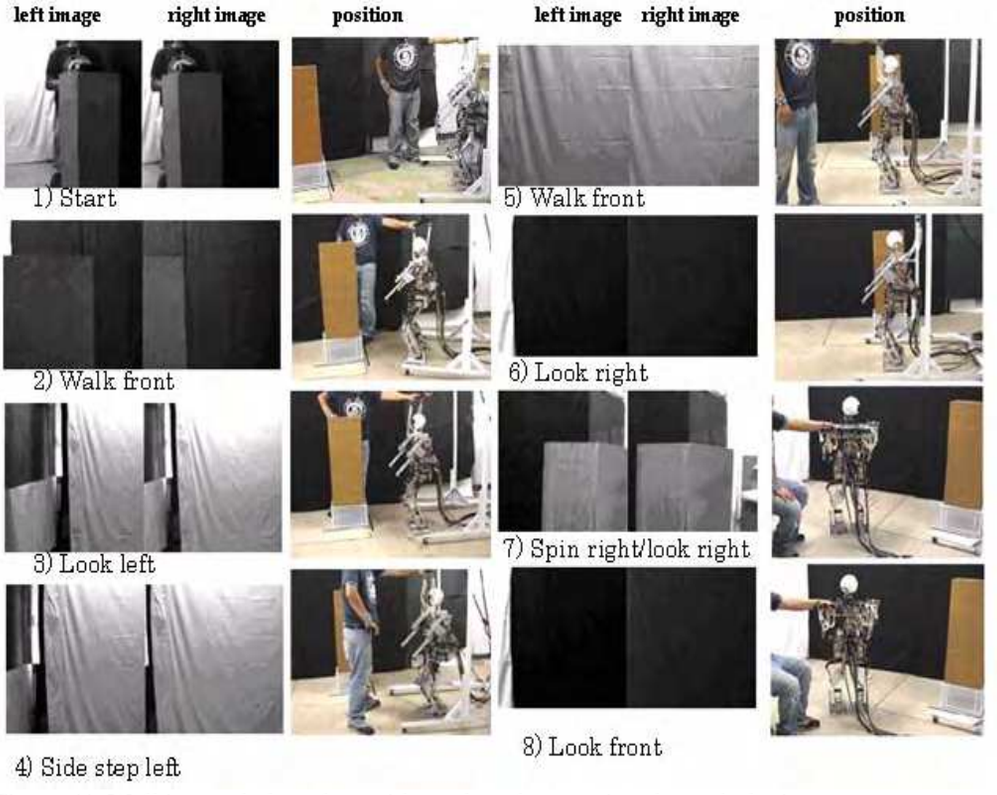


Figure 41. Walking and obstacle avoidance by teleoperation through the internet

Experiment 2: Sneaking under a low ceiling gate

At second, we set a low ceiling gate in front of the humanoid robot. The operator recognized it in the captured images data from the humanoid robot and judged that humanoid robot

could not go through the gate having the body in upright position. Fig. 42 shows a series of the sneaking under a low ceiling gate (obstacle). The client commanded the following motion; 1) look front, 2) squat, 3) crawl start, 4)-8) crawl, 9) stand up, and 10) look front. The humanoid robot could go through the gate successfully.

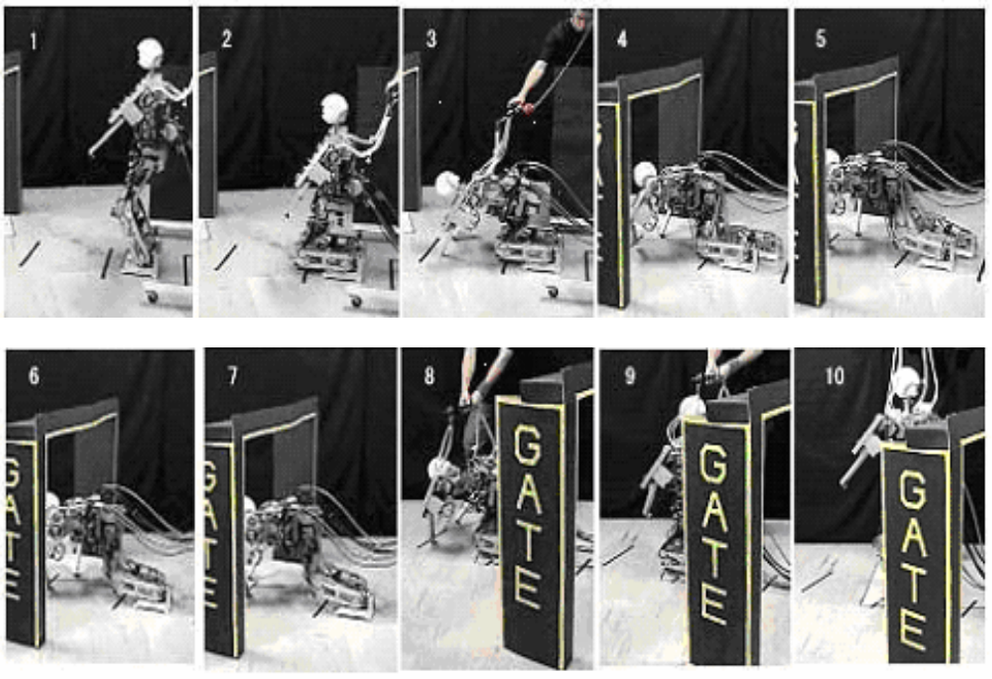


Figure 42. Sneaking and crawling under a low ceiling gate to avoid obstacle

7. Summary and Conclusions

We have developed anthropomorphic prototype humanoid robot; Bonten-Maru I and Bonten-Maru II. The Bonten-Maru humanoid robot series are one of few research prototype humanoid robots in the world which can be utilized in various aspects of studies. In this research, we utilized the Bonten-Maru in development of the CORBA-based humanoid robot control architecture, the optimal gait strategy and the teleoperation via internet.

7.1 CORBA-Based Humanoid Robot Control Architecture (HRCA)

In this section, we proposed a new robot control architecture called HRCA. The HRCA is developed as a CORBA client/server system and is implemented on the Bonten-Maru I humanoid robot. The HRCA allows easy addition, deletion, and upgrading of new modules. We have carried out simulations and experiments to evaluate the performance of the proposed HRCA. The experimental result shows that the proposed HRCA is able to control the static motion of humanoid robot accurately. By using the proposed HRCA various humanoid robots in the world can share their own modules each other via Internet.

7.2 Optimal Gait Strategy

This section presents the real time generation of humanoid robot optimal gait by using soft computing techniques. GA was employed to minimize the energy for humanoid robot gait. For a real time gait generation, we used the RBFNN, which are trained based on GA data. The performance evaluation is carried out by simulation, using the parameters of Bonten-Maru I humanoid robot. Based on the simulation results, we conclude:

- Each step length is optimal at a particular velocity;
- The stability is important to be considered when generating the optimal gait;
- The biped robot posture is straighter when minimum CE is used as the cost function, which is similar to the humans;
- The energy for CE is reduced 30% compared with TC cost function.

7.3 Teleoperation System and its Application

In this section, we described humanoid robot control architecture HRCA for teleoperation. The HRCA is developed as a CORBA client/server system and implemented on the new humanoid robot, which was designed to mimic as much as possible the human motion. Therefore, the humanoid robot can get several configurations, because each joint has a wide range rotation angle. A long distance teleoperation experiments between Japan and Australia were carried out through the internet. By using the image data from the humanoid robot, the operator judged and planned a series of necessary motion trajectories for obstacle avoidance.

This section also presented the teleoperation system for a humanoid robot and the operation assistance user interface. We developed an ultrasonic 3D mouse system for the user interface. In order to evaluate the system performance, we performed some teleoperation experiments the Bonten-Maru II humanoid robot. The results show that our system gives good results for control of humanoid robot in real time. However, there are still some problems which need to be considered in the future such as:

- The communication of live streaming system beyond network rooters.
- 3D mouse operation of robot hand postures.

Up to now we have applied the developed teleoperation system and the user interface on humanoid robot motion generation in simple environments. However, in complex environments the humanoid robot must generate skillful motions in a short time based on the visual information and operator's desired motion

The experimental results conducted with Bonten-Maru humanoid robot show a good performance of the system, whereby the humanoid robot replicates in real time the operators desired arm motion with high accuracy. The experimental results also verified the good performance of the proposed system and control.

8. Future Works

Recently, we focus in the development of contact interaction-based humanoid robot navigation (Hanafiah et al., 2006). Eventually, it is inevitable that the application of humanoid robots in the same workspace as humans will result in direct physical-contact interaction. We have proposed intelligent algorithm called groping locomotion (Hanafiah et al., 2005) to navigate humanoid robot locomotion by grasping using its arm and also avoiding obstacle. This method is useful during operation in dark area and also hazardous

site. In addition, for the humanoid robot to work along human effectively, especially for object handling tasks, the robot will require additional sensory abilities. Besides sensor systems that help the robot to structure their environment, like cameras, radar sensors, etc., a system on the robot's surface is needed that enables to detect physical contact with its environment. A tactile sensor system is essential as a sensory device to support the robot control system. This tactile sensor is capable of sensing normal force, shearing force, and slippage, thus offering exciting possibilities for application in the field of robotics for determining object shape, texture, hardness, etc. In current research, we are developing tactile sensor that capable to define normal and shearing force, with the aim to install it on the humanoid robot arm (Ohka et al., 2006). This sensor is based on the principle of an optical waveguide-type tactile sensor. The tactile sensor system is combined with 3-DOF robot finger system where the tactile sensor in mounted on the fingertip. We believe that the demand for tactile sensing devices will grow in parallel with rapid progress in robotics research and development.

9. Acknowledgement

A part of this research was supported by fiscal 2006 grants from the Japan Ministry of Education, Culture, Sports, Science and Technology (Grant-in-Aid for Scientific Research in Exploratory Research, No. 18656079). The authors would like to thank all Nasu Lab. members, Ohka Lab. members and all individual involved in this research for their contribution, work and effort towards successful of this project.

10. References

Booch, G.; Rumbaugh, J. & Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Addison-Wesley

Capi, G.; Nasu, Y.; Mitobe, K. & Barolli, L. (2003). Real time gait generation for autonomous humanoid robots: A case study for walking, *Journal Robotics and Autonomous Systems*, Vol. 42, No.2, (2003), pp. 169-178

Channon, P.H.; Pham, D.T. & Hopkins, S.H. (1996). A variational approach to the optimization of gait for a bipedal robot, *Journal of Mechanical Engineering Science,* Vol. 210, (1996), pp. 177-186

Fowler, M. & Scott, K. (1997). *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley

Hanafiah, Y.; Yamano, M.; Nasu, Y. & Ohka, M. (2005). Obstacle avoidance in groping locomotion of a humanoid robot, *Journal of Advanced Robotic Systems*, Vol.2 No. 3, (September 2005) pp. 251-258, ISSN 1729-8806

Hanafiah, Y.; Ohka, M.; Kobayashi, H.; Takata, J.; Yamano, M. & Nasu, Y. (2006). Contribution to the development of contact interaction-based humanoid robot navigation system: Application of an optical three-axis tactile sensor, *Proceeding of 3rd International Conference on Autonomous Robots and Agents (ICARA2006)*, pp. 63-68, ISBN-10: 0-473-11566-2, ISBN-13: 978-0-473-11566-1, Palmerston North, Dec. 2006, Massey Univ. Palmerston North, New Zealand

Harrison, T. H.; Levine, D. L. & Schmidt, D. C. (1997). The design and performance of a real-time CORBA event service, *Proceeding of the OOPSLA'97 Conference*, 1997

Hasunuma, H. (2002). A tele-operated humanoid robot drives a lift truck, *Proceeding of 2002 IEEE Int. Conf. on Robotics and Automation*, pp. 2246–2252, 2002

Haykin, S. (1999). *Neural Networks a Comprehensive Foundation*, Toronto, Prentice Hall International

Hirai, K.; Hirose, M.; Haikawa, Y. & Takenaka, T. (1998). The development of Honda humanoid robot, *Proceeding of IEEE Int. Conf. on Robotics & Automation*, pp. 1321-1326, Leuven, Belgium, 1998

Inaba, M.; Igarashi, T.; Kagami, S. & Inoue, H. (1998). Design and implementation of a 35 d.o.f full-Body humanoid robot that can sit, stand up, and grasp an object, *Journal Advanced Robotics*, Vol. 12, No.1, pp. 1-14

Kaneko, S.; Nasu, Y.; Yamano, M.; Mitobe, K. & Capi, G. (2005). Online remote control of humanoid robot using a teleoperation system and user interface, *WSEAS Transaction on Systems*, Issue 5, Vol. 4, May 2005, pp.561-568, ISSN 1109-2777

Michalewich, Z. (1994). *Genetic Algorithms + Data Structures = Evaluation Programs,* Springer-Verlag

Mita, T.; Yamaguchi, T.; Kashiwase, T. & Kawase, T. (1984). Realization of high speed biped using modern control theory, *Int. Journal Control,* Vol. 40, (1984), pp. 107-119

Mowbray, T. J. & Ruh, W. A. (1997). *Inside CORBA: Distributed Object Standards and Applications*, Addison-Wesley, 1997

Nasu, Y.; Kaneko, S.; Yamano, M.; Capi, G. & Nahavandi, S. (2003). Application of a CORBA-based humanoid robot system for accident site inspection through the internet, Proceeding of *7th WSEAS International Conference on Systems*, CD-ROM Proceedings, 6 pages, Corfu Island, Greece, July 7-10, 2003, *Computational Methods in Circuits and Systems Applications*, WSEAS Press, pp.177-184

Neo, E. S.; Yokoi, K.; Kajita, S.; Kanehiro, F. & Tanie, K. (2002). Whole body teleoperation of a humanoid robot -Development of a simple master device using joysticks-, *Proceeding of Int. Conf. on Intelligent Robotics and Systems (IROS)*, 2002

Ohka, M.; Kobayashi, H and Mitsuya, Y. (2006). Sensing precision of an optical three-axis tactile sensor for a robotic finger", *Proceeding of 15th RO-MAN2006*, pp 220-225, ISBN 1-4244-0565-3, Hatfield, U.K, 2006

Open Management Group, "UML Resource Page", http://www.omg.org/uml/

Open Management Group, Welcome to the OMG's CORBA Website, http://www.corba.org/

Open Management Group, The Object Management Group, http://www.omg.org/

Pancerella, C. M. & Whiteside, R. A. (1996). Using CORBA to integrate manufacturing cells to a virtual enterprise, *Proceeding of Plag and Play Software for Agile Manufacturing*, November 1996

Roussel, L.; Canudas-de-Wit, C. & Goswami, A. (1998). Generation of energy optimal complete gait cycles for biped robots, *Proceeding of IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 2036-2041

Silva, F. M. & Machado, J. A. T. (1999). Energy analysis during biped walking, *Proceeding of IEEE Int. Conf. On Robotics and Automation*, pp. 59-64, 1999

Takanishi, A.; Ishida, M.; Yamazaki, Y. & Kato, I. (1990). A control method for dynamic biped walking under unknown external force, *Proceeding of IEEE Int. Workshop on Intelligent Robots and Systems*, pp.795-801, 1990

Takeda, K.; Nasu, Y.; Capi, G.; Yamano, M.; Barolli, L. & Mitobe, K. (2001). A CORBA-based approach for humanoid robot control, *Industrial Robot: An International Journal*, Vol. 28, No. 3, pp. 242-250

Uno, Y.; Kawato, M. & Suzuki, R. (1989). Formulation and control of optimal trajectory in human multijoint arm movement, *Journal Biol. Cybernet*, Vol. 61, (1989), pp. 89-101

Vinoski, S. (1997). CORBA: Integrating diverse applications within distributed heterogeneous environments, *IEEE Communications Magazine*, Vol.14, No.2, pp. 1-12, February 1997

Vukobratovic, M.; Borovac, B.; Surla, D. & Stokic, D. (1990). *Biped Locomotion, Dynamics, Stability, Control and Application.* Berlin, Springer-Verlag

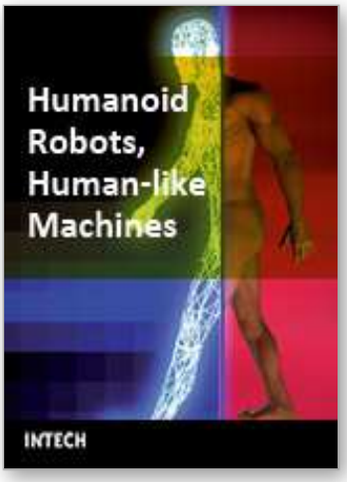
Whiteside, R. A.; Pancerella, C. M. & Klevgard, P. A. (1997). A CORBA-based manufacturing environment, *Proc. of the Hawaii International Conference on Systems Sciences*, Jan. 1997

XEROX PARC, http://www.parc.xerox.com/parc-go.html

XEROX PARC, "Inter-Language unification", ftp://ftp.parc.xerox.com/pub/ilu/

Yokoi, K.; Nakashima, K.; Kobayashi, M.; Mihune, H.; Hasunuma, H.; Yanagihara, Y.; Ueno, T.; Gokyuu, T. & Endou, K. (2003). A tele-operated humanoid robot drives a backhoe in the open air, *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, 2003

Yu, Z.; Nasu, Y.; Nakajima, S. & Mitobe, K. (2001). Development of position measurement system in wide-area using ultrasonic receiver Net, *Journal of Japanese Society of Precision Engineering*, vol.67, no.5, 2001,pp. 764-769, (in Japanese)



Humanoid Robots, Human-like Machines

Edited by Matthias Hackel

ISBN 978-3-902613-07-3

Hard cover, 642 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

In this book the variety of humanoid robotic research can be obtained. This book is divided in four parts: Hardware Development: Components and Systems, Biped Motion: Walking, Running and Self-orientation, Sensing the Environment: Acquisition, Data Processing and Control and Mind Organisation: Learning and Interaction. The first part of the book deals with remarkable hardware developments, whereby complete humanoid robotic systems are as well described as partial solutions. In the second part diverse results around the biped motion of humanoid robots are presented. The autonomous, efficient and adaptive two-legged walking is one of the main challenge in humanoid robotics. The two-legged walking will enable humanoid robots to enter our environment without rearrangement. Developments in the field of visual sensors, data acquisition, processing and control are to be observed in third part of the book. In the fourth part some "mind building" and communication technologies are presented.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen