Real-time Message Network System for a Humanoid Robot

Sang-Min Ahn*, Jung-Sik Gong**, Bo-Hee Lee***,
Jin-Geol Kim**** and Uk-Youl Huh*****

* School of Electrical Engineering, Inha University, Incheon, Korea
(Tel : +82-32-860-8923; E-mail: ahns@huro.inha.ac.kr)
** Department of Automation, Inha University, Incheon, Korea
(Tel : +82-32-860-8923; E-mail: tempus@dreamwiz.com)
*** Department of Electrical Engineering, Semyung University, Chechon, Korea
(Tel : +82-43-649-1305; E-mail: bhlee@semyung.ac.kr)
**** School of Electrical Engineering, Inha University, Incheon, Korea
(Tel : +82-32-860-7384; E-mail: john@inha.ac.kr)
***** School of Electrical Engineering, Inha University, Incheon, Korea
(Tel : +82-32-860-7394; E-mail: uyhuh@inha.ac.kr)

Abstract: This paper deals with the real-time message network system by a CAN (controller area network) based on the real-time distributed control scheme to integrate actuators and sensors in a humanoid robot. In order to apply the real-time distributed processing for a humanoid robot, each control unit should have the real-time efficient control method, fast sensing method, fast calculation and real-time valid data exchange method. Moreover, the data from sensors and encoders must be transmitted to the higher level of control units in maximum time limit. This paper describes the real-time message network system design and the performance of the system.

Keywords: Humanoid robot, Real-time network system, Real-time distributed control system

1. INTRODUCTION

Humanoid robots are one kind of biped robots, having the shape and ability like human being. They are well suited for working in the environment originally designed for people and coexisting with people friendly. But biped robots are one of the hardest to control, caused by instability especially during dynamic walking, and it is a challenging problem to control them. This is the reason why many researchers are working currently on the study and the development of biped robots [1-4]. The ideal humanoid robots are able to coexist and collaborate with humans and perform the difficult and dangerous works that humans can't do [5]. In order to do these tasks, humanoid robots must have autonomous system architecture [6]. To give free mobility, their basic control system, servo system, communication system and power supply system should be mounted on the body structure. The autonomous characteristic needs the high performance of the control system. Thus the autonomous robot becomes the huge control system. It is difficult to install control units on a body. Therefore the robot specification has to satisfy both the performance and structure of the control system. In this study we will suggest a distributed control system [7-8].

Since the system needs efficient control method, fast calculation, fast communication and real-time data exchange, we suggest a new real-time message control system for a humanoid robot based on real-time distributed control system with CAN. All this process is verified by experiment.

2. BASIC CONFIGURATION

Fig. 1 shows the basic configuration of a humanoid robot with 22 DOFs, which are 6 in each leg, 3 in each arm, 2 in main body and 2 in head. Each of joints has its own control units to drive the actuator, to acquire data of sensors and to communicate with main controller.

In order to realize the real-time distributed control, two kinds of communication media are employed. One is the wireless LAN used to connect PC simulator with user interface module in a body of the robot, and the other is the CAN, which is for communicating main controller with joint controllers. Force sensors are mounted on the feet to measure the ground reaction force and then calculate actual ZMP (zero moment point) position. Two acceleration sensors are mounted at a robot foot to measure the acceleration. And one inclinometer is mounted at a robot trunk to measure the orientation.

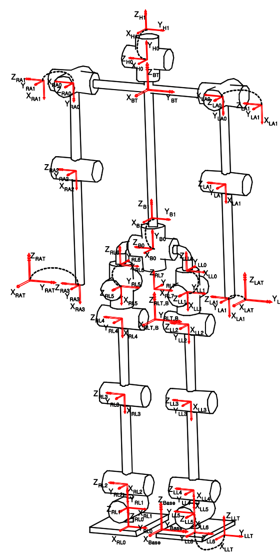

Fig. 1 Coordination of a humanoid robot

Table 1 shows the brief specifications of a humanoid robot and Table 2 presents the brief specifications of a user interface module in a humanoid robot. It contains a 32-Bit embedded main controller with 64Mbyte SDRAM for supporting a humanoid robot simulator and manual user command. Table 3 shows the brief specification of a main controller in a humanoid robot. It includes a high speed 32-Bit DSP and extra external memories for calculating kinematics, ZMP and path planning.

Table 1 Specification of a humanoid robot

Humanoid Robot	
Height	825 mm
Weight	6.84 kg
Width	330 mm
DOFs	22
Foot	135 mm
Sensors	force, acceleration, gyro, inclinometer, etc…

Table 2 Specification of user interface module

User interface module	
CPU	Intel XScale 400Mhz
Memory	64Mbyte SDRAM
OS	Linux with the GUI interface
Peripherals	USB 1.0, Wireless 802.11B, Ethernet, CF Memory, Touch screen, External CAN module, PCMCIA, etc…

Table 3 Specification of main controller

Main controller	
CPU	High speed 32-Bit DSP 225MHz(1G FLOPS)
Memory	External 512KByte SBSRAM, External 16MByte SDRAM
Peripherals	I2C module, SCI module, ADC module, etc…

Table 4 shows the brief specifications of message controller. It contains a high speed 16-Bit DSP(digital signal processor) and 1-MByte external memory for save and schedule information.

Table 4 Specification of message controller

Message controller	
CPU	High speed 16-Bit DSP 150Mhz
Memory	Internal FLASH memory, SRAM 1MByte
Peripheral unit	Internal CAN module, External memory interface, ADC module, Timer module, etc…

Table 5 shows the brief specification of a local controller. It include high speed 16-Bit DSP for control each joint and owns sensors.

Table 5 Specification of local controller

Local controller	
CPU	High speed 16-Bit DSP 150MHz
Memory	Internal FLASH memory, Internal SRAM
Peripheral unit	Internal CAN module, ADC module, Timer module, DC-Servo motor driver, etc…

To support easy user interface, Linux operating system is ported to user interface module. And user interface module is connected with main controller for control a humanoid robot.

3. THE DISTRIBUTED ARCHETECTURE

To realize the autonomous walking, a humanoid robot should have the modules: user interface, intelligent behavior, path planner, stereo vision and dynamics controller for control legs and arms as shown in Fig. 2.

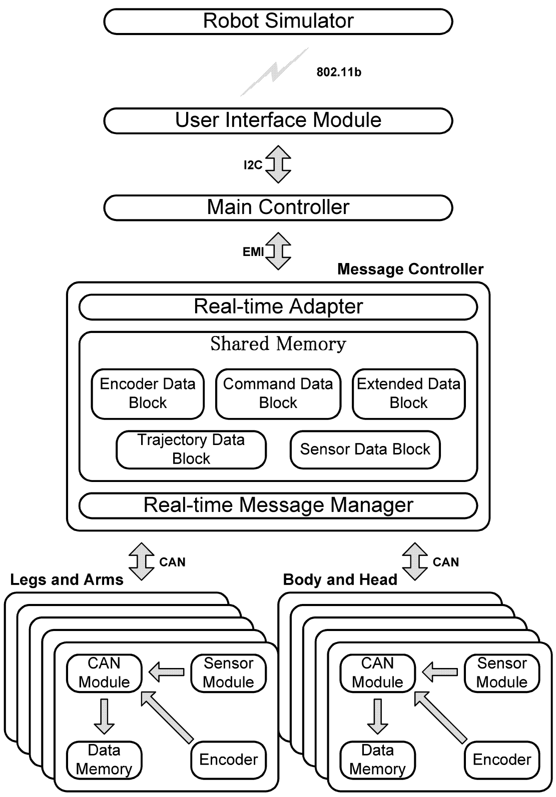

Fig. 2 Distributed architecture of a humanoid robot

3.1 Robot simulator

Robot simulator is based on PC environment. It is connected with user interface module in a humanoid robot using wireless 802.11b. Robot simulator is based on 3D environment and it can support dynamics simulation, trajectory simulation, path planning and ZMP simulation. It can access real humanoid robot using robot API.

3.2 User interface module

User interface module contains one CPU and many kinds of peripherals. 32-Bit low power CPU is used to interface with user, control overall operation, and its companion 32-Bit. DSP is able to do fast computation in calculation of inverse kinematics and trajectory planning job. Basically it has 64M SDRAM, additional memory can be extended using Multi Media Card. Host CPU and DSP use HPI (host port interface) protocol to transmit and receive data fast. To communicate with user interface module, main controller has wireless LAN, so user can send target position and data to the robot.

3.3 Main controller

Fig. 3 shows the picture of main controller. Main controller contains high speed 32-Bit DSP with external SRAM and external SDRAM. Main controller is most important part of a humanoid robot control system. Main controller has special structure for robot control. That is the plug-in manager [11]. Plug-in manager supports basic environment for robot control plug-ins (intelligent behavior plug-in, robot stability control plug-in, vision control plug-in, ZMP control plug-in, trajectory generator plug-in, API plug-in and so on). Each plug-in operates independently each other but they can share own data using shared memory block in message controller.

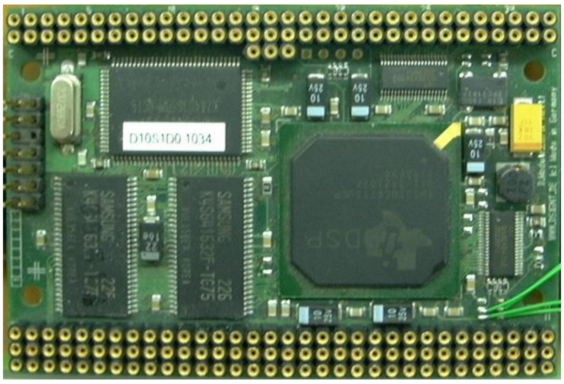

Fig. 3 Main controller

3.4 Message controller

Message controller contains high speed 16-Bit DSP and 4MByte high speed external SRAM. Fig. 4 shows the picture of message controller. It communicates with local controllers in lower layer using internal CAN module and main controller in upper layer using internal real-time adapter. This controller can control message stream for transmitting the messages to its own destinations in dead time. And it also can saves data in the messages to shared memory block for support effective real-time distributed control system.

Fig. 4 Message controller

3.5 Local Controller

Fig. 5 shows the picture of local controller. High speed 16-bit DSP mounted in local controller. It have the ability to control own actuator. It also has the CAN module, A/D converter, PWM unit, pulse encode unit and so on. Additionally, it is equipped with interfacing circuit related with DC motor. The command data from main controller can be received through by CAN and the local DSP controls its motor and sensors. Two or more local controllers will have force sensor module and a gyro-sensor interface module. Local controller gets sensor data from sensors every 1ms. And the data transmit to message controller every 10ms.

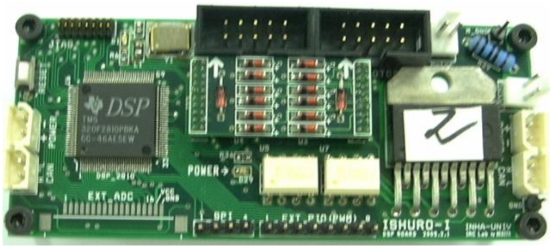

Fig. 5 Local controller

4. REAL-TIME MESSAGE NETWORK SYSTEM OF A HUMANOID ROBOT

To increase stability and accuracy of robot, a robot control system must be guarantee the data transfer stability and data exchange in time limitation. But lots of low level controllers and sensors create many problems in data communication. To solve the problems, the system needs BUS type data communication method, real-time data exchanges method, multi channel communication method and high speed processing method. So we design the real-time message network system using the CAN for supporting these methods based on high speed DSP.

In order to make good humanoid control system, five characteristics have been considered.

1. Local controller must receive trajectory data for next position control in trajectory data refresh in 10ms.
2. After tracking the trajectory data, actual position data must be transmitted in 10ms.
3. All sensor data in local controllers are sent to the main controller through by message controller in 10ms.
4. Manual command from user must be transmitted to the local controllers anytime.
5. Other parts (vision part, intelligent part, etc…) in a humanoid robot system can access the shared memory block in message controller.

4.1 Trajectory data control

Local controller has to receive one message by message controller every 80ms. One message has eight byte data filed and it can include eight trajectory data. Local controller controls the actuator using extra encoder with in 1ms. And one trajectory data is refreshed every 10ms. If the local controller finishes joint control every 10ms, it send actual joint position data to message controller immediately using the CAN. Fig. 6 shows the block diagram of trajectory data stream.

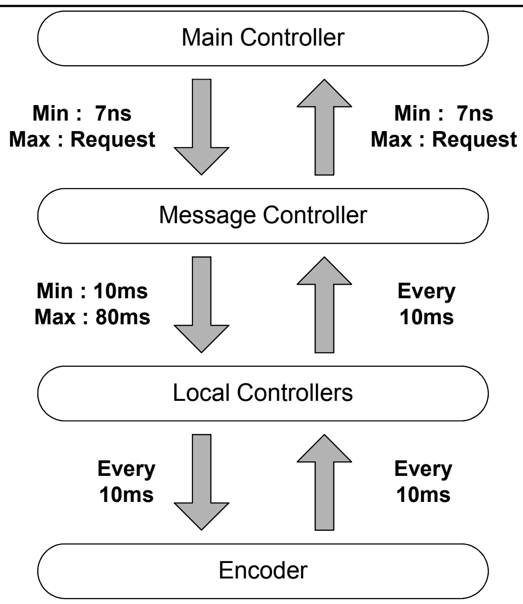

Fig. 6 Block diagram of trajectory data stream

4.2 Sensor data control

Local controllers in each joint have sensors for sensing some information: force, acceleration, shape of ground, torque of motor and so on. Local controller reads data from sensors using internal ADC in every 1ms and save the data in memory. When local controller has ten data, it starts filtering using saved sensing data. After filtering, it transfers from data format to message format and sends the message to message controller immediately using CAN. These processes occur in every 10ms. Fig. 7 shows the block diagram of sensor data stream.

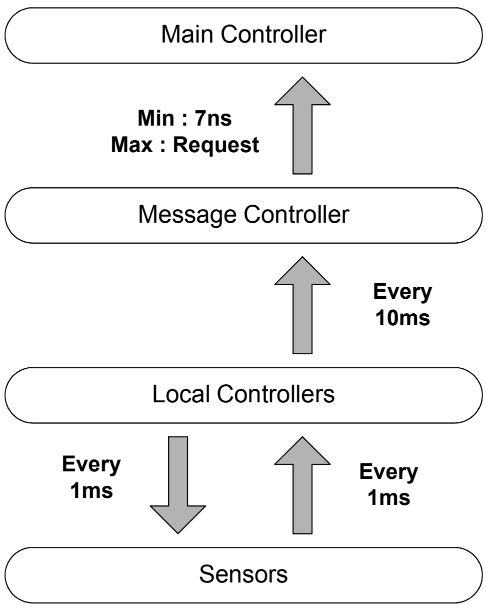

Fig. 7 Block diagram of sensor data stream

4.3 User command data control

User can send user's command to a humanoid robot anytime using the user interface module. This command is very important. Robot must execute user's command immediately. But this command creates many unfixed problems like increasing instability of a humanoid robot and the message scheduling crack. Thus the maximum time limitation is needed to avoid above problems. In this system, we select the maximum time limit of 10ms. And we use independent schedule method for processing the user's command message. Fig. 8 shows the block diagram of user command data stream. It describes that user can send command to a humanoid robot, but the system does not react immediately.

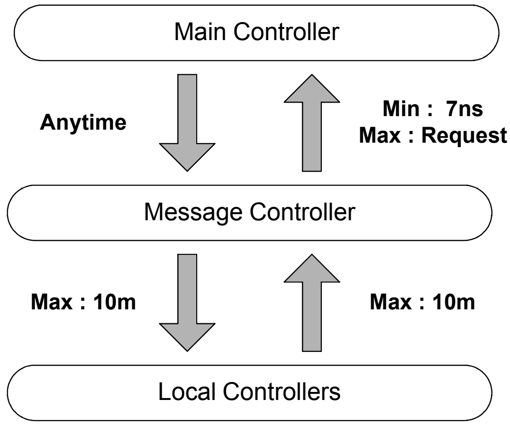

Fig. 8 Block diagram of user command data stream

4.4 Message management block

Message controller between main controller and local controller must have huge and fast memory for saving many data in messages. And it has high-speed 16-Bit DSP for real-time message scheduling. Data extract from message is saved to shared memory block by message management block. If main controller requests some data using real-time adapter, message management block sends the data to main controller immediately. This block can communicate with real-time adapter, and the real-time adapter connects to main controller using EMI (external memory interface). Many messages created by local controllers are transmitted to message manage block through the CAN. Message manage block is receives messages in every 10ms. And it sends eight trajectory data in one message to local controller in every 80ms.

5. EXPERIMENT

After design and implementation, we experiment the system. Fig. 9 shows the picture of timing of messages in one time slot. In this picture, trajectory data is transmitted from main controller to local controllers every 80ms. And main controller receives sensor data from local controller every 10ms. Table 6 shows message description of a humanoid robot. And the number of messages means own priority. We use oscilloscope and timer in DSP to measure the message time in data exchange. A message exchange time in CAN spends about 117μ sec when transmission speed is applied in 1-Mb/s. And one message has eight bytes in CAN message field. Table 7 shows result of experiment. The total length of twelve messages exchange time is 1.412ms. And the length of feedback sensor data exchange time is 1.412ms. Trajectory data exchange time and sensor data exchange time are very similar. Because both data create by twelve local controller and basic exchange time is same. The length of User command data exchange time is 117us.

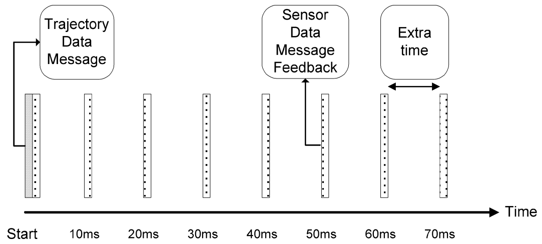

Fig. 9 Timing of messages in one time slot

Table 6 Messages description of a humanoid robot

	MESSAGE	SIZE (Byte)	FROM	TO
1	Motor 1-trajectory	8	Message controller	LC-1
2	Motor 2-trajectory	8		LC-2
3	Motor 3-trajectory	8		LC-3
4	Motor 4-trajectory	8		LC-4
5	Motor 5-trajectory	8		LC-5
6	Motor 6-trajectory	8		LC-6
7	Motor 7-trajectory	8		LC-7
8	Motor 8-trajectory	8		LC-8
9	Motor 9-trajectory	8		LC-9
10	Motor 10-trajectory	8		LC-10
11	Motor 11-trajectory	8		LC-11
12	Motor 12-trajectory	8		LC-12
13	Sensor 1	8	LC-1	Message Controller
14	Sensor 2	8	LC-1	
15	Sensor 3	8	LC-1	
16	Sensor 4	8	LC-1	
17	Sensor 5	8	LC-1	
18	Sensor 6	8	LC-1	
19	Sensor 7	8	LC-1	
20	Sensor 8	8	LC-1	
21	Sensor 9	8	LC-1	
22	Sensor 10	8	LC-1	
23	Sensor 11	8	LC-1	
24	Sensor 12	8	LC-1	
25	User Command	8	Message Controller	LC 1-12

Table 7 Measurement of message exchange time

MESSAGE	PERIOD	TOTAL LENGTH
Motor trajectory	80ms	1.412ms
Sensor	10ms	1414ms
User command	anytime	117us

6. CONCLUSION

In this paper, the real-time message network system for a humanoid robot is suggested. A humanoid robot control system consists of a robot simulator, a user interface module, a main controller, a message controller and local controllers. A robot simulator connects with user interface module through the wireless 802.11b. And the robot simulator sends API data and user command to user interface module. After data access process is finished, user interface module sends command data to the main controller using I2C. And then, main controller communicates with local controllers through the CAN. And they exchange information in time. In this case, the real-time message network system guarantees data exchange with no errors and stability of a humanoid robot.

Designed message exchange method is suggested and implemented to a humanoid robot system. Also message exchange method is verified by experiment. In the future, we will focus on realization of real-time control in a humanoid robot system using a real-time operating system.

ACKNOWLEDGMENTS

This work was supported by Grant No. R01-2003-000-10364-0 from Korea Science & Engineering Foundation.

REFERENCES

[1] K. Hirai and M. Hirose, "The Development of Honda Humanoid Robot," *ICRA'98*, pp.1321-1326
[2] K. Hirai, "The Honda Humanoid Robot: Development and Future Perspective," *Industrial Robot: An International Journal*, Vol.26, No.4, 1999, pp.260-266
[3] Y. Sakagami and R. Watanabe, "The intelligent ASIMO: System overview and integration," *Proceedings of the IEEE Intl. Conference*, 2002 Intelligent Robots and Systems
[4] M. Fujita and Y. Kuroki, "A Small Humanoid Robot SDR-4X for Entertainment Applications," *Proceedings of the IEEE Intl. Conference*, 2003 Advanced Intelligent Mechatronics
[5] N. Kanehira and T. Kawasaki, "Design and Experiments of Advanced Leg Module for Humanoid Robot Development," *Proceedings of the IEEE Conference*, 2002 Intelligent Robots and Systems
[6] Y. Sakagami and R. Watanbe, "The intelligent ASIMO: System overview and integration," *Proceedings of the IEEE Conference*, 2002 Intelligent Robots and Systems.
[7] Shi. Zonging and Xu Wenli, "Distributed Hierarchical Control System of Humanoid Robot THBIP-1," *Proceedings of the 4th World Congress*, 2001 Intelligent Control and Automation.
[8] M. Fujita and Y. Kuroki, "Autonomous Behavior Control Architecture of Entertainment Humanoid Robot SDR-4x," *Proceedings of the IEEE Conference*, 2003 Intelligent Robots and Systems.
[9] Tindell, K., Burns, A. and Wellings, A.: "Calculating Controller Area Network (CAN) Message Response Time," *Control Engineering Practice*, Vol. 3, No. 8, pp. 1163-1169, 1995
[10] Tindell, K. and Burns, A. "Guaranteed Message Latencies for Distributed Safety Critical Hard Real Time Networks," *Technical Report YCS 229*, Department of Computer Science, University of York, England, May 1994.
[11] Kazuhito Yokoi, Fumio Kanehiro, Kenji Kaneko, Shuuji Kajita, Kiyoshi Fujiwara and Hirohisa Hirukawa. "Experimental Study of Humanoid Robot HRP-1S" *International Journal of Robotics Research*, Apr, 2004.