Design and Implementation of Humanoid Robot Controller Based on CAN

Hua Zhong\textsuperscript{1,2}, Zhenwei Wu\textsuperscript{1}
\textsuperscript{1}Shenyang Institute of Automation
Chinese Academy of Sciences
Shenyang, China
hzhong@sia.cn

Bin Li\textsuperscript{1}, Chunguang Bu\textsuperscript{1}, Yanjie Li\textsuperscript{1,2}
\textsuperscript{2}Graduate School of the Chinese Academy of Sciences
Beijing, China

Abstract—Aiming at Humanoid Robot structure and requirement for control performance, this paper designs and realizes joint controller based on CAN, and structures an effective credible control system by connecting controllers, force sensors and harmonious level computer together. This paper includes mainly: overall structure of the system, design and implement of hardware and software, topology of control system, and bring forward some imaginations to enhance performance of the system.

Keywords- CAN; humanoid robot; MCU

I. INTRODUCTION

Robotics is an integrated subject that includes mechanics, electronics, computer, automation, sensor and artificial intelligence. Humanoid robot embodies centrally up-to-the minute research production. Humanoid robot has own characteristic compared with other robot: removability, can arrive working place, enlarge workspace; more degree of free (DOF), includes double arms, neck, waist, dual legs, can complete more dexterous task; ability of vision disposal, can identify object, achieve arm-eye harmony; ability of hearing disposal, can carry out action according to vocal command. Humanoid robot mentioned in this paper has 21 DOF. Besides 3 DOF of mobile plant, other 18 DOF must be connected together, and controlled harmoniously. So the control system needs higher reliability and real time. As a kind of field bus applied widely in industrial field, CAN (Controller Area Network) \cite{1} has lower cost, better reliability, more simple configuration and so on, that makes it suit for control system of humanoid robot.

II. OVERALL STRUCTURE OF SYSTEM

A. Mechanical Structure

Humanoid robot in this paper includes an all-orientation wheel mobile flat with 3 DOF, two arms, neck, waist and trunk. Arm composes of shoulder with 3 DOF, elbow with 2 DOF, wrist with 2 DOF. Neck has 2 DOF that realizes the pitching and gyration of the head. Waist \cite{2} has 2 coupling DOF that realizes the pitching and lean of the waist. The waist is attached to the underside of the trunk, and the neck and arms is attached to the upside of the trunk. Two 6-dimension force sensors are fixed at the top of the arms. The appearance of humanoid robot is shown as Fig. 1.

B. Selection of Field Bus

The whole system has 21 DOF. Besides 3 DOF of mobile plant, other 18 DOF must be connected together, and controlled conformably. So higher reliability and real time of the control system be requested. Conventional centralized control system needs more hardware and line, so it is less credible. Distributed control system based on bidirectional, serial, multi-node field bus has applied widely in control system with the development computer, control, information, network and microelectronics.

CAN is put forward and applied in automotive industry by BOCH in the end of 80’s. CAN has particular advantage relative to other field bus: its structure is simple, equipment can be increased or decreased arbitrarily in the range of permission; its cost is low, ordinary twisted-pair can satisfy requirement on the occasion of low speed or short line; its data field includes 8 bytes that can satisfy request of control command, status and testing data; its speed is so high that it can reach 1M b/s within 40 meters; character of real time is ensured by adopting short data frame; reliability of communication is guaranteed by adopting Cyclic Redundancy Check (CRC) and function of error handling; the message is recognized according to not node.
A. **Kernel Module**

resistances restrain reflection of the running, twisted-pair.

PLC

input

output

, ,

Figure 2. Structure diagram of control system of humanoid robot based on CAN

address but identifier in the message. Above advantage satisfy the request of the control system of humanoid robot, so we take CAN as the foundation of the control system of humanoid robot.

C. **Implementation of CAN**

There are three kind of method to implement CAN: micro controller (MCU or DSP) + CAN controller + CAN transceiver; micro controller with CAN controller + CAN transceiver; CAN serial I/O component + CAN transceiver. This system is implement by the way of micro controller with CAN controller + CAN transceiver. The MCU is 16-bit lower-power consumption microcontroller of FUJITSU Inc-MB90F549G [3] in which 2 CAN controllers are integrated. The CAN transceiver is PCA82C250 of PHILIPS Inc [4], [5]. The physical medium is ordinary twisted-pair.

All nodes that include the controller of each joint, two force sensors on the tip of arms, harmony level computer are connected together by means of serial bus. As harmony level computer, the PC-104 computer takes charge of task planning, resolution of inverse kinematics and inverse dynamics, communication and so on. As execution level controller, the controller of each joint takes charge of the function shown in the following content. Appropriate resistances are linked at the end of CAN bus so as to restrain reflection of the signal on bus [5]. The structure diagram of control system is shown as Fig. 2.

III. **HARDWARE DESIGN OF CONTROLLER**

The controller of each joint consist of MCU, pulse counter, D/A converter, CAN transceiver and so on. The controller receives the position command transmitted by the harmony level computer and read the position feedback information from motor code, then calculates the control value to drive DC motor through servo amplifier. The block diagram of controller is shown as Fig. 3.

A. **Kernel Module of the Controller**

MCU is the kernel module of controller, which takes charge of servo control, feedback of the information about running, fault checking, fault warning, fault handling and so on. The MCU selected in this system is MB90F549G of FUJITSU Inc. It includes 256K bytes Flash ROM which support In System Program (ISP), 2 CAN controllers, 6K bytes RAM, 8 8-bit or 16-bit A/D converter, 2 16-bit reload timer, 4 programmable pulse generator (PPG), 8 external interrupt, 2 output compare, 4 input capture.

B. **Servo Drive Module**

All joints of the system are driven by DC motor. In this system, we select brush type PWM servo amplifier of A-M-C Inc to drive motor. The primary characteristic of this servo amplifier is low cost, single supply operation, four quadrant regenerative operation, protection against over-voltage, over-current, over-heating and short-circuits.

C. **Position Detection Module**

Photoelectric coder attached at motor shaft outputs 3 channels pulse signal when motor rotates. A phase and B phase have same frequency, but phase discrepancy is 90°. C phase output a pulse per circle of motor. We can get the angle displacement of motor by dealing with the pulse signal of A phase and B phase. We implement above-mentioned function by writing a corresponding program into a chip of Field Programmable Gate Array (FPGA). Firstly, the counter takes count of at rising edge and falling edge of this two phase in a pulse period, namely 4 multi-frequency. Then the rotational direction of motor is confirmed by relation between A phase and B phase. The counter is added 1 or subtract 1 due to the rotational direction of motor. Lastly, we can get the angle displacement.

D. **Signal Output Module**

The servo amplifier needs analog signals from -15V to +15V to drive motor, so the digital signals from MCU must be transformed the analog signals through D/A converter. Presently, D/A converter can be grouped under serial D/A and parallel D/A according to input interface. The serial D/A has less pins and area compared with the parallel D/A. By adopting the serial D/A, we can reduce the size of controller, failure ratio and power consumption. In this system, 12-bit serial D/A converter MAX531 is selected as the D/A converter [6]. MAX531 can be operated with bipolar supplies (-5V to +5V) by appropriate connection. Its input range is from 0 to 4095, and its output range is

Figure 3. Block diagram of controller
from $-2.048V$ to $+2.047V$. The analog signals are magnified by operation amplifier properly, and then enter the servo amplifier.

E. Communication module

The CAN controller integrated in MCU conforms to CAN Specification Version 2.0 Part A and B. PCA82C250 is selected as the CAN transceiver to realize the electric transform between the CAN controller and the physical medium. The physical medium is ordinary twisted-pair.

IV. SOFTWARE DESIGN OF CONTROLLER

A. Design of Control Program

As the execution level unit, the controllers fulfill function such as servo control, feedback of the information about running, fault checking, fault warning, fault handling. We compile program and debug in integrated debug environment (IDE) offered by FUJITSU Inc. The main body of program is written with C language so as to ensure good performance of maintenance and clear structure. The part that is executed frequently is implemented by embedding assembly sentence into C language to quicken run speed. Position, velocity and acceleration of motor are detected and restricted in an relevant range to ensure the safety of system.

B. Design of Communication Protocol

1) Setting of bit time

The speed of CAN bus can reaches $1M$ b/s (within 40 meter) theoretically, but considering the factor of physical medium, electro-optical isolation component, and the reliability of communication, the speed of this system adopted is $320K$ b/s.

The calculating methods of bit time for different CAN controller may different appreciably. The CAN controller in this system is the one integrated in MCU. Its formula for calculation the speed of bit time is:

$$BT = (3 + TS1 + TS2) \times (PSC + 1) \times CLK$$  \hspace{1cm} (1)

Where $BT$ denotes bit time; $TS1$ denotes the time of phase buffer segment1 and propagation segment; $TS2$ denotes the time of phase buffer segment2; $PSC$ denotes the time quanta of the CAN controller; $CLK$ denotes machine cycle of MCU, $CLK$ equal to 62.5 nanosecond (4 MHz oscillation clock, 4 multiplied PLL clock).

$TS1$, $TS2$ and $PSC$ should satisfy restriction as follow:

When $1 \leq PSC \leq 63$,

$$\begin{align*}
TS1 & \geq 1 \\
TS1 & \geq RSJ \\
TS2 & \geq 1 \\
TS2 & \geq RSJ
\end{align*}$$ \hspace{1cm} (2)

When $PSC = 0$,

$$\begin{align*}
TS1 & \geq 4 \\
TS2 & \geq 1
\end{align*}$$

According to above formulie, we set $PSC$ to 4, $TS1$ to 4, $TS2$ to 3, $RSJ$ to 3 when bit time is 3125 nanosecond (namely speed is 320K b/s).

2) Setting of the identifier

The CAN specification adopts not conventional station address coding, but communication data block coding. The CAN controller of each joint has 8 or more message buffers that can setup identifier respectively to aim at particular purpose, this means that the same CAN node may hold different identifiers. Considering the number of the joint and the capacity of extension, this system adopts CAN specification 2.0 A defined by 11 identifier bits.

In each joint of humanoid robot, a message buffer is setup to receive the command transmitted by the harmony level computer. The synchronization signal transmitted by the harmony level computer to all joint is a special command that must be received by all joint. We make use of the broadcasting characteristic of CAN bus to realize above request. In each joint, a message buffer is setup to receive the synchronization command with the same identification. Some message buffers in the force sensor are setup to transfer the value of force and moment to the harmony level computer and other joint.

3) Settings of the data format

The data transferred on the bus are classified as five type in terms of content and direction. There are three kinds of data down: the command transmitted by the harmony level computer, the synchronization signal transmitted by the harmony level computer to the joint controllers, the command transmitted by the harmony level computer to the force sensors. There are two kinds of data up: the status message transmitted by the joint controllers to the harmony level computer, the value of force and moment transmitted by the force sensors to the harmony level computer.

In data frame, the length of data field is 0 to 8 bytes. The first byte denotes the type of the data, the second byte denotes the destination of the data, and else bytes denote the numerical value.

4) Disposal of Bus Off

The node on CAN bus has four kind of bus status: Error active, Warning, Error passive, Bus off. The node status changes as a result of the change of value of the receive error counter (REC) and the transmit error counter (TEC). The node status change diagram is shown in Fig. 4.

The Error Active is the normal status of node. If the number of the node error accumulates to some degree, the node will come into Warning. Error Passive, and Bus Off orderly. In Bus Off the node neither transmits data nor receives data. There are two kind of method to make the node return Error Active from Bus Off: hardware reset and setting the bus operation stop bit (HALT) to 0. The uncontrollable status of motor will not appear by adoption latter method. After the node return Error Active, a
message will be transmitted by the node to the harmony level computer. The harmony level computer reminds the operator to adopt corresponding step.

V. CONCLUSION

A controller is designed and implemented based on the framework characteristic and request of performance of humanoid robot, and a control system based on CAN bus is constituted. This system has run reliably, the control periods of the harmony level computer is 20 millisecond, servo periods of the joint controller is 2 millisecond, the rise time of respond for step is about 16 millisecond, those can satisfy the request of system. For consummating function and improving performance, we consider adopting faster MCU or substituting MCU with DSP; ameliorate control arithmetic by introducing forward back and compensation; consummate communication protocol so as to strengthen the communication capability of all nodes and realize the capability of distributed computing.

Figure. 4. Node Status Transition Diagram

REFERENCES