Applied Mechanics and Materials *Online: 2010-11-11*
ISSN: 1662-7482, Vol. 42, pp 307-312
doi:10.4028/www.scientific.net/AMM.42.307

Control System Design Based on CANopen Network for Multi-legged Robot with Hand-fused Foot

Yuxiao Zhang[1,a], Xianqun Zeng[1,b] and Xinjie Wang[1,c]

[1]Department of Electromechanical Science and Engineering, Zhengzhou University of Light Industry, Zhengzhou, Henan, 450002, P.R. China

[a]Jasonzyx@126.com, [b]zengxianqun012345@163.com, [c]weibo1991@126.com

Keywords： CANopen, Robot, Control system, Closed-loop control

Abstract： In this paper, a distributed motion control system is designed and achieved for the modular multi-legged robot with hand-fused foot based on CANopen network. A communication network is built among modules and joints through CANopen network. Controllers and encoders are used to achieve closed-loop control for each joint. The system has completed real-time multi-jointed linkage. Moreover, experimental results have proved the reliability and validity of this control system.

Introduction

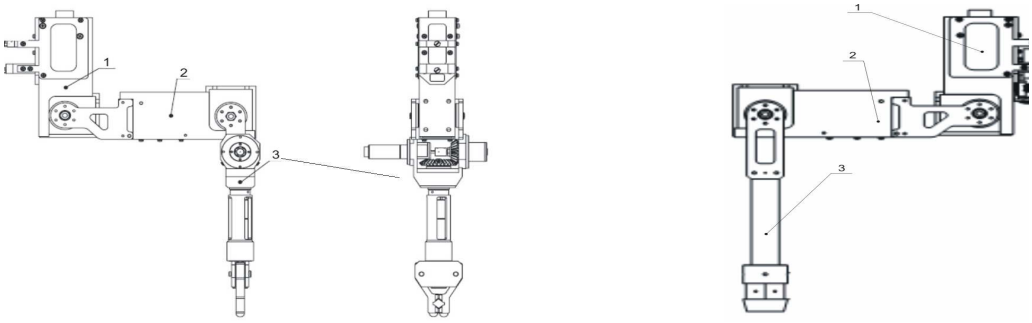
The multi-legged robot with hand-fused foot is created through the work that one leg is fused a shank with hand like foot and fingers like halluces for a common multi-legged robot. This robot could walk like a common multi-legged robot, furthermore, it could capture a target, which extends the function of a multi-legged robot. Modularization of the robot itself required its control system must have extendability, modularity and preferable timeliness.

Many scholars have studied robot's motion control system. Hajime, Kawabata and Lin[1] developed a quadruped robot adopting independently decentralized control strategy. The central controller produced a synchronization clock signal for each leg unit, basing on which the leg unit controller controlled the robot leg's movement independently. Guido, Michele, Giovanni and R.Sinatra[2] made a wall climbing robot adopting master-slave control method. User interface as master controller was supported by a computer and local controller controlled each leg adhering and crawling on the wall. Teresa advanced the concept hierarchical control. From bottom to top, the control system was divided into several levels, which were respectively responsible for joint control, inverse kinematics calculation, shank terminal trajectory generation, gait planning, body motion trail generation[3]. Christophe, Hiroshi and Kunikatsu[4] adopted biologically inspired neural controller to quadruped robot, where gesture and rhythmic movement of each leg could be controlled, and control mode could also be shifted from one to another. Subudhi and Morris[5] completed motion control to a multi-jointed flexible robot manipulator adopting hybrid fuzzy neural control.

At present, the robot control system generally uses two-level distributed control strcture[6]. The upper computer is dedicated for kinematic analysis and motion planning. The lower computer is made up of multiple CPU, and each control a joint. In the paper, we design and achieve a control system based on CANopen network for the multi-legged robot with hand-fused foot.

Leg Structure of Multi-legged robot with Hand-fused Foot

The multi-legged robot with hand-fused foot was employed with module design. Each leg is an independent module, which has 3 driving joints, including hipjoint, thigh and shank, as shown in Fig.1. To achieve hand and foot fusion, one leg is fused a shank, which contains a hand like foot and two fingers like halluces, having two degree of freedom. The shank could be used to capture and walk. When walking, it’s locked, and the robot become a common multi-legged robot.



(a)One leg with a hand-fused foot (b)One common leg

1.Hipjoint 2.Thigh 3.Shank

Fig.1 Leg structure of multi-legged robot with hand-fused foot

Overall Structore

Multi-legged robot with hand-foot fusion adopted modular design, which required that every modular could govern itself separately and could not interfere each other[7]. Therefore, the distributed hierarchical control system based on CANopen network was used to control the robot. Fig.2 is the overall structure of the control system, which shows obviously that the whole control system is divided into 3 levels, including organization level, coordination level, implementation level.

1) Organization level makes real-time control, path planning, forward and inverse kinematics analysis and gait generation by monitoring the environment of the robot.

2) Coordination level could store control instructions from the upper level and transmit them to each joint controller.

3) Implementation level is responsible for receiving instructions from the coordination level and passing them to joint motor after analysis and translation.

Implementation level is made up of 4 independent leg modules and each of them containes 3 control modules. To achieve hand and foot fusion, a shank was fused to Leg 1 with a hand like foot and two fingers like halluces. As a result, the whole control system have 14 independent control modules which need 14 controllers connected to the CANopen network for real-time communication.

Multi-legged robot with hand-fused foot has many joints and every action needs coordination among each joint, which requires the whole control system have high timeliness and reliability. In addition, extendibility of the robot itself demands its control system has a preferable extendibility. If using serial communication such as RS232, RS485, it is very difficult to satisfy these demands. CAN bus has the advantage of high speed, reliability, good extendibility, but itself only contains two bottom layer protocols, physical layer protocol and data link layer protocol. CANopen is a higher level protocol designed for CAN's incompleteness, largely reducing the data load in CAN bus [8-10]. Therefore, CANopen network is used to achieve coordination movement among multi-joints.

Working process of the control system is as followed. As the highest level of the control system, PC synthesizes received information to make inverse kinematics analysis to gain motion parameters of each joint, and then sends control commands to the CAN interface card. CAN interface card distributes these commands to each joint controller after filtering and buffering. Each controller could control its motor to rotate. Through coordination of various joints, the quadruped robot with hand-fused foot could move to target position in the light of specified motion mode.

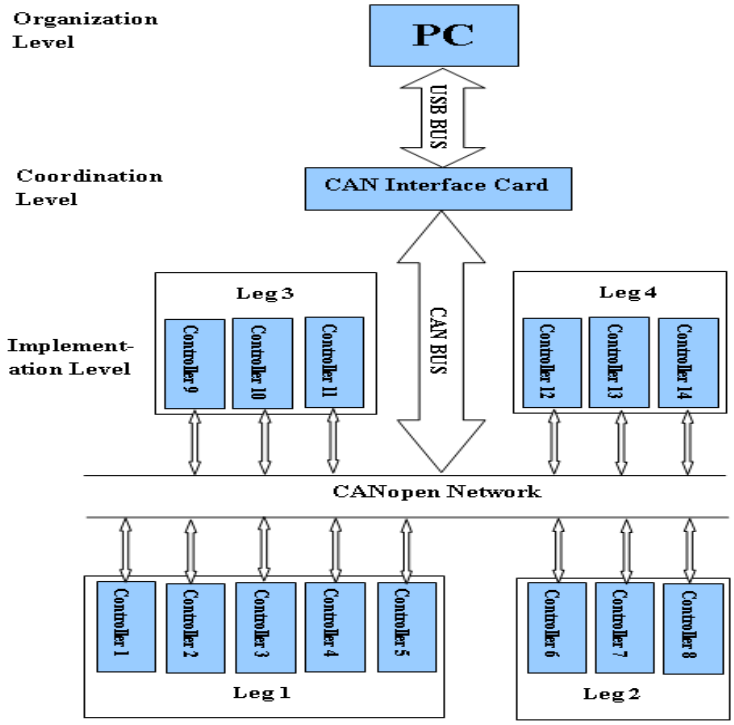


Fig.2 Overall structure of the control system

The Hardware Design and Control

A. CAN Interface Card. The CAN interface card is mainly responsible for linking organization level and implementation level, and transmitting control commands from the organization level to the implementation level after storing. One USB-to-CAN compact was connected to the USB bus. The USB-to-CAN compact integrates a microcontroller, a CAN Controller SJA1000, a RAM and a Flash. The microcontroller system enables reliable, loss-free transmission and reception of messages in CAN networks with both a high transmission rate and a high bus load. In addition, messages are provided with a time-stamp and can be filtered and buffered directly in the USB-to-CAN compact. SJA1000 supports two communication protocols, CAN2.0A and CAN2.0B, and can be effectively responsible for transmitting and receiving the data.

B. Joint Controller. Joint controller could adopt processor, such as MCU, DSP, ARM. The Accelnet series dedicated motor controller ACJ-055-18 was chosen to make the control system more stable and shorten the development cycle. It is a compact, DC powered servo driver, supporting the CANopen distributed control architecture. ACJ-055-18 uses three nested control loops – current, velocity, and position - to control a motor in three associated operating modes. As shown in Fig.3, the position loop drives the nested velocity loop, which drives the nested current loop, which generates the PWM command to drive motor rotating in turn. Encoder assembled in motor could feed actual rotation position, velocity and current back to controller real time, which forms a closed loop control system integrating position, velocity and current.

The controller receives position command from PC over the CAN interface. The internal trajectory generator calculates a trapezoidal motion profile based on trajectory limit parameters. The output of the generator is an instantaneous position command (limited position). In addition, values for the instantaneous profile velocity and acceleration are generated. These signals, along with the actual position feedback, are processed by the position loop to generate a velocity command.

The velocity loop limiting stage accepts a velocity command, applies limits, and passes a limited velocity command to the input filter. The filter then passes a velocity command to the summing junction. The summing junction subtracts the actual velocity, represented by the feedback signal, and produces an error signal. The error signal is then processed using the integral and proportional gains to produce a current command. Programmable digital filters are provided on both the input and output command signals.

The "front end" of the current loop is a limiting stage. The limiting stage accepts a current command, applies limits, and passes a limited current command to the summing junction. The summing junction takes the commanded current, subtracts the actual current (represented by the feedback signal), and produces an error signal. This error signal is then processed using the integral and proportional gains to produce a command. This command is then applied to drive the joint controller.

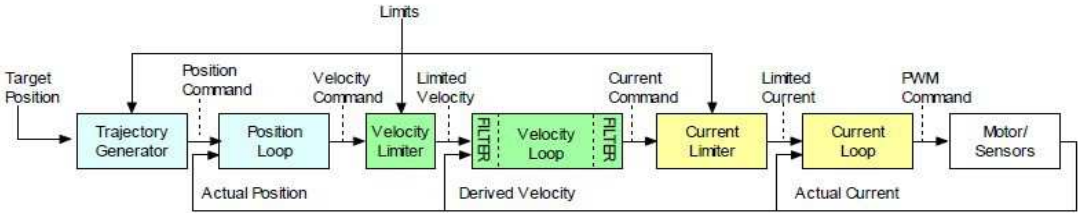


Fig.3 The overall closed-loop control diagram

THE SOFTWARE DESIGN

This chapter has two parts, motor control and CANopen network communication.

CMO(Copley Motion Objects) were used in motor control, which were low level dynamic link library functions provided by Copley amplifier. In the CANopen network, control software needs to be prepared for the motor controllers based on Windows Operation System.

The Copley Motion Objects simplify creation of Windows-based software for the control of Copley Controls amplifiers over a CANopen network. They give programmers direct access to an amplifier's CANopen functions without having to learn the complicated CANopen objects. Copley Motion Objects were built using the Microsoft Component Object Model (COM). Fig.4 illustrates the CMO/ COM model. CMO functions link PC and Copley amplifiers, which translate PC's control commands into motor movement instructions so that the robot can complete certain actions. Fig.5 shows the flow chart of robot joint control system.

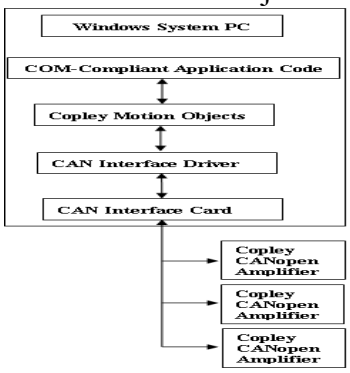


Fig.4 CMO/ COM model

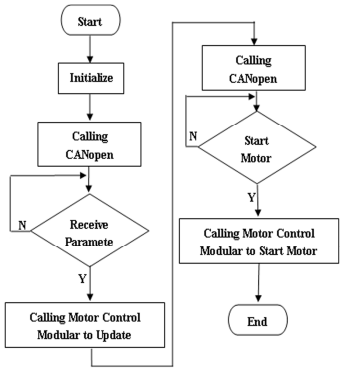
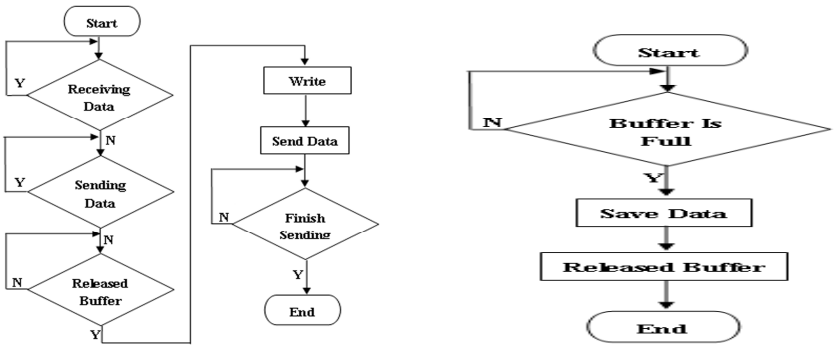


Fig.5. Joint control system flow chart

CANopen communication program was designed with hierarchy. Bottom layer included the use of SDO (Service Data Object) for configuration parameters, such as setting the baud rate, acceptance code filter and output mode. In middle layer, independent CAN controller SJA1000 called bottom functions to initialize parameters and used PDO (Process Data Objects) to send and receive data real-time. Top layer could complete CANopen communication through calling series of functions supplied by middle layer.

Query mode was taken during sending and receiving. As shown in Fig.6, send buffer should be checked whether it was released before sending data. Receive buffer should also be checked whether it was full before receiving.

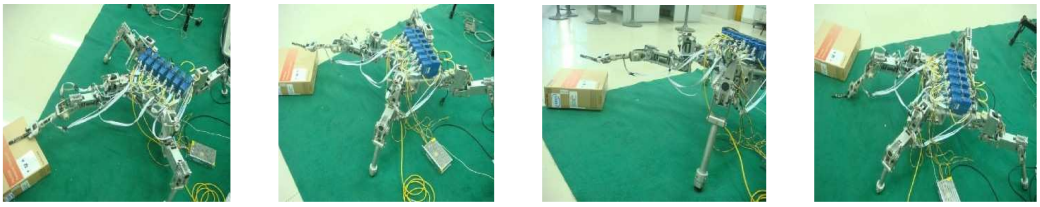


(a)Send data flow chart (b)Receive data flow chart

Fig.6 CANopen communication flow chart

EXPERIMENT RESEARCH

Before communication, motor parameters need to be initialized through the visual software CME2. Then, controller parameters could be calculated. CME2 offers 4 kinds of baud rate: 9600b/s, 19200b/s, 57600b/s and 115200b/s. It was found from many experiments that using low baud rate resulted in very slow performance of CME. During the design of the quadruped robot, a clamping mechanism was fused as a shank in one leg, making the robot have a function with hand and foot fusion, which meant that the leg could lift as a robot arm, capture an object and put it down. As illustrated in Fig.7, Robot captured and transferred the target. Many experimental results showed that the distributed control system based on CANopen network could coordinate the control of various joints precisely, complete multi-joint linkage and achieve hand-foot fusion.



(a) Capture the target (b) Clip the target up (c) Shift the target (d)Place the target

Fig.7 Capture and transfer the target

CONCLUSIONS

The distributed control system used module design based on CANopen network, in which every module and every joint communicated with each other through CANopen network, achieved multi-joint coordinated motion and strengthened reconfigurability and fault tolerance. Hierarchical control made the control accuracy increasing step by step and obtained good consistency and scalability.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (50875246).

REFERENCES

[1] Hajime K, Kawabata K, Lin P C, et al. Leg Synchronization by Distributed Control Structure. The Fourth International Workshop on Advanced Motion Control, Tsu, Mie, Japan,1996

[2] Guido La Rosa, Michele Messina, Giovanni Muscato, R.Sinatra. Mechatronics, 2002, 12: 71～96

[3] Xuedong Chen, Minghao Zhou, “Study on Motion Control System of a Modular Multi - legged

Robot," China Mechanical Engineering, Vol.10, June 2006.
[4] Christophe Maufroy, Hiroshi Kimura, Kunikatsu Takase. Biologically Inspired Neural Controller for Quadruped. IEEE International Conference on Robotics and Biomimetics, Sanya, China,2007
[5] B. Subudhi, A.S. Morris. Soft computing methods applied to the control of a flexible robot manipulator. Applied Soft Computing, 2009, 9: 149～158
[6] Yong Fan, Min Tan, "Robot Controller Status and Prospects," Robots, Vol.21, 1999,21(1) :75～80
[7] Alberto Brunete, Jose EmilioTorre,Miguel Hernand.Multi-Drive Control for In-pipe Snakelike Heterogeneous Modular Micro-Robots. IEEE International Conference on Robotics and Biomimetics,2007
[8] BOTERENBROOD H . CANopen: high-level protocol for CAN-bus [M]. Amsterdam: Nikhef , 2000.
[9] Copley Motion Objects (CMO) Programmer's Guide [Z].2006.
[10] Qingnian Cao, Bo Zhao, Kaiyuan Meng, "CANopen protocol application in industrial control network," Journal of Xi'an Shiyou University(Natural Science Edition), Vol.24, July 2009