Digital communication technology for teaching automatic control: the level control case

Nicolás H. Beltrán, Manuel A. Duarte-Mermoud and Pablo A. Kremer
Department of Electrical Engineering, University of Chile, Santiago, Chile
E-mail: mduartem@cec.uchile.cl

Abstract A prototype application for a communication network between the controller and the sensors in a laboratory plant for teaching purposes using the controller area network (CAN) standard is discussed in detail in this paper. The basis of the CAN digital communication system is revised and results are presented of the designed experiment, carried out by the students, to control the tank level. It is concluded that the introduction of modern communication technology in teaching process control as part of the electrical engineering curriculum is feasible.

Keywords digital control; controller area network; level control; teaching automatic control

Advances in telecommunications technologies are permeating other areas of engineering like control systems, where analogue communications between components of a plant are changing to digital. The classic 4–20 mA system which communicates the sensor or the actuator to the controller device in multidrops control systems is being replaced in the industry by a communication bus making possible the use of distributed control systems

Several standards for digital communications in a plant have been proposed, most of them related to different manufacturers. The Fieldbus standard proposed by the Fieldbus Foundation features distributed functions in the field devices and according to its proponents, it is an open standard which will allow a wide diffusion.[1] One significant difference with semi-distributed schemes is related to the feedback loops which are included in the local sensors and actuators intelligence, giving in this way a constant bidirectional communication between devices and computer.

Another standard, first proposed in Europe in the automobile industry, is the controller area network (CAN) which is based on a serial communications protocol oriented to real-time applications. The transmission media, the media access control, the error control, and the multiple reception properties, are perfectly adaptable to distributed control systems.[2]

These efforts have been triggered not only by the technological requirements of control processes, but also by emission regulations for manufacturing industries required by the Environmental Protection Agency and by ISO quality requirements standards in Europe (ISO 9000 standard series). This work presents a prototype application for teaching purposes of a distributed digital control using the controller area network (CAN) protocol for digital communications. This technology was applied to a laboratory scale level plant with a conical tank, where students have to solve the problem of controlling the liquid tank level.

CAN protocol

CAN is a serial communications protocol that allows efficient management of real-time distributed systems with good performance. The protocol satisfies requirements for high-speed networks and allows communication with sensors and actuators using a single communication bus, thus reducing the wiring costs of a classical multipoint control system. The CAN protocol was first designed for controlling different car systems by using a 1 Mbps communication bus.

The protocol is simple and does not require a physical address for the control system devices. The message interchange is managed through a priority identifier, giving bus access to those with higher priority. The multi-access message feature of the controllers ensures synchronism for the distributed process in the network. When a message is transmitted, no one node is physically addressed to receive it, but instead the message is characterised by the identifier, which classifies it into one out of four possibilities: data frame, remote frame, error frame, and overload frame. The nodes, which are in a 'bus hearing state', filter the identifier to determine whether or not they have to receive the message for processing.

OSI model

Specifications of the CAN protocol, which are found in the standard ISO 11898, consider the two lower layers of the reference OSI model:

- The Data Link Layer (DLL) which is divided into two sublayers:
  - Sublayer Logical Link Control (LLC)
  - Sublayer Medium Access Control (MAC)
- The physical layer.

The data link layer manages the data at bit and byte levels, providing a reliable information transfer between a node and the network. It builds the message to be sent to the physical layer by framing it and decodes the messages delivered by the physical layer. In the devices that control the CAN protocol, the data link layer is usually hardware implemented. Because of the complex functionality of the DLL it is divided into two sublayers. The first one, called LLC, receives and transmits messages from and to the upper layers (Fig. 1). Among the functions accomplished by it are to allow data transfer, to request remote data, to filter the messages and to notify overloads. It is also used for recovering the management system.

The MAC sublayer codes and organises in series the messages to transmit and decodes the received messages as well. Among its functionality are to control and code the message carrier structure, to arbitrate the bus access, to detect errors and to confine failures.

Operation background

When a message is transmitted, none of the nodes is explicitly addressed as a receiver. Nevertheless, the message content has an identifier which is unique and gives its own characterisation. In this way the reminder nodes that are always listening to the bus, when they receive the message, filter the identifier to accept or

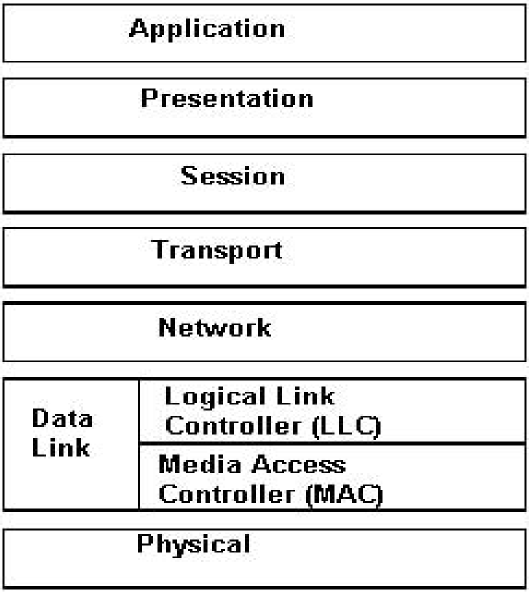

Fig. 1 *OSI communication reference model.*

reject it with a combination of software and hardware. Basically there are two communication forms:

- One node sends information to which the other listens;
- A node A sends an information requirement to a remote node B.

By means of the RTR (remote transmission request) bit located in the frame it is decided which mode of communication is used.

Structure and data management

The term frame refers to the serial structure carrying the message information from one transmission node to another one or several receiving nodes. The standard format of CAN protocol (Version 2.0A) handle messages with an 11-bit identifier. The extended format (Version 2.0B) uses identifiers from 11 to 29 bits.

The standard format structure of the CAN protocol is shown in Fig. 2, and it consists of the following fields:

- A starting field SOF (start of frame) indicating the beginning of the message.
- An arbitration field formed by the message identifier and the RTR (remote transmission requirement). The RTR bit is used to discriminate whether it is a data frame or a remote request.
- A control field of six bits; two dominant bits (r0 and r1) used for future purposes, and four bits DLC (data length code) indicating the number of bytes of data field.

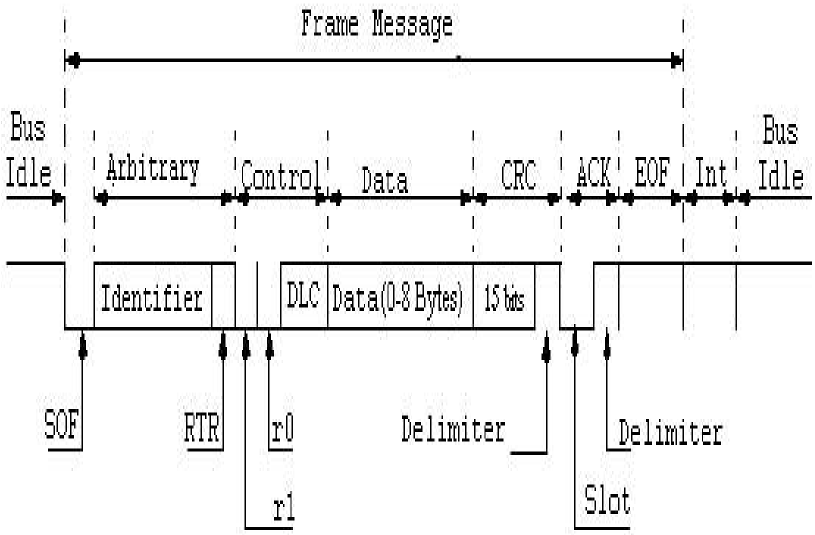

Fig. 2 *Frame structure for standard format.*

- A data field containing zero to eight bytes.
- An error check field CRC containing 15 cyclic bits to check redundancy plus a delimiter recessive bit.
- An acknowledgement field ACK formed by two bits. The first is a slot bit transmitted in a recessive way but subsequently overwritten by dominant bits transmitted by some node that has successfully received the message. The second bit is a recessive delimiter.
- An end field EOF consisting of seven recessive bits.
- An intermission field INT formed by three recessive bits. It corresponds to the minimum space between messages, after which the bus is released.

Based on the frame architecture of Fig. 2, four different types of messages can be identified; data frame, remote frame, error frame and overload frame.

CAN physical level

CAN protocol uses NRZ (non return to zero) coding with bit refill for communication in a two-wire differential bus. When five bits exist having identical logic level, the protocol controller device will insert a complementary bit at the end.

The physical layer defines the electronic levels, the signal scheme and the cable impedance amongst others. There exists several others physical layer schemes. Standard ISO 11898 defines a balanced two-wire signal scheme. The less frequently used standard ISO 11519 defines a balanced two-wire signal scheme for low speed purposes. There exist a variety of proprietary physical layer systems.

A widely used transmitter is the 82C250 by Philips, which implements the physical layer defined by standard ISO 11898.

The number of nodes existing in a network is theoretically infinite. However, handling capacities of bus interface devices constrain the theoretical situation. Depending upon the devices, a number of 32 to 64 nodes is normal, being the transmission rate dependent on total bus length. For devices satisfying standard ISO 11898 bit rates as high as 1 Mbps can be obtained in a 40 m length bus.

The physical media corresponds to two lines with common return, ending with resistors in both extremes representing the line characteristic impedance. Standard ISO 11898 indicates that nominal cable impedance has to be 120 ohms. Nevertheless, impedances between 108 and 132 ohms have proved to be suitable where the physical medium was twisted pair (shielded or unshielded). The reader is referred to reference [3] for a detailed study of the CAN protocol.

Hardware description

To implement CAN networks, devices working according to specifications of the two lower layers of the OSI reference model are used. In an upper level we have CAN controllers which handle the protocol allowing communication between the higher levels such as application and physical layers.

Controller protocol CAN 82527 (Intel)

The 82527 is an integrated device which allows the complete handling of the CAN protocol.[4] The programming is based on internal register handling and allows configuration of the device operation according to communication characteristics within the bus. Its interface abilities allow connection of it to several types of CPU or microcontrollers.

The control of messages is carried out in an internal RAM addressed by the CPU externally, as if it were a peripheral memory device. The CPU can drive the 82527 device by programming each one of their internal registers, configuring functions such as acceptation filter, transfer, interruptions by messages, etc. The fundamental unit, denoted as object message, has an extension of 15 bytes and defines all messages characteristics, either as receptor or transmitter. The 82527 device handle a total of 15 object messages, the first 14 can be transmitters or receivers and they operate on a unique global acceptation filter (global mask).

CAN transmitter PCA82C250 (Philips)

The PCA82C250 device is the interface between the CAN controller and the physical bus.[5] For high speed applications (over a 1 Mbps) the device provides differential transmission reception abilities. It is completely compatible with standard ISO 11898.

By means of an external resistor Rs connected between pin 8 (see Fig. 3) and ground, the up and down slope of the bit can be regulated. Slopes are proportional to the electrical current flowing through the resistor Rs. For lower speed or short length buses, a twisted pair can be used as physical communication media.

With the aim of programming control algorithms to be applied at the level control plant, the CAN network was implemented by using a personal computer (PC) with

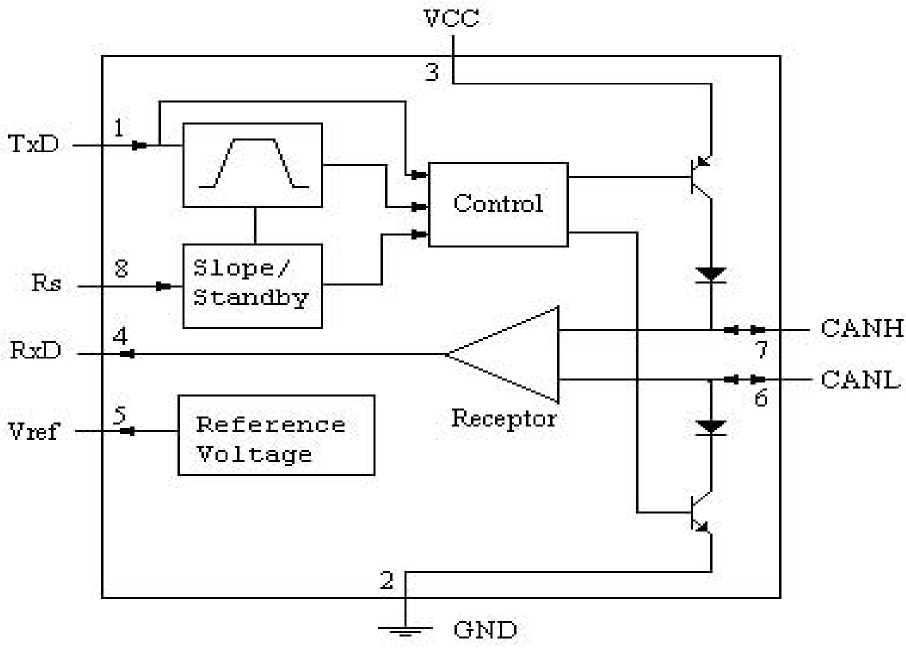

Fig. 3 *Block diagram of PCA82C250 device.*

a communication card serving as the interface CAN-PC. The card chosen was the PCCAN manufactured by Kvaser connected to the PC ISA bus.[6]

The address space is configured through jumps located on the card.[6] Interruption lines must be programmed by software. The card has four devices to control the CAN protocol; two Philips 82C200 and two Intel 82527. Both types of controller are commercially available, and the decision between them is made based on the particular application. The drivers are of two types; one for the Intel controllers and the other for Philips controllers.

Subroutines provided by the manufacturers and running under several programming environments (DOS, Win 3.1x, Win NT and Win 95) were used. These subroutines simplify the controller handling located on the card, by taking the message space as if it were an intermediate buffer.

Application to a laboratory plant

In the Automatic Control Laboratory of the Electrical Engineering Department of the University of Chile, a level and flow control plant is used in teaching automatic control subject. This plant is made of two tanks; one conical (main tank) and other cylindrical (recirculating tank) as shown in Fig. 4. Two flows enter the conical tank, one from the water network and other from the recirculation tank. The recirculating tank has two outputs; one to the conical tank controlled by an on/off valve and the

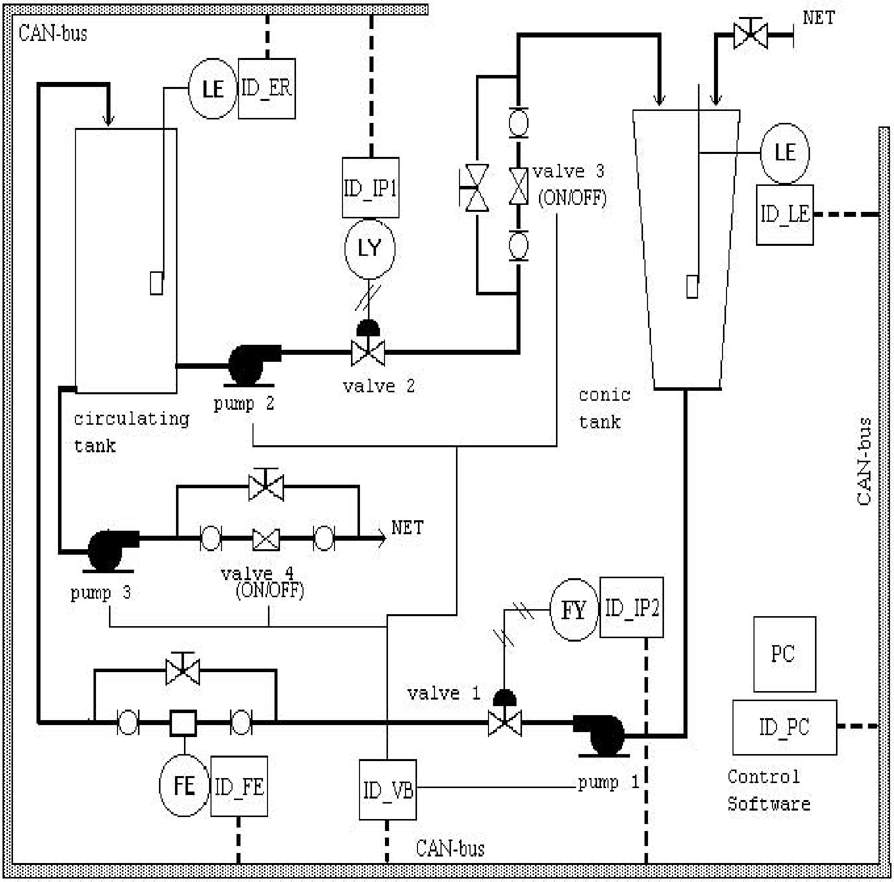

Fig. 4 *P&I flow and level laboratory plant with the CAN communication system.*

other to the drainage. Two control loops are identified in this plant; the flow-valve 1 loop and the level-valve 2 loop. The control objective is to maintain the level of the main (conical) tank at a value given by an external reference, even in the presence of external perturbations acting on the plant.

Once the sensors and actuators have been identified, the design of the digital control system can be started, based on a unique CAN serial communication bus. A typical scheme is that shown in Fig. 4 including the CAN bus and the PC with the PCCAN card. The physical medium is a twisted pair. Each plant instrument is associated with an identifier, depending on its hierarchical importance relative to another instrument. This capacity allows different handling of variables, assigning a higher priority to some critical variables over others that can be considered less important in the process. In this particular case, the signal coming from the level sensor will have a higher priority than the signal coming from the recirculating tank, to avoid possible water spillover. The transmission-reception CAN node is formed by five functional blocks: the transceiver (Philips PCA 82C250), CAN controller (Intel CAN82527), microcontroller (Intel 8051/8751), conversing stage and synchronism clock.

The CAN controller builds and interprets messages from and to the bus. The microcontroller manages signals coming from and going to sensors and actuators. The PC user interface is software developed in Windows 95 that graphically emulates the processes taking place. The network communication is performed through the PCCAN card using the libraries provided by the manufacturer. The controller parameters are introduced through a pull out menu allowing the user to select the manual or automatic control modes. A window displaying the evolution of the main process variables was also included.

Once the PID control loops were tuned using the Ziegler-Nichols method and the plant model was suitably determined, the plant was controlled on-line.[7] A typical system response can be found in Figs 5 and 7 showing the evolution of the controlled variables (level and flow) when several changes on the level and flow references were applied. In Fig. 5 the flow at the bottom of the tank has been kept constant, whereas in Fig. 7 the level is kept constant. From these figures one can observe that a satisfactory control response is obtained, since no big overshoot and slow settling time are observed.

Figs 6 and 8 show the control variables for the cases shown in Figs 5 and 7 respectively. All of them remain inside the control range with no big effort on the valves. Though a quite simple control strategy was implemented (PID control), more advanced control strategies can be easily implemented using the same general scheme developed. Thus every semester students will have the chance to study and learn about different kinds of control strategies. During the first and second semes-

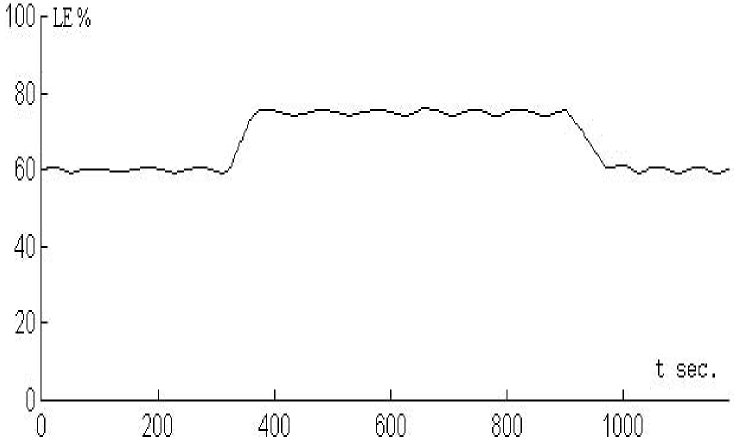

Fig. 5 *Tank level under set point changes.*

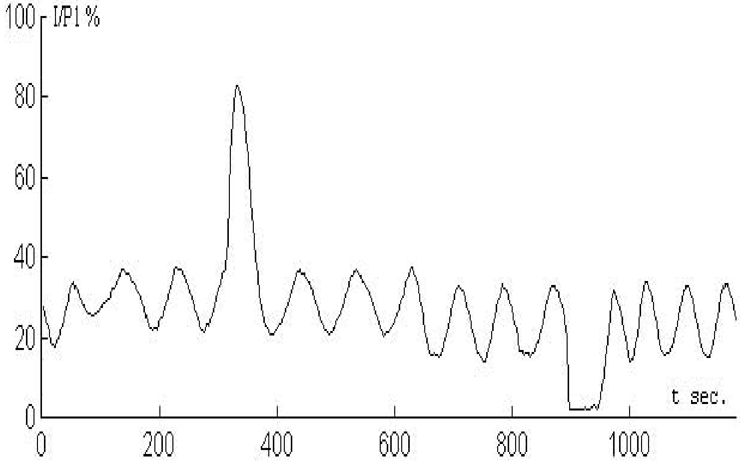

Fig. 6 *Control action for the level loop.*

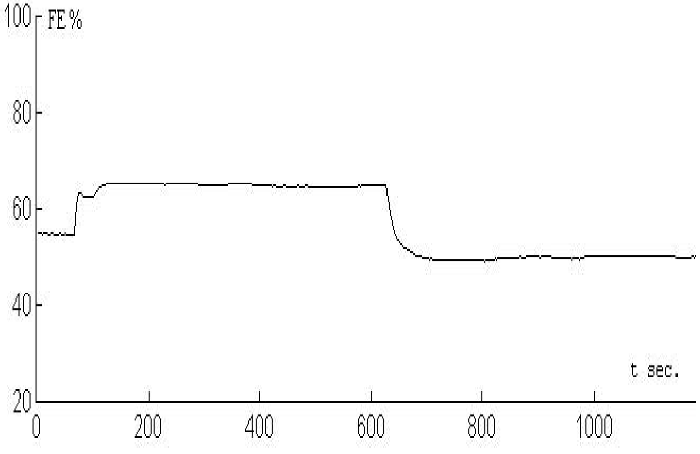

Fig. 7 *Water flow under set point changes.*

ter of 1999 the control system was included in the course EL42D Systems Control as a complementary activity. About 35 students each semester had the chance to operate the laboratory plant, operating the whole system and registering information about plant operation. After finishing the experimental work students showed a high motivation for obtaining a deeper knowledge about modern technologies in communication systems applied to the case of process control and its on-line control application.

The overall material cost of the CAN system was about US$1,500 when it was first implemented (autumn 1999).

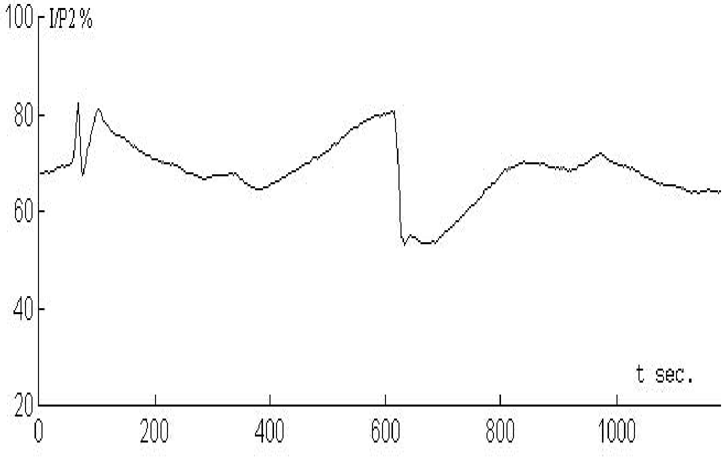

Fig. 8 *Control action for the flow loop.*

Conclusion

Advances in digital communications are being reflected in the process control area, by the development of new data transmission technologies. Even though the application of communication buses in process control is being done over a decade, the design of new integrated control communication systems or microcontrollers that include protocols based on the OSI reference model have allowed these technologies to be applied to distributed control systems on-line, regardless of size. A communication technology based on the CAN protocol has been implemented to control the level of a conical tank for educational purposes, to train electrical engineering students in modern control techniques. After taking the course, undergraduate students showed a higher motivation than in the past to learn these control techniques and to gain experience in how digital communications are applied in a plant for these purposes.

Acknowledgements

The results reported in this paper have been supported by CONCITY under grant FONDECYT 1970351 and by the Electrical Engineering Department of the University of Chile.

References

1 Fieldbus Foundation, available at http://www.fieldbus.org.
2 R. Bosh, *CAN Specification*, ver. 2.0, (GmbH, Stuttgart, 1991), pp. 33–68.
3 P. Kremer, ‘Study of the CAN communication protocol for applications in distributed digital control processes’ (in Spanish), Electrical Engineering Thesis, Department of Electrical Engineering, University of Chile, October 1998.

4 *82527 Serial Communications Controller Architectural Overview* (Intel, 1996), pp. 3–37, available at http://www.intel.com/design.
5 *PCA82C250 CAN Controller Interface*, (Philips Semiconductors, 1997), available at http://www.philips.com.
6 *PCCAN The CAN Card For PC Computers*, ver. 1.0B (Kvaser, Sweden, 1997), available at http://www.kvaser.se.
7 K. Ogata, *Modern Engineering Control*, second edition (Prentice Hall, Englewood Cliffs, 1993), pp. 634–646.