2011 IEEE International Conference on
Automation Science and Engineering
Trieste, Italy - August 24-27, 2011

The Time Delay Control of CAN Messages for Real-Time Communication

Dan PUIU and Florin MOLDOVEANU, *Members, IEEE*

Abstract—**The latest advances in the field of microelectronics and modern control systems demand for more flexible control architectures. As a result of that, the centralized control architectures are replaced by flexible distributed control systems based on intelligent drives and powerful communication networks. CAN network was designed for car networking in order to reduce the complexity of the related wiring harnesses. To achieve the real-time communication constraints the traditional CAN application technique must be updated. This paper presents a new approach of using the dynamically priority allocation technique for a CAN network. Our solution is to increase gradually the message priority until it becomes the message with the highest priority from the network and it is transmitted. This method of message priority allocation allows us to determine its maximum transmission time.**

I. Introduction

THE latest advances in the field of microelectronics and modern control systems demand for more flexible control architectures. As a result of that, the centralized control architectures are replaced by flexible distributed control systems based on intelligent drives and powerful communication networks such as: CAN, PROFIBUS and Industrial ETHERNET. CAN was designed for car networking to reduce the complexity of the related wiring harnesses. In order to achieve the real-time communication constraints the traditional CAN application technique must be updated [1,6].

A distributed solution does not imply only the physical spreading of the components, but there is necessarily an array of instruments and specific mechanisms, that facilitate the efficient exploitation of the system distributed resources. From this reason, the implementation of distributed control systems concept determines the apparition of new specific implementation and operation problems that in centralized control systems do not exist [4, 5].

When is designed a distributed control system it must be considered and compensated the drawbacks introduced by the industrial communication networks such as the eventual loss of information packages and the messages transmission times [2].

Furthermore, it is important that all the intelligent units, such as microcontrollers, digital signal processors and programmable logic controllers, to work in a synchronous way. This imply, to be implemented one or more synchronization mechanisms that permits the cooperation among the elements in order to solve their common goal [3].

From these reasons several studies had been made to ensure the real-time capabilities for the industrial communication networks.

Tovar and Vasques in the paper [11] modify the Medium Access Control (MAC) layer for a PROFIBUS network. They implement a low priority messages counter and this approach permits the calculation of the messages transmission times.

In order to solve the real-time communication aspects for a CAN network Pinho and Vasques in the paper [10] design a new middleware that should be included between the CAN hardware module and the application program. The software has three different protocols that solve the problems generated by inconsistent, duplicated and undelivered messages.

Lee and Jeong have designed in the paper [7] a dynamically priority allocation algorithm for a CAN network. They consider that the Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol determines some messages to be transmitted with significant delays because they have low priority and the communication network is loaded with high priority ones. They solved this problem by implementing the deadline base priority allocation algorithm, which allocates for each message a transmission time. If the network is load with high priority messages and the message transmission time becomes bigger than allocated transmission time than the priority of the message is increased. A similar method is presented by Wu, Li and Qin in the paper [12].

Another dynamic priority algorithm for CAN network, which is based on the fuzzy logic and neural network technique was developed by Lin et. all in the paper [8].

This paper presents a new approach of using the dynamically priority allocation technique for a CAN network. Our solution is to increase gradually the message priority (decrease the message identifier field) until it becomes the message with the highest priority from the network and it is transmitted. This method of message priority allocation allows us to determine its maximum

Manuscript received March 10, 2011. This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/6/1.5/S/6.

D. Puiu is with the *Transilvania* University of Brasov, Automation Department. (corresponding author; e-mail: puiudan@unitbv.ro).

F. Moldoveanu is with the *Transilvania* University of Brasov, Automation Department. (e-mail: moldof@unitbv.ro).

978-1-4577-1732-1/11/$26.00 ©2011 IEEE

transmission time.

The paper is organized into four sections including this introduction. Section 2 presents the dynamic scheduling algorithm, while section 3contains the mathematical model of the messages maximum transmission times and the experimental model. Finally, conclusions and further work are presented in section 4.

A. *Real-Time Control Systems with Delays*

Generally, a distributed control system has a hierarchic architecture with four layers, like the one presented in Figure 1. It is know that the time constraints are more stringent as we go down in the automation hierarchy [12].

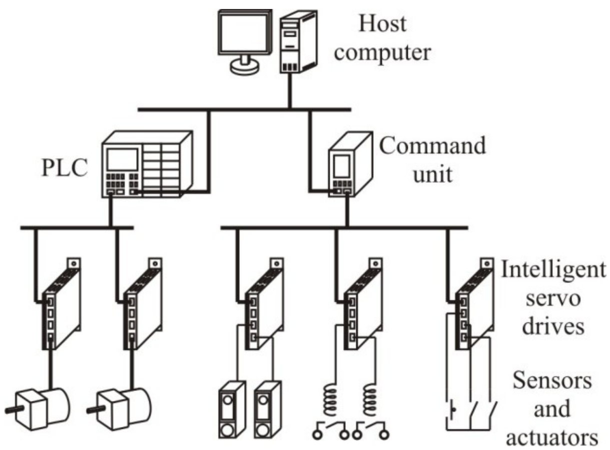

Fig. 1. Distributed control architectures based on four layers.

For all types of industrial communication networks the transmission time of a message has three main components [9]:

- the processing time, this represents the time used by the sender node to prepare the message and to get access to the communication environment;
- the network time, this represents the time required by a package of information to go through the communication network;
- the synchronization time, this represents the time that the message waits in the input stack of the receiver.

For a CAN network the synchronization time can be reduced if in the receiver node is implemented an efficient interrupt service routine. Furthermore this time can be calculated easily because it depends by the clock frequency and by the interrupt service routine of the receiver.

Because of the CSMA/CD mechanism, the network time is always the same for all messages, without any influence from the physical distance between the sender and the receiver. This time depends by the format of the message (CAN 2.0A or CAN 2.0B) and by the baud rate of the CAN network.

The processing time of a CAN message depends: by the network load and by the priority of the message. In the worst case scenario it is possible that a low priority message not to be transmitted or to be transmitted with significant delays because the network is full of high priority messages.

For a real-time control system the guarantee of a medium transmission time is unsatisfying. The industrial communication network should have a deterministic behavior that permits the exact evaluation of the transmission time. In order to obtain these rigorous constraints should be used one of the following techniques [6]:

- by using of deterministic algorithms to access the communication medium;
- by predicting the network load;
- by using an explicit mechanisms of planning the messages.

B. *The CAN Communication Protocol*

The medium access control (MAC) mechanism adopted in CAN [13, 14] is basically a carrier-sense multiple access (CSMA) technique, enhanced with a special arbitration phase to ensure that the collisions occurring on the bus when two or more nodes start transmitting at the same time are solved in a deterministic way. As for the conventional CSMA/CD networks, CAN is based on a shared bus topology. The most distinctive feature which distinguishes the physical layer of CAN from other kinds of local area networks is the electrical interface to the bus, which is similar to the open collector wiring and performs a wired OR function among the connected nodes. In particular, the bus can assume two complementary values, indicated as dominant and recessive; the level of the bus is dominant if at least one node is writing a dominant value, otherwise it is recessive.

The MAC sublayer in each station can start transmitting a frame as soon as the bus is idle, that is to say, when no message is being transmitted. In the arbitration phase, which precedes the actual transmission of data, each transmitting station compares the bit being written with the level on the bus. If the bit written is recessive, but the level read is dominant, the node understands it has lost the contention (because a higher priority message is being sent) and switches to the receiving mode. All the nodes which lose the contention have to retry their transmission after the higher priority message has been completely sent by the winning station. In this way, when a collision takes place, the contention is solved by stopping all the stations which are transmitting the lower priority messages, hence, no information nor time are lost. Since the arbitration is carried out on the identifier field, the lower this identifier is, the higher the priority of the message. The arbitration technique of CAN is able to determine with a single operation which of a given set of nodes is transmitting the lower valued identifier. This When it receives a frame, the MAC sublayer does not perform any type of check on the identifier field. The actual identification of the incoming messages is carried out by the frame acceptance filtering function in the upper layer, which decides if the object enclosed in the received message has to be forwarded to the application programs which are executing on the node.

The general transmission procedure for a CAN message from a Fujitsu microcontroller is presented in Figure 2. It can be observed that at the beginning is defined the validity period of the message (timeout), than is set the message identifier and its' contend and at the end the message is placed in the output stack. Finally, the processor waits until the CAN message is transmitted or the network is halted or the CAN message reached the maximum transmission time (timeout).

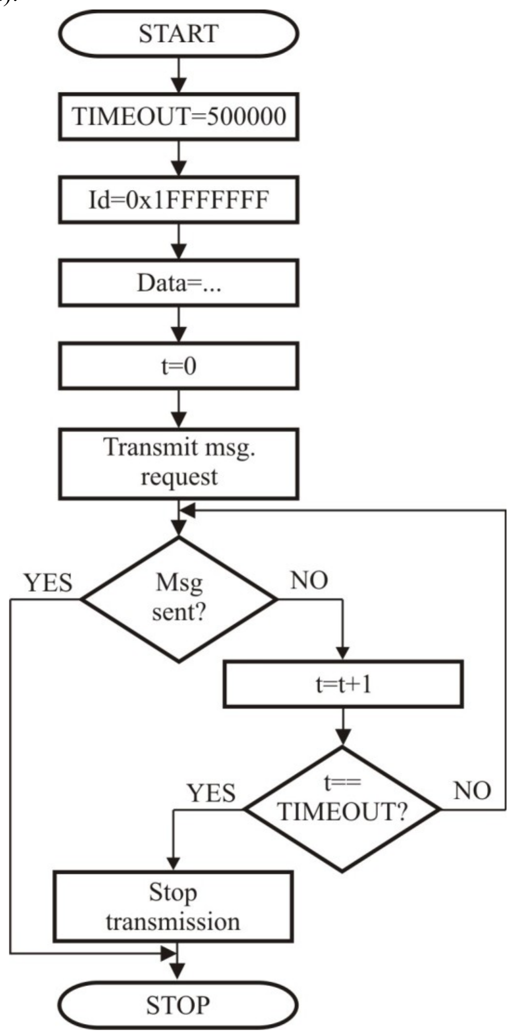

Fig. 2. The transmission procedure of a CAN message.

II. DYNAMIC PRIORITY ALLOCATION ALGORITHM

The information from the message identifier of a CAN message is used by the Logical Link Control (LLC) structure for determining the node address and by MAC for determining the message priority. As a result of that the performances of a CAN network are related to the message priority allocation algorithm.

Usually the message identifiers are allocated when the design of the control system with CAN network begins, after it had been determined the messages that should be transmitted on the network. This method works, with good results, in case of small real-time distributed control systems. In case of more complex systems it happens that a small priority message is not transmitted or is transmitted with a delay bigger than the delay estimated in the developing stage if the CAN network is loaded with high priority messages. The solution, for this problem, is to allocate the message identifies, respective the message priority, during control system operation. In this case one node of the CAN network can increase the priority of a message if the message is not transmitted.

A. *CAN Message Identifier Configuration*

For presenting the dynamic priority allocation algorithm it will be used an extended CAN network, which has 29 bits for the message identifier field.

The model proposed in this paper represents a combination between the static ways of message priority allocation and the dynamical ones. As is presented in Figure 3, the user has a set of high priority messages, with static identifier, which can be used in case of critical operating errors. For the general purpose transmission the user should use dynamically priority allocation messages. The number of high priority messages must be calculated with the equation:

$$N_p = 2^k \tag{1}$$

where, k represents an integer number not bigger than 11. When $k = 6$ there are 64 messages for critical operating errors.

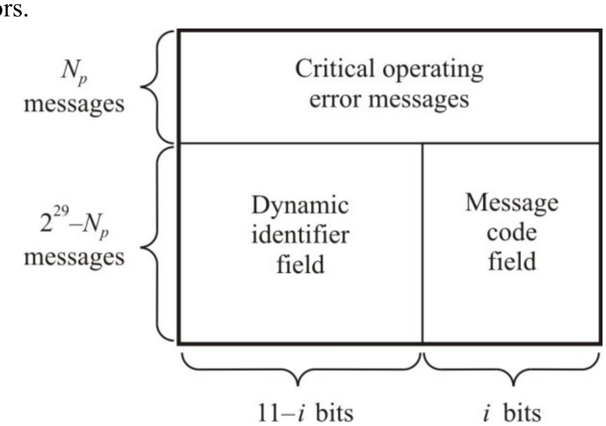

Fig. 3. The CAN messages allocation table for the dynamic priority allocation algorithm.

When a node needs to send a general purpose message it will allocate the biggest identifier ($2^{29}-1$ for extended CAN), which represents the smallest priority. Each time the node tries to send the message and it fails, because there are other messages with bigger priority, it will decrease the message priority. After several consecutive failures the message has the biggest priority from the network and it is transmitted.

According to this approach, if two nods need to transmit an general purpose message in the same time, both messages will receive the same message identifier and will result a collision on the CAN network. To avoid this unpleasant situation each message that has to be transmitted on the network will receive a code, as is presented in Figure 3. As a result of that the dynamic priority allocation algorithm will decrement only the rest of the bits from the message identifier.

In conclusion the CAN message identifier contains two fields: the dynamic identifier field and the message code field. The user allocates the message code field in static conditions, after it decides which messages has to be transmitted on the CAN network. The user must declare no more than:

$$N_{mc} = 2^i \tag{2}$$

message code fields, where i is an integer number smaller than 12. As a result of that, the user can use only 4096 messages.

In Figure 4 is presented the transmission procedure of a CAN message which respects the above priority allocation algorithm when $i = 12$. The main differences from the general transmission procedure from Figure 2 are:

– the validity period of a message (timeout) is much smaller;

– at the beginning the processor chooses, for the message, the smallest priority: 0x1FFF**0005** (the bold number represents the message code field);

– if the message reaches the maximum transmission period, its' dynamic identifier field is decremented and the transmission is requested;

– when the message is transmitted, the transmission loop is stopped (msgSent=TRUE).

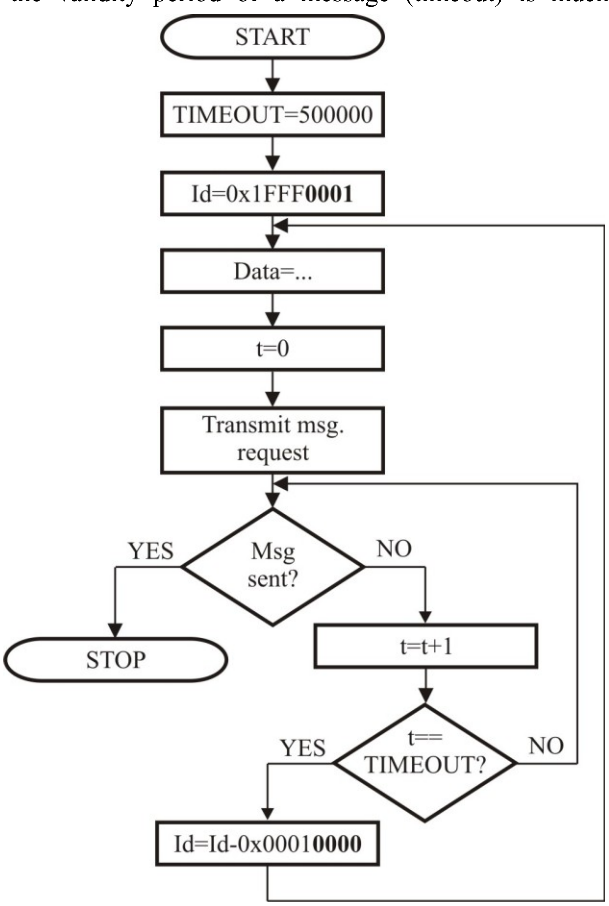

Fig. 4. The CAN message transmission procedure using the dynamic priority allocation algorithm.

III. THE TRANSMISSION TIME OF A CAN MESSAGE

This approach allows the designer of a real-time distributed control system to estimate from the beginning the maximum transmission time of a CAN message.

The maximum transmission time depends on: the validity time of the CAN message (allocated by designer in the transmission procedure) and the maximum possible number of attempts to transmit the message.

In the worst case scenario, the maximum number of attempts to transmit the message is direct proportional with the maximum number of decrements of the dynamic identifier field plus the total number of messages that have the message code field smaller than the current message code field.

According to this, it has to be determined the total number of identifiers allocated to a message code field. Because on the CAN message table are messages with fixed identifier the total number of messages with dynamic identifier is:

$$N_{MDI} = 2^{29} - N_p \,. \tag{3}$$

As a result of that the total number of messages with dynamic identifier allocated to a message code field is:

$$N_{MC} = \frac{N_{MDI}}{2^i} \,. \tag{4}$$

In worst case scenario, the transmission time of a message can be calculated, based on (1), (2), (3) and (4), with the equation:

$$t = \left(N_{MC} + N_x\right) \cdot T \,, \tag{5}$$

where t is the worst case scenario transmission time, N_x is the number of message code fields smaller than the current message code field and T is validity period of a message.

The number which is defined in TIMEOUT variable do not represents exactly the validity period of a message. The TIMEOUT determines how many times the processor has to check if the message is transmitted. Using TIMEOUT and the system clock, it can be determined the approximate value for T. For best result the checking procedure can be implemented using a timer. In this case every time the timer interrupt occurs, the interrupt service routine of the processor has to check if the message is transmitted or not. As a result of that the validity period of the CAN message is direct proportional with the period of timer interrupts and with the total number of timer interrupts.

For a simple example it had been considered an extended CAN network with the baud rate 125 Kbit/s. Because the message identifier field has 29 bits resulted that the length of a message is 125 bits. The transmission time of a message is 1mS if the network is empty.

In order to simulate the dynamic priority allocation algorithm, it had been implemented in Matlab/Simulink an arbitration structure identical with the one from the CAN network. The structure contains several threads that implement the arbitration structure and the message generation mechanisms from the nodes.

The thread that simulates the arbitration structure has the ability to generate graphs (like the one from Fig. 5) that represents the evolution identifiers from a network. In Figure 5 had been considered a CAN network with two nodes. The first node (“o” message identifiers) has the role of network load generator because it transmits the message 0x0007**0001** in a continuous mode. In the second node (“♦” message identifiers) is implemented the dynamical priority allocation algorithm and it has to fight to receive the network. The node is initially configured to transmit the message 0x000C**0200** and it succeeds only when the message identifier is 0x0006**0200**.

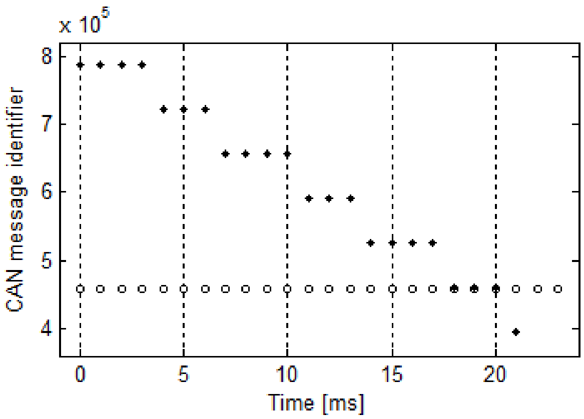

Fig. 5. The simulation of the CAN message identifier allocation algorithm.

Evaluating the equation (5), it can be found out that the all the messages from the network have the same maximum transmission time and as a result of that the same priority. In order to vary the transmission times the user can do one of the following actions:

- modify the validity period of the message;
- modify the decrement value;
- modify the initial value of the dynamic identifier field.

A. The Experimental Model

For experimental testing of the dynamically priority allocation algorithm was used a distributed control system as the one presented in figure 6. The system is composed of three developing boards with microcontrollers MB90F357AS from Fujitsu and a computer. All these logical units are connected using a CAN network. The microcontroller boards have their own CAN interface, while the computer is connected to the network using a CANCardX board from Vector.

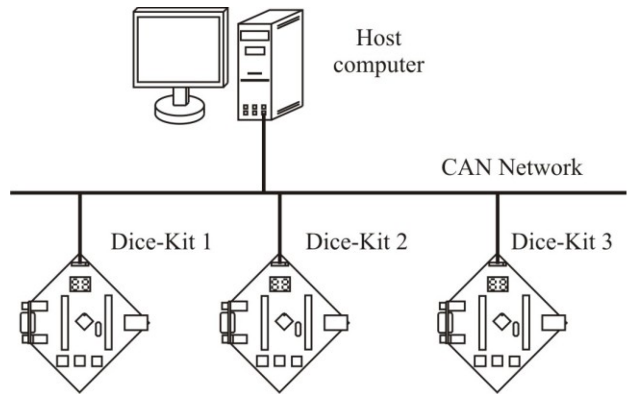

Fig. 6. The distributed control system used for validating the models.

In the first experiment, it had been considered that the DiceKit 1 board transmits in a continuously the message with a high priority identifier. This board has the role to create a load in the CAN network. The second board was configured transmit, using the classical method, a message with a bigger identifier (smaller priority). Using the facilities of the CANCardX board, the CAN network was monitored and it resulted that only DiceKit 1 managed to transmit the messages.

In the second experiment, the dynamic priority allocation algorithm was implemented In the DiceKit3 board. The node succeeds to transmit the message, but after it modifies it identifier.

IV. Conclusions

In this paper is presented the time delay control problems from a distributed control system. The obtained solutions will be implemented in a distributed system for multi-axis motion control applied to industrial robots.

In order to solve this problem for the distributed control system with CAN network, it have been developed an explicit mechanism for message identifier allocation. This approach allows the designer to estimate, from the developing stage, the maximum transmission time of a message.

The dynamic allocation algorithm has the following advantages:

- it permits the estimation of the message maximum transmission time;
- the receiver can determine exactly the information from the CAN message (because of the message code field);
- it is based on the “classical principles” of the CAN message identifier allocation.

In the “classical solution” the designer, at the beginning, determines the messages that should be transmitted using the CAN network and creates the CAN message identifier table. In this new approach, the designer also has to do that step and the CAN message identifier table represents the message code field.

The new mechanism of message identifier allocation presents several disadvantages:

- it needs supplementary processing power from the node processor;
- the debugging and testing of the system is more complicated;
- the distributed system can be extended only if the new node has implemented the dynamic allocation algorithm.

Considering the advantages and the disadvantages of the time delay control method, it can be implemented in distributed multi-axis control systems. The distributed structure of an arm robot contains a set of local intelligent servo drives that are connected using a CAN network to a motion coordinator. The developed algorithm can be implemented in such a system because:

- the motion coordinator needs to know the transmission time;
- the intelligent drives and the motion coordinator have a lot of processing power;
- the structure does not need extensions.

REFERENCES

[1] Alldredge, G. W. „PID and Model Predictive Control in a Networked Environment”, Master’s Thesis, Case Western Reserve University, Ohio, USA, 2007.

[2] Antsaklis, P., Baillieul, J. „Special Issue on Technology of Networked Control Systems”, Proc. of the IEEE, Vol. 95, No. 1, Jan. 2007, pp. 5÷8.

[3] He, F., Zhao, S. „Research on Synchronous Control of Nodes in Distributed Network System”, Proc. of the IEEE Inter. Conf. on Automation and Logistics, Qingdao, China, Sept. 2008, pp. 3029÷3034.

[4] He, F., Tong, W., Wang, Q. „ Research on Synchronization Control Strategy of Multi-motor System Based on Profibus Network”, Proc. of the IEEE Inter. Conf. on Automation and Logistics, Jinan, China, Aug. 2007, pp. 3029÷3034.

[5] Gil-Castineira, F., Gonzalez-Castano, F. J., Franck, L. „Extending Vehicular CAN Fieldbuses with Delay-Tolerant Networks”, IEEE Trans. on Industrial Electronics, Vol. 55, No. 9, Sept. 2008, pp. 3307÷3314.

[6] Gupta, R. A., Chow, M. Y. „Networked Control Systems: Overview and Research Trends”, IEEE Trans. On Industrial Electronics, Vol. 57, No. 7, July 2010, pp. 2527÷2535.

[7] Lee, H.H., Jeong, U. H. „Design of Distributed Scheduler on CAN for Real-Time Networking”, Proc. of 5th Russian Korean Int. Symp. Science and Technology – Korus 2001, Tomsk, Russia, May 2001, pp. 22÷25.

[8] Lin, C.K., Yen, H.W., Chen, M.S., Hwang, C.P., Thanh N.M. “A Neural Network Approatch for Controller Area Network Message Scheduling Control”, Proc. of the Int. Multiconf. of Engineers and Computer – IMECS 2006, Kowloon, China, 20÷22 June, 2006, pp. 36÷41.

[9] Nilsson, J. „Real-Time Control Systems with Delays”, Ph.D. Thesis, Lund Institute of Technology, Lund, Sweden, 1998.

[10] Pinho, L. M., Vasques, F. „Reliable Real-Time Communication in CAN Networks”, IEEE Trans. On Computers, Vol. 52, No. 12, April 2003, pp. 1594÷1607.

[11] Tovar, E., Vasques, F. „Real-Time Fieldbus Communications Using Profibus Networks”, IEEE Trans. on Industrial Electronics, Vol. 46, No. 6, Dec. 1999, pp. 1241÷1251.

[12] Wu, Q., Li, Y., Qin, Y. „A Scheduling Method Based on Deadline for CAN-based Networked Control Systems”, Proc. of the IEEE Inter. Conf. Mechatronics and Automation, Luoyang, Henan, June 2006, pp. 345÷350.

[13] ***, „Road Vehicles: Interchange of Digital Information. Controller Area Network for High-Speed Communication”, ISO 11898, Nov. 1993.

[14] ***, „Road Vehicles: Interchange of Digital Information. Controller area Network for High-Speed Communication, Draft Amendment”, ISO 11898:1993/DAM 1, Feb. 1994.