Bulletin of the *Transilvania* University of Braşov • Vol. 3 (52) - 2010
Series I: Engineering Sciences

# THE TIME DELAY CONTROL OF A CAN NETWORK WITH MESSAGE RECOGNITION

**D. PUIU**[1] **F. MOLDOVEANU**[1]

***Abstract:*** *Due to the quick evolution of manufacturing processes, the demand for more flexible automation systems is on the rise. To answer to these requirements, distributed control architecture based on intelligent drives and CAN networks tends more and more to replace the traditional solutions. CAN was designed and applied in car networking in order to reduce the complexity of the related wiring harnesses. The traditional CAN application technique must be changed in order to achieve the real-time communication constraints of a distributed control system. In this paper is designed a special purpose scheduler for CAN, which ensures a message transmission time smaller than a value estimated from the design stage. Furthermore, the designed architecture has a message recognition mechanism.*

***Key words:*** *distributed control system, time delay control, real-time control systems.*

## 1. Introduction

Due to the quick evolution of manufacturing processes, and especially because of the products and technologies life cycles reduction and because of the fixed automation prohibitive prices, the demand for more flexible automation systems is on rise [7]. To answer these requirements, the "traditional" solutions are replaced by distributed architectures based on intelligent drives. Unlike the "classical" solution, where the control structure tasks are in a central controller, the distributed control architecture is based on industrial communication networks technique (with their specific protocols) and on digital signal processing technology. As a result of that the control tasks can be decentralized and can be solved locally by the intelligent drives [5].

Generally, a distributed control system has a hierarchic architecture with several layers. It is know that the time constraints are more stringent as we go down in the automation hierarchy [2]. Several studies had been made to ensure real-time communications capacities for the fieldbusses.

Tovar and Vasques, in the paper [2], modify the medium-access control (MAC) level of a PROFIBUS network, by implementing of a low-priority message counter and achieve a known transition time for the low-priority messages.

The real-time communication aspects for CAN networks were studied by Pinho and

---

[1] Dept. of Automation, *Transilvania* University of Braşov.

Vasques in the paper [8] where they designed a middleware, with three different protocols, that solves the problems determined by inconsistent, duplicated or undelivered messages.

In the paper [6], Lee and Jeong design a distributed scheduler that ensure real-time capabilities for a CAN network. They consider that the CSMA/CD priority protocol determines some messages to have significant delays because they have small priority, while the network is occupied with high priority messages. This problem is solved by the deadline based priority allocation algorithm which increases the priority of a message when it reaches the maximum delay time.

In this paper is presented a special purpose scheduler for a CAN network. The algorithm increases the priority of a message gradually each time the transmitter node fails to send the message because the CAN network has to send a higher priority message. The paper offers a mathematical model, which calculates the maxim transmission time of a message. The message priority modification algorithm does not allow message recognition and as a result of that the algorithm is modified, in the last part of the paper, in order to remove this drawback.

## 2. Real-time Systems with Delays

The design of a distributed control system, like the one presented in Figure 1, implies the compensation of the problems introduced by the industrial communication network such as the possible loss of packages and the delays determined by transmission times [4].

The transmission time can be divided in three categories [9]:

- the processing time, which represents the time needed by a nod of the network to prepare and send a package of information;
- the network time, which is the time required by an information package to go through the network;
- the synchronization time represents the time that the information package waits in the input stack of the receiver node.

In real-time control systems, the guarantee of a medium transmission time is unsatisfying. The network must have a deterministic behaviour, which allows the evaluation, from the design stage, of message transmission time [3]. The industrial communication networks used in distributed control applications must be optimized in order to guarantee for a message a limited time interval in which it will be transmitted. This requirement can be

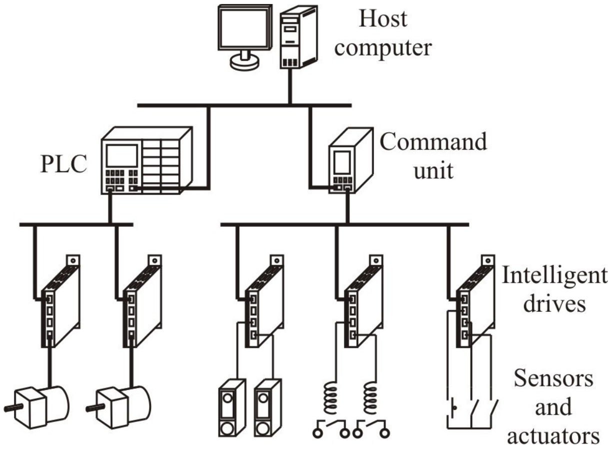

Fig. 1. *Distributed control architectures based on four layers*

archived by combining several techniques [1], [9]:

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

The deterministic access to the communication network can be achieved using communication protocols which eliminates the hazard generated when a collision between two messages appears and which guarantee, for a node of the network, a default period to access the communication network. This method can be found at the communication protocols that periodically transmit the network access right from a node to another. These mechanisms are less efficient because they cannot transfer big volumes of data, but allows the designer to mathematically evaluate the maxim transmission time of a message.

Generally, the distributed control systems have a stable configuration of hardware resources involved. As a result of that the designer can evaluate the load of the communication network and the configuration of the messages flux. The majority of the tasks of a node are periodical in time (for ex. data acquisition, data processing and commands transmission). This implies a periodicity of the transferred messages. The designer can determine from the beginning the list of the periodic messages and their periodicity. For example the period of a message that carries the value measured by a sensor depends by the sampling period of the sensor.

Unfortunately not all the data transfers can be solved by periodic messages because in a distributed control system appears non periodic events generated by errors in the system. In this case the objective of the message planning algorithm is to find an optimum solution, which respect all the transmission times.

## 3. The Dynamic Priority Allocation Algorithm

The information from the identifier field of a CAN message is used by the logical link control (LLC), from the CAN data link layer, for address recognition and by the MAC to determine the priority of the message. As a result of that, the performances of the CAN network depends on the priority decision function.

Generally the identification fields of the CAN messages are allocated in off line conditions. The fixed identifier field allocation method is working with good results in case of small size networks, but in large distributed control systems, the messages with lower priority are not transmitted until the high priority messages are finishes. In case of affluence of high priority messages the allocated time by the designer for a small priority message is passed. The solution is to gradually increase the priority of the message until it becomes a high priority one.

### 3.1. CAN Identifier Field Configuration

The difference between CAN 2.0A and CAN 2.0B is basically located in the format of the message header, especially the identifier. The CAN 2.0A defines CAN systems with a standard 11 bits identifier while CAN 2.0B is for extended 29 bits identifier. The time delay model presented in this paper is for standard CAN and the message recognition algorithm is elaborate for extended CAN.

The new concept of identifier allocation represents a combination between fixed identifier field allocation and the dynamic identifier field allocation. As is presented in Figure 2, the user, when designs the message allocation table, can allocate messages, with high priority for the system error messages. For a good utilization of the CAN identifier allocation table, the number of high priority messages of the network must be calculated with the following equation:

$$N_f = 2^k, \tag{1}$$

where $k$ is an integer number from the interval [0; 11]. The recommended value for $k$ is 3 and it implies eight priority messages.

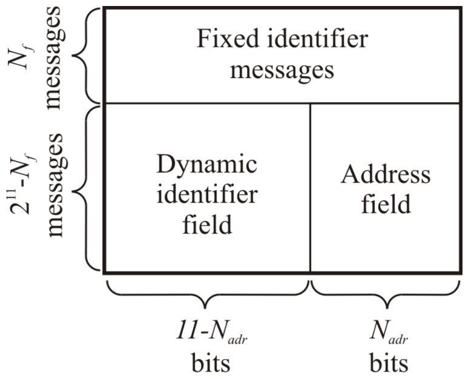

Fig. 2. *CAN identifier allocation table*

For the rest of information packages that have to be transmitted through the network each node will have to fight to receive the CAN network. It is known that if there are two nodes that need to transmit simultaneously a message, the message with the lower identification number will receive the CAN network. When a node of the CAN network needs to transmit a message it will allocate the biggest identification number ($2^{11} - 1$ for standard CAN) for its message and every time it tries to receive the CAN network and it fails, it will decrement with 1 the identification number. After several consecutive fails, depending on the load of the CAN network, the message will have the smallest identification number and will be transmitted.

This approach has a big disadvantage because if two nodes need to transmit a message in the same time, the two messages will receive the same identification number and will determine the CAN network to fail. To avoid this problem, each node will receive an address represented by a number and the last bits from the identification field of a message will be the address of the node (see Figure 2). In addition to that only the rest of the bits will be decremented.

The designer can calculate how many bits has to reserve for the address field with the equation:

$$N_{adr} \geq \log_2 N_n, \tag{2}$$

where $N_{adr}$ is the minimum number of bits required for address and $N_n$ represents the number of nodes from the CAN network.

The algorithm of the dynamic identifier allocation is presented in Figure 3, where had been considered that $k = 4$; *Id* is message identifier and *adr* represents the address of the node. An example of how evolves the identifier of the messages is presented in the Table 1, where the bold numbers represent the address of the node and the italic identification numbers represents the messages

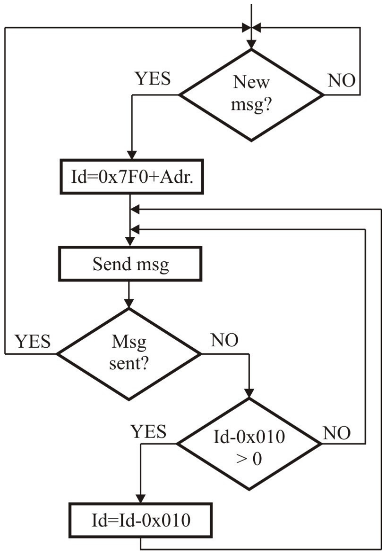

Fig. 3. *CAN identifier allocation algorithm*

that are transmitted. In the Table 1 can be observed that, at time $t_0$, the identification numbers of messages generated by Node1 and Node2 are the same except the node address. In conclusion the address field creates an advantage for the messages that had been generated by different nodes at the same time. This advantage is no more available for Node3 because after it sends a message at time $t_2$, when it wants to send a new one at time $t_3$ it has to wait after Node4, because Node4 has an older message.

### 3.2. The Message Transmission Time

This approach allows the designer of a distributed control system to determine the maximum transmission time of a message. He, first, has to determine for its CAN network the transmission time of a message, which represents the time between the beginning of the message transmission until the network is able again to transmit a new one. This time depends by the configuration of the CAN network, especially the baud rate.

The maximum transmission time of a message is direct proportional with the message transmission time and with the maximum possible number of fails to access the network.

Considering the worst case scenario the number of fails represents the number of decrements from the maximum value of the identification field to the minimum value of it plus the number of nodes with smaller address than the address of the considered node. In conclusion it has to be determined the number of message identifiers allocated to a node.

Because there are fixed identifier messages the total number of messages with dynamic identifier is:

$$N_{MDI} = 2^{11} - N_f\,. \tag{3}$$

As a result of that, the total number of messages allocated to a node is:

$$N_{NM} = \frac{N_{MDI}}{2^{N_{adr}}}\,. \tag{4}$$

Considering the equations (1), (3) and (4) the maximum transmission time of a message for node $x$ is:

$$t_x = (2^{11-N_{adr}} - 2^{k-N_{adr}} + N_x)\cdot T_0\,, \tag{5}$$

where: $t_x$ is the maximum transmission time, $N_x$ is the number of nodes with a smaller address than $x$ node address and $T_0$ is the transmission time of a message.

As a simple example, it had been have assumed a 125 Kbit/s CAN network. Assuming 11-bit identifiers and worst-case bitstuffing, the maximum length of each message is 125 bits. The maximum transmission time of each message is therefore 1 ms.

The dynamic identifier allocation algorithm was simulated in MATLAB and it resulted the graphic from Figure 4. The CAN network contains two nodes. The first one send the same message (with the fixed identifier 1898) in a continuous mode. It has the role of network load generator and its message identifiers are represented with blue.

*An example of message identifier evolution* Table 1

| | $t_0$ | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ |
|---|---|---|---|---|---|---|
| **Node1** | *0x7F3**0*** | - | - | - | - | - |
| **Node2** | 0x7F3**1** | *0x7F2**1*** | - | - | - | - |
| **Node3** | 0x7FF**2** | 0x7FE**2** | *0x7FD**2*** | 0x7FF**2** | *0x7FE**2*** | - |
| **Node4** | - | 0x7FF**3** | 0x7FE**3** | *0x7FD**3*** | - | - |

- no message to transmit.

The second node has implemented the dynamic allocation algorithm and its message identifiers are represented in Figure 4 with red. The empty circles represent the message identifiers that lost the CAN network because there are messages with bigger priority while the full colour circles represent the messages that are transmitted at certain moment of time.

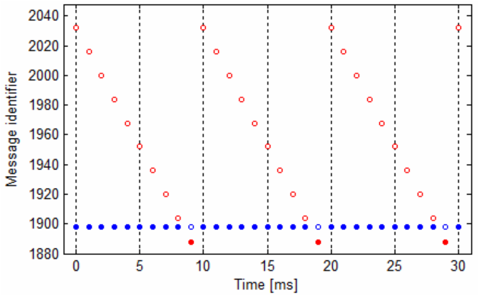

Fig. 4. *The simulation of the dynamic identifier allocation algorithm*

In the simulation had been considered that $N_{adr} = 4$, $k = 3$, $N_x = 3$ and with equation (5) results that the maximum transmission time of the messages generated by the second node is 130.5 ms. From simulation results that the transmission time is 9 ms.

### 3.3. The Message Recognition

In normal CAN applications the identifier field is used for message priority determination and for message recognition. Because the algorithm changes the identifier field of a message, the receiver node can determine only the address of the transmitter node. But one node may need to transmit several messages. One solution is to use the first byte from the data field for message recognition but it is not optimal.

The solution is to allocate in the identifier field a few bits for message recognition.

Considering that the number of bits allocated for the message recognition is $N_{mi} = 4$, in standard CAN protocol remains only 3 bits for dynamic identifier field. For this reason it will be used the extended CAN protocol and results the CAN identifier allocation table from Figure 5.

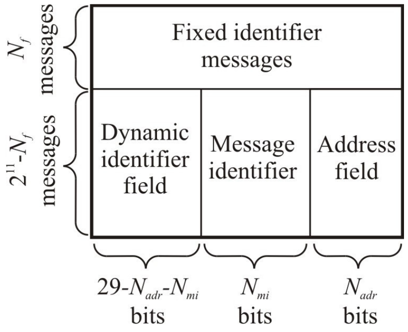

Fig. 5. *CAN identifier allocation table with message recognition field*

In this case the transmission time of a message is:

$$t_x = (2^{29-N_{adr}-N_{mi}} - 2^{k-N_{adr}-N_{mi}} + N_x) \cdot T_0, \quad (6)$$

where $N_{mi}$ represents the number of bits allocated for message identifier field.

## 4. The Experimental Validation

To test the models presented in this paper had been used a distributed control system with a CAN network, like the one presented in Figure 6. The three nodes of the CAN network are represented by three Dice-Kit developing boards from Fujitsu. The Dice-Kit 1 has the role of network load generator, because he always tries to transmit a high priority message. The Dice-Kit 2 and 3 are used to send different messages with dynamic identification fields and test their transmission times. A computer is also connected to the network using a CANcardX adaptor from Vector and is used to monitor the messages transmitted through the CAN network.

In the first experiment the Dice-Kit 2 board has a program that always transmits a message with an identification number bigger than the one of the message from the first board. The result is that the Dice-Kit2 board never managed to transmit the message.

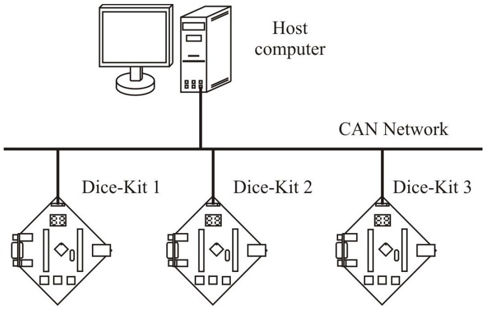

Fig. 6. *The distributed control system used for validating the model*s

In the second experiment the Dice-Kit2 board sends dynamically identifier allocated messages and it is able to send its messages and further more the transmission time is smaller than the maximum transmission time calculated with the equation (5).

In the last experiment had been implemented the identifier allocation algorithm with message recognition and had been validated the equation (6) and the receiver ability to recognise the message.

## 5. Conclusions

The presented algorithm allows the designer to exactly estimate the maximum transmission time of a message. On one hand the presented model has the big advantage of knowing the maximum transmission time of a message, but on the other hand it has several small disadvantages:

- it uses supplementary computation power from the processor of the node;
- the network can be extended only if the new node has implemented the distributed scheduler algorithm and if there are free addresses on the network;
- the verification of the messages validity is harder.

Taking into account the advantage and the disadvantages, the model is perfect for the multi-axis motion control of an articulated arm robot with a distributed control structure, because:

- the control system need to know the maximum transmission times;
- the intelligent drives have very powerful digital signal processors, so the computation power used by the algorithm does not matter;
- the control system does not need extensions.

Further more, the presented algorithm can also be used in other control system where the transmission times of a message are a critical factor and where the designer knows from the beginning the exact number of the network nodes.

Evaluating the equation (5) results that:

- all the messages from the network have relatively the same maximum transmission time, because in the majority of cases $N_x$ is very small compared to $2^{11-N_{adr}}$;
- the value of $t_x$ is relatively big considering that $2^{11-N_{adr}} - 2^{k-N_{adr}} + N_x$ is a big number.

As a result of that this solution is appropriate for distributed control systems where all the messages have the same priority and do not need very small transmission times.

For the systems where the messages must have different priorities, the designer can allocate smaller beginning values for the identification field of the messages or can increment the priority of the messages with a value bigger than 1. These changes reduce the maximum transmission time.

In the next developing stage the transmission algorithm will be modified in order to permit the transmission of messages with different priorities.

## Acknowledgment

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.

## References

1. Davis, R.I., Burns, A., Bril, R.J., Lukkien, J.J.: *Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised.* In: Real Time Systems **35** (2007) No. 3, p. 239-272.
2. Gupta, R.A., Chow, M.Y.: *Networked Control Systems: Overview and Research Trends*. In IEEE Trans. On Industrial Electronics **57** (2010) No. 7, p. 2527-2535.
3. He, F., Tong, W., Wang, Q.: *Synchronization Control Strategy of Multi-motor System Based on Profibus Network.* In: Proc. IEEE Inter. Conf. on Automation and Logistics, Jinan, 2007, p. 3099-3034.
4. He, F., Zhao, S.: *Research on Synchronous Control of Nodes in distributed Network Systems.* In: Proc. IEEE Inter. Conf. on Automation and Logistics, Qingdao, 2008, p. 2999-3004.
5. Jouve, D., Bui, D.: *CANopen Servo Drives Provides High Performance Motion Control*. In: Proc. 7th Int. Intelligent Motion Conf., Nuremberg, 2002, p. 1-6.
6. Lee, H.H., Jeong, U.H.: *Design of Distributed Scheduler on CAN for Real-Time Networking*. In: Proc. 5th Russian Korean International Symposium Science and Technology, Tomsk, 2001, p. 22-25.
7. Pereira, C.E., Carro, L.: *Distributed Real-time Embeded Systems: Recent Advances, Future Trends and Their Impact on Manufacturing Plant Control*. In: Annual Reviews in Control **31** (2007) No. 1, p. 81-92.
8. Pinho, L.M., Vasques, F.: *Reliable Real-Time Communication in CAN Networks*. In: IEEE Trans. On Computers **52** (2003) No. 12, p. 1594-1607.
9. Sebestyen, G.: *Industrial Informatics.* Cluj Napoca. Editura Albastră, 2006.