Real-time Communications for Distributed Control Systems

Dan PUIU[1], Florin MOLDOVEANU[2], Caius SULIMAN[3]
Transilvania University of Brasov
bld.Eroilor nr.3,Brasov
[1]*puiudan@unitbv.ro,* [2]*moldof@unitbv.ro,* [3]*caius.suliman@unitbv.ro*

***Abstract* — Due to the quick evolution of manufacturing processes, the demand for more flexible automation systems is on the rise. To answer 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 change 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 ensure a maximum transmission time for a message. A distributed control system is developed and the proposed scheduler is experimentally validated.**

***Index Terms* — Distributed control, delay estimation**

I. 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 the rise [1]. To answer to these requirements, the "traditional" solutions are replaced by distributed architectures based on intelligent drives. Unlike the "classical" solution, where all the control structure tasks are in a central controller, the distributed control architecture is based on the technique of the industrial communication networks (with their specific protocols) and on the digital signal processing technology and in this way the control tasks can be decentralized and can be solved by the intelligent drives [2].

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 [3]. As a result of that, several studies had been made to ensure real-time communications capacities for the field-busses.

Tovar and Vasques, in the paper [3], modify the medium-access control (MAC) level of a PROFIBUS network, by im-plementing of a low-priority message counter, which achieves a known transition time for the low-priority messages.

The real-time communication aspects for CAN networks were studied by Pinho and Vasques [8], an they designed a middleware, with three different protocols, that can solve the problems introduced by inconsistent, duplicated or undelivered messages.

In the paper [6], Lee and Jeong design a distributed sched-uler that ensure the real-time capabilities for a CAN network. They consider that in the CSMA/CD priority protocol some messages have significant delays because they have small priority and the network is occupied by the 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 contains on increasing the priority of a message gradually each time the transmitter node fails to send the message because there it is another message with a bigger priority. The paper offers a mathematical model, which calculates the maxim transmission time of a message. Further more there are presented three ways of how can be increased or decreased the maxim transmission time of a message.

The paper is organized into six sections including this introduction. Section 2 discusses the main problems about the delays introduced by the industrial communication networks of a distributed control system. There are also presented several ways of how can be compensated the delays effect. In section 3 are briefly presented the communication and synchronization mechanisms of a CAN network. Section 4 presents at the be-ginning the dynamic scheduling algorithm and it ends with the mathematical models of the messages maximum transmission times. In section three are presented three experiments that prove the models presented in previous section. Finally, conclusions and further work are presented in section 6.

II. REAL TIME SYSTEMS WITH DELAYS

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

The transmission times can be divided in three main categories [7]:

- the processing time, which represents the necessary time required by a nod of the network to prepare and send a package of information;
- the network time, which is the time required by an in-formation package to go through the network;
- the synchronization time, witch represents the time when 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 poor. The network must have a deterministic behavior, which allows the evaluation, from

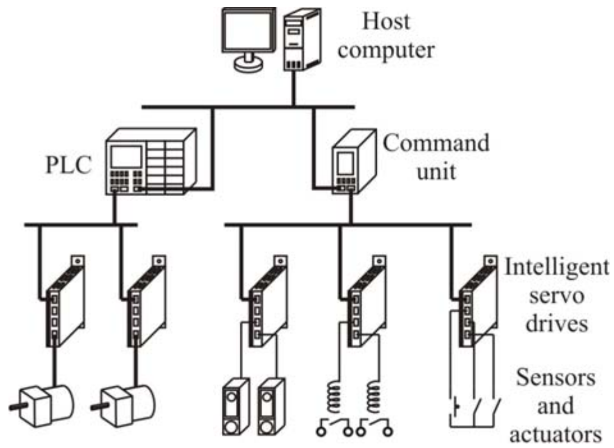

Figure 1. Distributed control architectures based on four layers.

the design stage, of the transmission time of a message [5]. The industrial communication networks used in distributed control applications must be optimized in order to guarantee a maximum transmission time. This requirement can be archived by combining several techniques [7]:

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

The deterministic access to the communication medium can be achieved by using communication protocols which eliminates the hazard generated when a two messages collision appears and which guarantee, for a nod of the network, a default period to access the communication medium. 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 can not transfer big volumes of data, but allows the designer to mathematically evaluate the maxim transmission time for a message.

Generally, the distributed control systems have a stable configuration from point of view of hardware resources involved. As a result of that the designer can evaluate the load of the communication medium and the configuration of the messages flux [9]. The majority of the tasks of a node are periodical in time (for ex. data acquisition, data processing and commands trans-mission). 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 [9]. In this case the objective of the message planning algorithm is to find an optimum solution, which respect all the transmission times, if it exists.

III. THE CAN COMMUNICATION PROTOCOL

The medium access control (MAC) mechanism adopted in CAN [10], [11] 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 net-works, 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 inter-face 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 com-pares 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 trans-mission 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 phase is carried out in a small and deterministically bounded time, which does not depend on the number of nodes effectively involved. If up to M different identifiers are allowed on the network, the asymptotic complexity of this operation is log2M.

When it receives a frame, the MAC sublayer does not per-form 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 for-warded to the application programs which are executing on the node.

There are four types of frames that can be transferred in a CAN network. Two are used during the normal operation of the CAN network: the Data Frame, which is used to send local data, and the Remote Frame, which is used to request remote data. Besides these two frames, there are also the Error Frame, which signals the detection of error states in the CAN network, and the Overload Frame, which is used by nodes requiring extra delays before the transmission of Data Frames.

Fig. 2 shows the structure of a Data Frame (specific fields: SOF, Identifier, Control, DLC, CRC, ACK, and EOF are de-scribed in [5]). A Remote Frame has the same identifier and structure (without data field) as the remotely requested Data Frame.

At the physical layer, frames are transmitted using the NRZ (Non-Returning to Zero) coding technique, with the insertion of stuff bits. That is, whenever there are more than five equal consecutive bits (up to the end of the CRC Field), there is the insertion of an opposite bit in the frame. This opposite bit will be detected and removed by the physical layer at the receiving side. This bit stuffing technique ensures that, in the normal behavior, there will never be more than five consecutive equal bits on the bus.

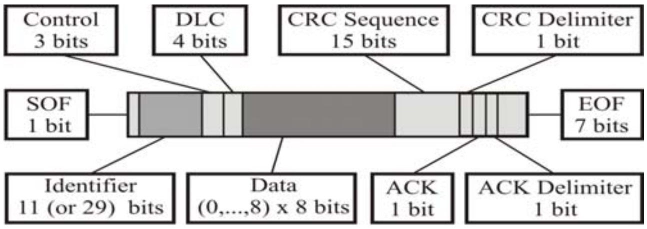

Figure 2. Structure of a data frame.

IV. THE DINAMIC 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 is working very well in case of small size nodes or messages, but in large distributed control systems, the messages with lower priority may not be 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 will become a high priority one. This approach is appropriate because a low priority message becomes more urgent with the passing of time.

A. 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 model presented in this paper is for standard CAN, but it can be easily adapted 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 3, the user, when designs the message allocation table, can allocate N_f 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 three and it implies eight priority messages.

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 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 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.

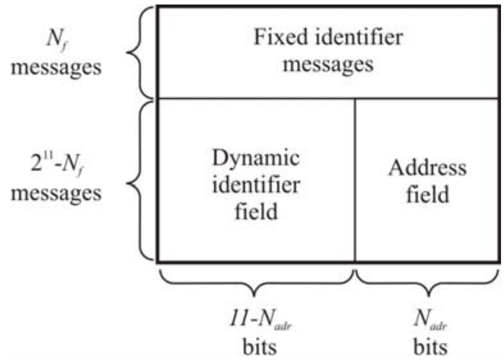

Figure 3. CAN identifier allocation table.

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 number of a message will be the address of the node (see Fig. 3). In addition to that only the rest of the bits will be decremented.

The designer can calculate how many bits have 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 4, 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 that are transmitted in the certain moment of time. In the table can be observed that, at time t_0, the identification numbers of messages generated by Node1 and Node2 are the same except the 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.

TABLE I. AN EXAMPLE OF MESSAGE IDENTIFIER EVOLUTION

	t_0	t_1	t_2	t_3	t_4	t_5
Node1	*0x73***0**	-	-	-	-	-
Node2	0x73**1**	*0x72***1**	-	-	-	-
Node3	0x7F**2**	0x7E**2**	*0x7D***2**	0x7F**2**	*0x7E***2**	-
Node4	-	0x7F**3**	0x7E**3**	*0x7D***3**	-	-

- no message to transmit;

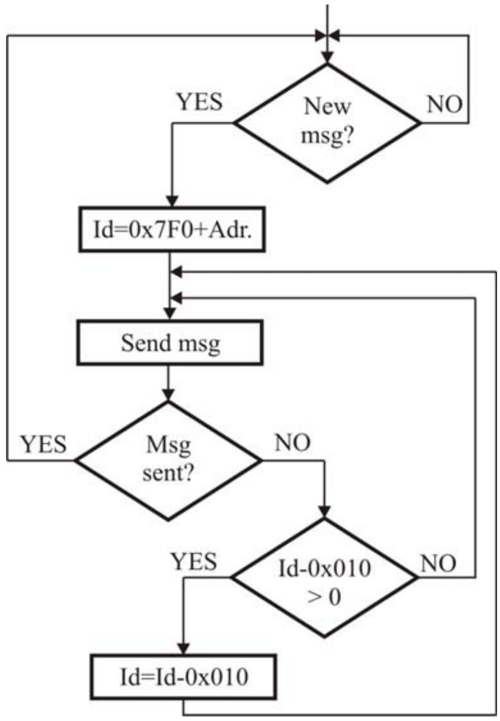

Figure 4. CAN identifier allocation algorithm.

B. The Transmission Time of an Message

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 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 rep-resents the number of decrements from the maximum value of the identification field to the minimum value of it plus the number of nodes with the address smaller 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:

$$N_{MDI} = 2^{11} - N_f \qquad (3)$$

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

$$N_{NM} = \frac{N_{MDI}}{2^{N_{adr}}} \qquad (4)$$

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

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

where t_x is the maximum transmission time, N_x is the number of nodes with a smaller address than the *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 5. The CAN network contains two nodes and 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.

The second node has implemented the dynamic allocation algorithm and its message identifiers are represented in figure 5 with red. The empty circles represents the identifiers that lost the CAN network because there are messages with bigger priority while the full colored circles represents the messages that are transmitting at a certain moment of time.

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 130,5ms. From the simulation results that the transmission time is 9 ms.

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 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.

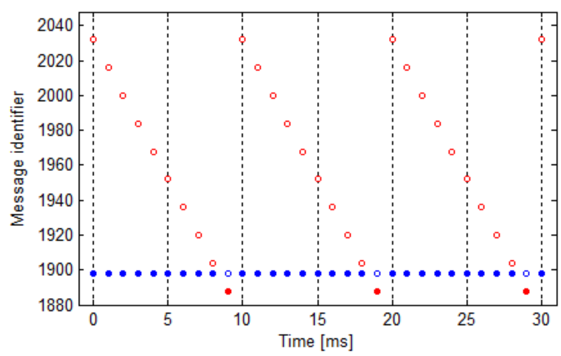

Figure 5. The simulation of the dynamic identifier allocation algorithm.

For the systems where the messages must have different priorities, the designer can allocate smaller beginning values for the identification field of the messages. For example the message transmitted by the Node4 from Table 1, instead of starting at 0x7F**4** he could start at 0x0F**4** and would had been the first message transmitted.

The transmission time for the messages with smaller beginning for the identifier field is:

$$t_{xbv} = \left(2^{11-N_{adr}} - 2^{k-N_{adr}} + N_x - 2^{11} + bv\right) \cdot T_0 \qquad (6)$$

where *bv* is the beginning value. From equation (6) results that if *bv* is smaller, the t_{xbv} is smaller, fact that is also proved by the simulation from figure 6, where the second node, in the same conditions as in the first case, transmits a message in 5 ms instead of 9 ms. In this case, the second node start at 0x7B**0** instead of 0x7F**0**

Another method to implement distributed control systems with message that have different priorities, respective different maxim transmission times, is to decrease identification field with a value bigger than 1. In this case the maxim transmission time is:

$$t_{xdv} = \frac{2^{11-N_{adr}} - 2^{k-N_{adr}} + N_x}{dv} \cdot T_0 \qquad (7)$$

where *dv* is the decrement value and if *dv* is bigger, the transmission time is smaller. In the simulation from figure 7 the second node starts at 0x7F**0** but it decrements the identification field with 3.

The last possibility is to combine the last two methods, and the transmission time results:

$$t_{xbvdv} = \frac{2^{11-N_{adr}} - 2^{k-N_{adr}} + N_x - 2^{11} + bv}{dv} \cdot T_0 \qquad (8)$$

The simulation of the last case is presented in figure 8.

V. 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 9. 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 message with a high priority from the domain [N_f ;0x7FF] and with the node address 0. 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.

In the second experiment the Dice-Kit2 board sends dynamically identifier allocated messages and he is able to send its messages and further more by using the modification of the beginning value technique respectively the modification of the decrement value or bought of them, the transmission time is smaller.

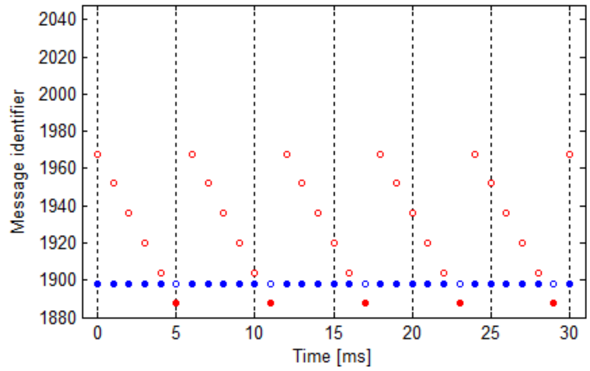

Figure 6. The simulation of the dynamic identifier allocation algorithm with smaller beginning value.

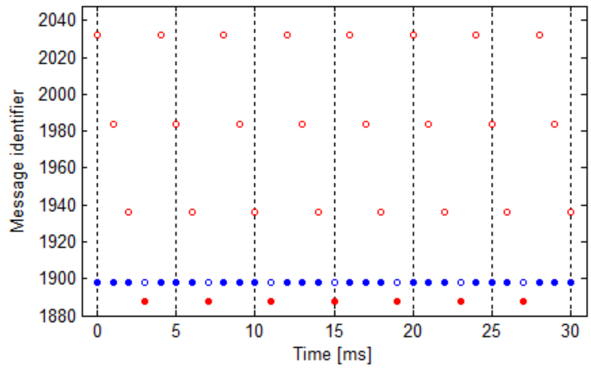

Figure 7. The simulation of the dynamic identifier allocation algorithm with bigger decrement value.

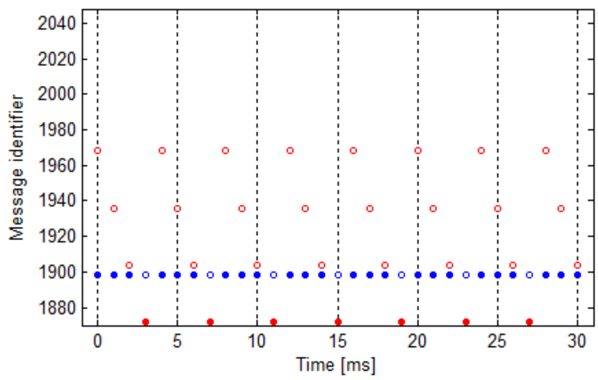

Figure 8. The simulation of the dynamic identifier allocation algorithm with bigger decrement value and smaller starting value.

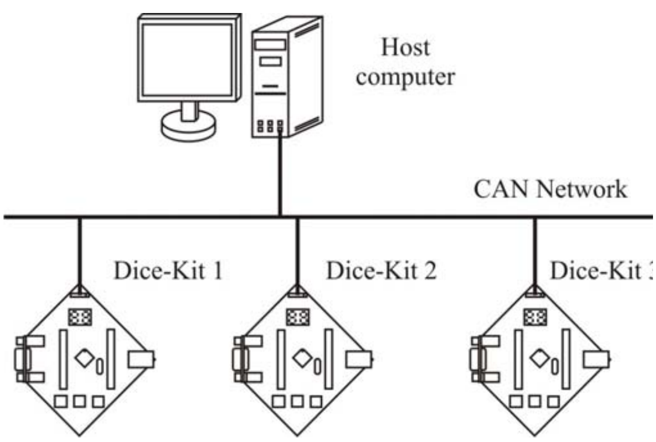

Figure 9. The distributed control system used for validating the models.

In the last experiment had been on-line measured, using the facilities of the CANcardX the transmission time T_0 of a message. Than in the Dice-Kit 2 board had been implemented the beginning value modification algorithm, respective the decrement value modification algorithm in the Dice-Kit3 board. Using T_0 had been calculated the corresponding maximum transmission times for the two boards and had been experimentally validated that the transmission times are smaller than those calculated.

VI. CONCLUSIONS

The presented algorithm allows the designer to exactly estimate the transmission time as long as no message with fixed identifier is transmitted, which is assumed that are used only for high emergency problems in the system. As a result of that the designer can develop easier and more efficient the control loops for the distributed control system.

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.

The next developing stage of the dynamic scheduling algorithm will improve the message recognition, because at this moment the receiver can not determine the significance of the message from the identification field and it has to be used the first byte from the data field for message coding. The solution for this is to allocate a few bits on the identifier field for the message coding.

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] C.E. Pereira and L. Carro "Distributed Real-time Embeded Systems: Recent Advances, Future Trends and Their Impact on Manufacturing Plant Control", Annual Reviews in Control, vol. 31, no. 1, pp. 81÷92, 2007.

[2] D. Jouve and D. Bui, "CANopen Servo Drives Provides High Performance Motion control." Proc. 7th Int. Intelligent Motion Conf., Nuremberg, Germany, 2002, pp. 1-6.

[3] E. Tovar and F. Vasques "Real-Time Fieldbus Communication Using Profibus Networks:, IEEE Trans. on Electronics, Vol. 46, No. 6, pp. 1241-1251, 1999.

[4] F. He and S. Zhao "Research on Synchronous Control of Nodes in distributed Network Systems", Proc. IEEE Inter. Conf. on Automation and Logistics, Qingdao, China, 2008, pp. 2999-3004.

[5] F. He, W. Tong and Q. Wang "Synchronization Control Strategy of Multi-motor System Based on Profibus Network", Proc. IEEE Inter. Conf. on Automation and Logistics, Jinan, China, 2007, pp. 3099-3034.

[6] H.H. Lee and U. H. Jeong "Design of Distributed Scheduler on CAN for Real-Time Networking", Proc. 5th Russian Korean Int. Symp. Science and Technology, Korus2001, Tomsk, Russia, 2001, pp. 22-25.

[7] G. Sebestyen, "Industrial Informatics", (romanian language), Ed. Albastră, Cluj-Napoca, Romania, 2006.

[8] L. M. Pinho and F. Vasques "Reliable Real-Time Communication in CAN Networks", IEEE Trans. On Computers, Vol. 52, No. 12, pp. 1594-1607, 2003.

[9] R. I. Davis, A. Burns, R. J. Bril and J. J. Lukkien "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised", Real Time Systems, Vol. 35, No. 3, pp. 239-272, 2007.

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

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