IMPROVING THE RESPONSIVENESS OF THE SYNCHRONOUS MESSAGING SYSTEM IN FTT-CAN

Paulo Pedreiras, Luís Almeida, José A. Fonseca
pedreiras@alunos.det.ua.pt, {lda,jaf}@det.ua.pt

DET-IEETA
Universidade de Aveiro
P-3810-193 Aveiro, Portugal

Abstract: A flexible distributed real-time communication system must support modifications to the message set which it conveys. These changes can require a degree of responsiveness ranging from a few milliseconds to some seconds.

In FTT-CAN protocol, the responsiveness of the synchronous communication system depends on the plan duration, which, in general cannot be set arbitrarily short. This paper presents a method that uses the asynchronous messaging system to temporarily convey the synchronous messages until the synchronous messaging system can handle them. Furthermore, methods to evaluate offline if a set of requests for modifications can be timely handled are presented. *Copyright © 2000 IFAC*

Keywords: Real-time communication, Real-time systems, Communication protocols, Distributed computer control systems, Fieldbus.

1 – INTRODUCTION.

1.1 – Levels of system responsiveness.

During normal operation, processes controlled by real-time computer systems experience phases of continuity as well as of changes (Fohler, 1995). Changes in the environment can be reflected in the real-time system as modifications to the task set, as well as to the message set when the system is distributed. Kopetz (Kopetz, 1997) states that resource utilization is improved if only those tasks that are needed in a particular operational mode are scheduled. In these circumstances the message set can change too. Consequently, a flexible real-time communication system must support changes to the message set which it conveys, namely allowing dynamic creation and elimination of message streams and change of parameters of existing ones. However, in the context of real-time systems, the timeliness of the communication system must always be guaranteed, even while changes to the message set are made. Thus, the requests for changes must be supported in a way that new requirements are handled within adequate response time and without disturbing the timeliness of the remaining message streams.

The maximum time allowed between a change in the environment and the respective reaction in the control system is a critical parameter, which depends on the dynamics of both environment and control system. For example, consider a car traction control system in which a central unit receives information from wheels speed sensors and actuates on the breaking system if it detects that one or more wheels are losing grip. This kind of system can be implemented in a distributed fashion and, to improve resource utilisation, the wheels speed sampling rate might vary according to the driving conditions. When driving in a road with good adherence, the sampling rate can be lower. If the car suddenly enters a slippery road, the traction control system faces a sudden change in its operational conditions, requiring, among other things, a higher sampling rate. Since a car running at 100 Km/h travels 27,7m in a second, if the communication system requires 100ms to adjust the message set properties related to the sampling rate of the wheel sensors, the car travels about three meters until the system behaves accordingly to the new environmental conditions, jeopardizing the security of the driver and, eventually, other people. In this system a responsiveness of a few milliseconds is required. However, when (hopefully) the car returns onto good road again, the sampling rate can be reduced.

1.2 – About this paper.

The FTT-CAN (Flexible Time-Triggered communication on CAN) protocol is well suited to support the kind of system described above. Particularly, its synchronous message system, based on the time-triggered paradigm, can efficiently convey the message streams resulting from the periodic sampling of the wheels speed sensors.

This paper focuses on the responsiveness of the synchronous message system of FTT-CAN. It will be shown that some protocol key parameters cannot be adjusted only in function of the required responsiveness since they have wider implications. A method to improve the responsiveness to changes made to the synchronous message set is presented, and its implications in the protocol architecture are analysed. In section 2 the FTT-CAN synchronous and asynchronous messaging systems are briefly presented. The new method used to improve the responsiveness to changes in the communication requirements is presented in section 3. Section 4 presents methods to evaluate at pre-runtime if a given set of change requests can be timely handled. Finally, section 5 concludes the paper.

2. FTT-CAN BRIEF PRESENTATION.

The FTT-CAN protocol has been briefly presented in (Almeida *et al*., 1999) and further developed in (Almeida, 1999). A feature that distinguishes this protocol from other proposals concerning time-triggered communication on CAN (Peraldi *et al*., 1995) is that it supports dynamic communication requirements by using centralized scheduling with on-line admission control whilst the communication overhead is kept low by using the native distributed arbitration of CAN.

A Synchronous Requirements Table (SRTable) holds the properties of the synchronous message streams, namely: identifier, period, relative deadline, initial phase, maximum transmission time and priority. Using this information, the scheduler builds static schedules for consecutive fixed duration periods of time called plans. The creation of a plan is concurrent with the dispatching of the previous one.

As usual in table-based scheduling, a finite time resolution in used to express all the properties of the message set. This basic time unit is called Elementary Cycle (EC). The EC duration is fixed and set at pre-run-time.

Within each EC, the protocol supports two types of traffic, synchronous and asynchronous. The former one is time-triggered and its temporal properties (i.e. period, deadline and relative phasing) are represented as integer multiples of the EC duration. A particular node (Master), scans the current plan and generates a periodic message used to synchronize all other nodes in the network. The transmission of this message represents the start of one elementary cycle (EC) and is known as EC trigger message (TM).

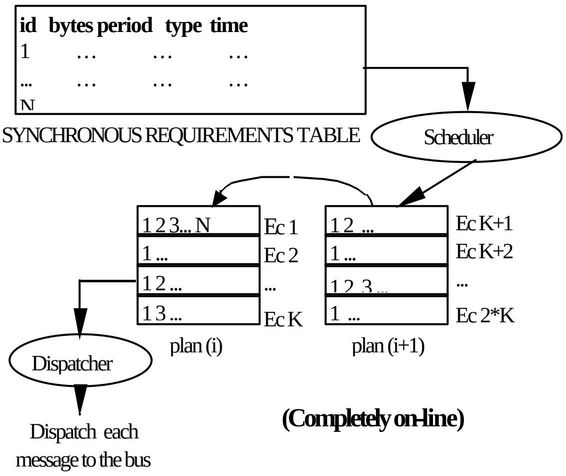

Fig. 1. The planning scheduler

The EC trigger message conveys in its data field the identification of the synchronous messages that must be transmitted by the producer nodes in that EC. The nodes that identify themselves as producers by scanning a local table containing the messages to be produced / consumed, transmit the respective synchronous messages in the synchronous phase of that EC (fig. 2). Collisions on bus access are resolved by the native distributed MAC protocol of CAN. This is known as the synchronous messaging system (SMS).

The FTT-CAN protocol also supports asynchronous traffic for event-triggered communication, with external control. This sort of traffic is transmitted during the periods of the EC not used by the synchronous messages. However, depending on how the desired temporal isolation between these two sorts of traffic is enforced, the asynchronous messaging system (AMS) can operate in one of two modes. In controlled mode any asynchronous message is transmitted only if it is guaranteed not to interfere with the timeliness of the EC trigger message or of the synchronous messages. This way, the temporal isolation between the synchronous and asynchronous traffic is strictly guaranteed. In uncontrolled mode, stations wishing to transmit asynchronous messages can try to do it as soon as they receive the respective requests from the application layer.

Although these messages may now cause a certain blocking to the transmission of synchronous ones, such blocking can be upper bounded by using a proper choice of identifiers. Thus, a bounded level of interference between the synchronous and asynchronous traffic is allowed, in exchange of a higher flexibility. In this case legacy nodes not committed with the FTT-CAN protocol are supported, if the identifiers are properly selected.

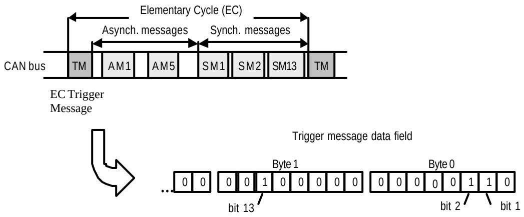

Fig. 2. EC Trigger Message data contents.

3 – IMPROVING FTT-CAN RESPONSIVENESS.

3.1 – Flexibility limits.

Once a change request is made concerning the current synchronous message set, a certain period of time elapses until that request takes effect at the bus level. This is referred to as the synchronous transient response time (STRT). Note that, when using SMS alone, the scheduler must, first, build a plan using the new requirements.

The STRT can be decomposed in three parts (see fig. 3): the time from the request to the end of the current plan, the plan in which the scheduler starts to take into account the new requirements, and finally the initial phase (φ) of the message stream relative to the beginning of the plan where changes are already reflected. The minimum value (marker A in fig. 3) occurs when cumulatively the request is made just before the end of one plan, and φ is zero. The maximum value occurs if the request is issued just after the beginning of one plan (marker B in fig. 3), and the initial phase has its maximum value. Therefore, the absolute bound for the synchronous transient response time, when using the SMS alone, varies between one and two plans plus the initial phase (as defined above) ($LPlan<STRT_{SMS}<2*Lplan+\varphi$). Since the $STRT_{SMS}$ is a direct function of the plan duration, the responsiveness can be improved by shortening the plan. However, the reduction of the plan duration increases the CPU load (Almeida *et al.*,1999)(Almeida,1999). Below a given value, the scheduler might not have enough time to build next plan in time, that is, before the dispatcher processes the current one. Moreover, some interesting properties of the planning scheduler, like the look-ahead feature (Almeida, 1999), are negatively affected by the reduction of the plan length. As a consequence, there is a lower bound to the plan duration, limiting the responsiveness that can be achieved this way. Another way to improve the responsiveness while still using the SMS alone is to start the scheduler as late as possible. Since the worst case execution time of the scheduler (wcetSch) can be estimated on-line (Almeida, 1999), using this approach the synchronous transient response time can be bounded to the interval: $wcetSch < STRT_{SMS} < LPlan+wcetSch+\varphi$, where LPlan stands for the plan duration.

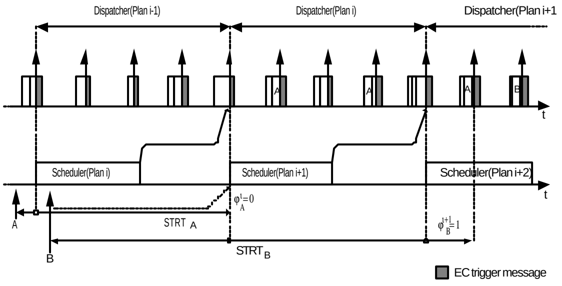

Fig. 3. SMS Responsiveness bounds.

3.2 – Improving FTT-CAN responsiveness.

As seen above, the responsiveness of the SMS when considered alone is upper bounded by the plan duration plus the scheduler execution time. Since these cannot be made arbitrarily short, further improvement to the responsiveness of SMS in FTT-CAN requires that change requests are handled even during the current plan, bypassing the planning scheduler for a short period of time, but without disturbing the other synchronous messages already scheduled. The proposed way of achieving this, consists in using the asynchronous messaging system (AMS) to produce the required message(s) until the requested changes are handled by the SMS as described in the previous section. This is shown in figure 4. Notice that the message associated with the change request (e.g. a new message stream) begins to be transmitted using the asynchronous message area. After the dispatcher starts processing the plan in which the new message parameters are reflected (plan i in figure 4), the system resumes normal operation, that is, as the message is included in the synchronous message area it is removed from the asynchronous one. The period of time during which the AMS is used to support the transmission of synchronous messages is referred to as synchronous support period (SSP). The Master station, by means of a specific control message, establishes the beginning and duration of the SSP for each change request. This message has higher priority than regular synchronous and asynchronous messages.

The following relationship can be established between the STRT with and without the AMS support:

$$STRT_{AMS} = STRT_{SMS} - SSP \qquad (1)$$

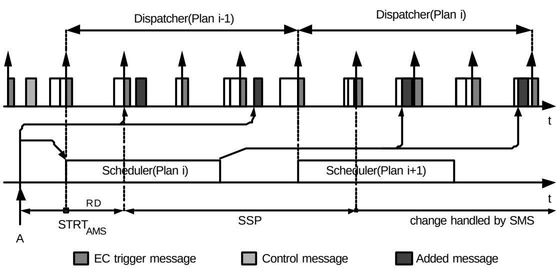

Fig. 4. Using the AMS to temporarily convey a new synchronous message.

If the change to the message set consists only in the addition of a new message, the process described above is sufficient. However, if the change request is performed over a message stream already present in the SRTable (e.g., to change the stream's period), the existing instances of the message in the SMS during the synchronous support period (SSP) should be eliminated. Those instances still use to the older parameters (before the change) while the updated instances are transmitted in the asynchronous area. Thus, the elimination is required to avoid replication of message production in both synchronous and asynchronous systems. The elimination is achieved by filtering the trigger message. This filter is applied to the trigger messages just before they are sent. To eliminate one stream present in one plan already built, it is only necessary to set a bit in the filter.

Depending on the type of the change request that is made, one or several of the following actions can result:

1. A change of one bit in the filter;
2. The production of a control message to signal the start and duration of the SSP (synchronous support period);
3. A set of data messages produced in the AMS, during the SSP.

If the change request consists in the elimination of one message stream, only action 1 is required. However, if the change request consists in adding a new message, control and data messages will be produced in the AMS during the SSP (actions 2 and 3). If the change request concerns a modification in the parameters of an existing message (ex. period), actions 1,2 and 3 are required.

3.3 – Implementation issues.

From the operational point of view, several steps must be performed in order to process the request for a change to the synchronous message set. In figure 5 a flowchart describing the operational diagram of the proposed method for improving the responsiveness of the planning scheduler is presented.

When a change request to the synchronous message set is made, a schedulability test must be performed in order to filter out changes that would result in a non-schedulable message set. However, for the purpose of this work, we will consider that any requested change has already been analysed and it does not compromise the message set schedulability. In case the on-line analysis is performed, its execution time must be included in the STRT. Current work is being carried out in order to reduce such execution time (e.g. by using simple schedulability tests) so that its impact on the response time is minimized.

When a change request is accepted, the change is made to the SRTable, and then it is evaluated whether the response time requirements, expressed as a deadline, can be handled by the SMS alone (Response deadline > $STRT_{SMS}$). If so, no further handling is necessary. Otherwise, two more steps must be performed. In first place it is verified if the request is made over a message already present in the SMS (change of period or elimination), and, if so, a request is made to the dispatcher to eliminate the message from the synchronous message area. Next, it is evaluated if the request implies to add a message; if so, a request is made to the AMS to start its production in asynchronous mode.

The start and duration of the temporary production of synchronous messages using the AMS is controlled by the dispatcher, which sends a control message to the respective producer station to notify it about the required action. During this period of time (SSP as defined before) each station produces the required messages autonomously. The communication overhead of this control protocol is thus one message per change request.

The start of production message (SP_SSP) must convey the ID of the message to be produced, its period (expressed in EC's), a release delay (also in EC's) that must be applied between the reception of this message and the effective start of stream production, and the number of instances that must be produced using the AMS. Seven data bytes are used, one for variable ID, and two for message period, release delay and number of instances.

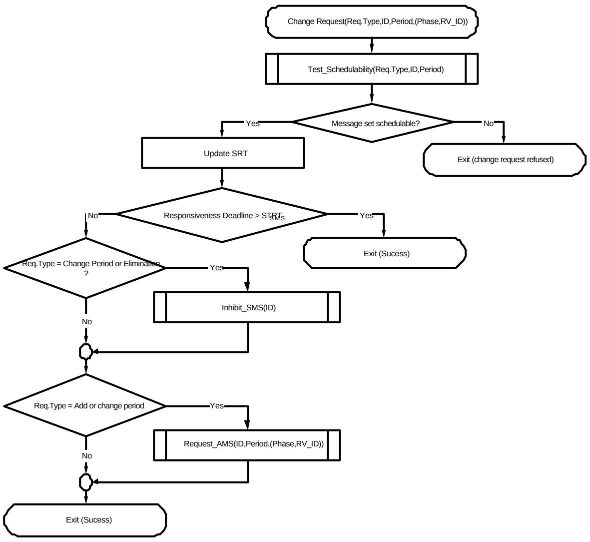

Fig. 5. Operational diagram.

4 – ANALYSIS OF THE SYSTEM PERFORMANCE.

During the synchronous support period (SSP), the control and synchronous messages corresponding to a change request are handled by the AMS, and will compete for the bus jointly with other asynchronous messages. If timeliness guarantees are required, an evaluation should be made since the bandwidth available to the AMS is limited. For this reason, in the following subsections sufficient conditions are presented, which allow to guarantee that a set of requests is handled within specific time bounds.

4.1 Bus demand and responsiveness.

As explained in section 3, during the SSP any new and modified messages are produced using the AMS. However, if the request is accepted by the schedulability test it means that the SMS has enough leeway to hold the message. As the AMS holds the remaining bandwidth, it can be concluded that the production of data messages during the SSP will use space borrowed by the AMS from the SMS. However, this argument requires that the start of synchronous support period (SSP) takes into account the phase of the variable. This is necessary to maintain the same relative phasing in both production periods, SSP and SMS, resulting in a smooth transition from one to the other.

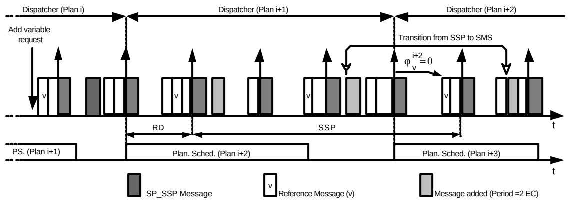

Fig. 6. Transition from SSP to SMS.

Consider for instance the example illustrated in figure 6, where a message is added with period of 2ECs and phase of 1EC relative to a reference message v. The SP_SSP message is sent by the Master Station, informing the respective producer node that it should start producing the new stream using the AMS with period of 2ECs and starting in the 2nd EC after the reception of the control message. This way, the release of the first message in the stream is appropriately delayed (RD in fig. 6) so that the relative phasing is the same in SSP as in SMS.

In order to evaluate where the SSP should start, the Master node must calculate which will be the initial phase relative to the start of the plan of the first instance of the message produced in the SMS. Notice that this plan (i+2 in fig. 6) is not yet built at the request instant.

However, knowing the initial phase of a variable v on plan i, its phase in plan (i+1) is given by equation (2), where W is the length of the plan (in ECs) and P_v is the period of variable v (also in ECs).

$$j_v^{i+1} = \left\lceil \frac{W - j_v^i}{P_v} \right\rceil * P_v - (W - j_v^i) \tag{2}$$

When the request for a change is performed, the current scheduler instance (i+1 in fig.6) can be either terminated or still in execution. In the former case, the next plan (i+1) is already built and φ_v^{i+1} is known. Thus, equation 2 is applied once, only, to determine φ_v^{i+2}. In the latter case, plan i+1 is not built yet and thus, equation (2) must be applied twice to evaluate φ_v^{i+2} based on φ_v^i.

Knowing the relative phase of a message u with respect to a reference message v (Ph_u^v), and the initial phase of this one (φ_v^{i+2}) the number of ECs between the SP_SSP and the first instance of the message stream produced in the SMS (fig.6) is given by equation (3), where W is the length of the plan, $curEC^i$ is the EC where the request is handled within plan i (1<=curEC<=W) and Ph_u^v is the phase of the message being added (u) relatively to message v.

$$L_{RD+SSP_u} = W - curEC^i + W + j_v^{i+2} + Ph_u^v \tag{3}$$

Finally, the number of instances that must be produced during the SSP (NI_{SSP_u}) is given by:

$$NI_{SSP_u} = \left\lfloor \frac{L_{RD+SSP_u}}{P_u} \right\rfloor \tag{4}$$

The release delay of the first instance relative to the reception of the control message (RD) is given by.

$$RD_{SSP_u} = L_{RD+SSP_u} - NI_{SSP_u} * P_u \tag{5}$$

When using the AMS support to increase the responsiveness to changes in the synchronous message set, the synchronous transient response time ($STRT_{AMS}$) is substantially reduced (fig. 4). In fact, its worst-case value occurs when the request is done before the beginning of the synchronous window of one EC and the respective control message (SP_SSP) can only be transmitted in the asynchronous area of the following EC. Unless the accumulated number of control messages, due to the queuing of several requests, is greater than the available space in the asynchronous window, the $STRT_{AMS}$ will be less than 2 ECs, plus the release delay RD. Since $0 \le RD \le P_u - 1$, the worst-case value of the responsiveness achieved by this method, expressed in ECs, is given by equation (6), where P_u is the period of variable u, measured in ECs.

$$STRT_{AMS_u} < P_u + 1 \tag{6}$$

4.2 Pre-run-time analysis.

The SP_SSP control messages are transmitted in the asynchronous windows, competing for the bus together with other asynchronous messages. Thus, to guarantee that the bound in (6) is respected, an evaluation at pre-run-time must be performed.

As seen in the previous section, during the SSP the production of the synchronous messages is made in space borrowed from the SMS by the AMS. However, the same assumption cannot be made concerning the control messages. For these, it must be evaluated if the minimum bandwidth reserved to the AMS at configuration time (LAW – minimum length of an asynchronous window) is enough to handle them in a timely way.

In (Pedreiras *et al*., 2000) the authors presented the analysis of the FTT-CAN Asynchronous Messaging System. Under controlled mode, due to a possible idle-time insertion (α), the effective bus time available in each EC for asynchronous transactions is less than LAW and it is given by (7).

$$LAW_{UT} = LAW - \alpha \tag{7}$$

The inserted idle-time term (α) is intended to prevent the AMS from interfering in the SMS and is bounded by the transmission time of the longest asynchronous message (CA), which is given by equation (8), where Ca_i is the maximum transmission time of asynchronous message i and Na is the number of asynchronous messages.

$$CA = \max\{Ca_i\},\ i = 1..Na \tag{8}$$

In a worst-case situation, when using either higher transmission rates or a low processing power microcontroller, the Master will take more time to handle a change request (i.e. perform the previous calculations) than to send the respective SP_SSP message. In this situation, the Master must release the bus between any consecutive SP_SSP messages. Consequently, in the meanwhile, the bus can be

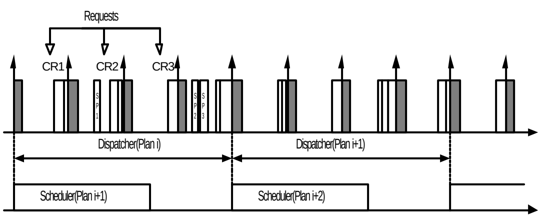

Fig. 7. Load due to SSP control messages.

taken by another asynchronous message which will cause a blocking to the following SP_SSP message. The maximum duration of such blocking is also given by CA (8). This same blocking can happen every time the Master tries to send an SP_SSP message.

Therefore, if there are N_{CR} change requests pending, in order to guarantee that the respective SP_SSP messages can be sent in one EC so that the bound in (6) is respected, the following condition must be verified:

$$N_{CR} * [Len(SP\_SSP) + CA] <= LAW_{UT} \quad (9)$$

This expression establishes a relationship between LAW and the maximum number of simultaneous change requests that the system is expected to handle so that the STRT of each request is still bounded by (6).

However, if there exists a minimum inter-arrival time (mit_{CR}) between the change requests, e.g. due to a serialization imposed by the schedulability analyser, the following condition results, where LEC is the length of the EC:

$$\left\lceil \frac{LEC}{\mathrm{mit}_{CR}} \right\rceil * [Len(SP\_SSP) + CA] \leq LAW_{UT} \quad (10)$$

Finally, it is important to remind that equations (6), (9) and (10) are pessimistic. In (6) the maximum value for RD is considered. In (9) and (10) either the maximum blocking (CA) is taken into account as well as the minimum guaranteed duration of the asynchronous windows (LAW_{UT}). Therefore, the average responsiveness to change requests is greater than the values that result from using the previous equations. These are nevertheless important since they establish lower bounds to the system responsiveness.

5. CONCLUSION.

This paper discusses the levels of responsiveness demanded from communication systems in dynamic environments. In particular, it focuses on the FTT-CAN protocol, which can efficiently handle periodic (synchronous) as well as aperiodic (asynchronous) messages. However, its planning-based operation imposes some limitations to the responsiveness to requests for changes in the synchronous message set. To analyse this problem, key parameters that have impact in the responsiveness of the synchronous messaging system (SMS) of the FTT-CAN protocol are presented and their influence is discussed, namely the plan duration and the instant in which the scheduler is started.

The proposed solution to improve the SMS responsiveness, without changing the plan length and/or the scheduler starting point, consists in using the asynchronous messaging system (AMS) to temporarily convey the changed message streams until they are taken into account by the SMS. Then, the synchronous transient response time (STRT), defined as the time lag that mediates between a change request and the instant at which the corresponding new requirements are reflected in the bus traffic, is substantially reduced. Such time lag is upper bounded by the period of the variable plus one EC. If the schedulability test is performed, its execution time adds to this bound.

Finally, two sufficient conditions are established which allow to define at pre-runtime the required minimum bandwidth for the AMS so that a given number of change requests, either simultaneous or separated by a minimum inter-arrival time,can be timely handled.

REFERENCES

Almeida, L (1999). Flexibility and Timeliness in Fieldbus-based Real-Time Systems. PhD Thesis. University of Aveiro-Portugal,

Almeida, L., J.A. Fonseca, P. Fonseca (1999). A Flexible Time-Triggered Communication System Based on the Controller Area Network: Experimental Results. *Proc. of FeT'99 (Int. Conf. on Fieldbus Technology).* Magdeburg, Germany.

Fohler, G. (1995). Joint Scheduling of Distributed Complex Periodic and Hard Aperiodic Tasks in Statically Scheduled Systems. *Proc. 16th Real-Time Systems Symposium.* Pisa, Italy.

Kopetz, H. (1997). *Real-Time Systems Design Principles for Distributed Embedded Applications.* Kluwer Academis Publishers.

Pedreiras, P. , L. Almeida (2000). Combining Event-triggered and Time-triggered traffic inFTT-CAN: Analysis of the Asynchronous Messaging System. *Proc. WFCS 2000*, Oporto.

Peraldi, M.A. and J.D. Decotignie (1995). Combining Real-Time Features of Local Area Networks FIP and CAN. *Proc. of ICC'95 (2nd Int. CAN Conference), CiA – CAN in Automation.*