REVISTA DO DETUA, VOL. 3, Nº2, OUTUBRO 2000

A Proposal to Improve the Responsiveness of FTT-CAN Synchronous Messaging System

P. Pedreiras, L. Almeida, J. A. Fonseca

***Resumo* – Um sistema de comunicações tempo-real flexível deve suportar modificações ao conjunto de mensagens que transporta. Estas modificações podem ser solicitadas por um operador humano, durante a configuração ou manutenção do sistema, ou autonomamente pelo sistema de controlo, quando responde a alterações no sistema controlado. No primeiro caso, um tempo de resposta até alguns segundos pode ser aceite. Todavia, no segundo caso, as propriedades físicas do sistema controlado podem exigir que as mudanças ao conjunto de mensagens sejam executadas num prazo mais curto. Em alguns casos poderá ser necessário um tempo de resposta da ordem de milisegundos.**

Neste artigo, analisamos a reactividade do protocolo FTT-CAN, em particular do seu sistema de mensagens síncronas. Em seguida apresentamos um método para melhorar essa reactividade face a pedidos de alterações com requisitos temporais urgentes.

***Abstract* - A flexible real-time communication system must support modifications to the message set which it conveys. These changes can be requested by a human operator, during system set-up or maintenance, or autonomously by the control system while responding to variations in the environment. In the former case, a response time up to a few seconds can be acceptable. However, in the latter, the physical properties of the environment can require that changes to the message set are carried out in a short term. In this case, a response time in the order of a few milliseconds can be demanded.**

In this paper we analyse the responsiveness of the FTT-CAN protocol, in particular of its synchronous messaging system. Then, a method is proposed to improve such responsiveness with regard to urgent requests for changes in the message set.

I. INTRODUCTION

A. Levels of system responsiveness

During normal operation, processes controlled by real-time computer systems experience phases of continuity as well as of changes [5]. 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 [1] 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. If a few hundreds of milliseconds are taken in this operation, the security of the driver is not compromised.

B. About this paper

The FTT-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 shows some guidelines that can be used to analyse the performance of that method. Finally, section 5 concludes the paper.

II. BRIEF PRESENTATION OF FTT-CAN

The FTT-CAN (Flexible Time-Triggered communication on CAN) protocol has been briefly presented in [3] and further developed in [4]. A feature that distinguishes this protocol from other proposals concerning time-triggered communication on CAN [2] 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.

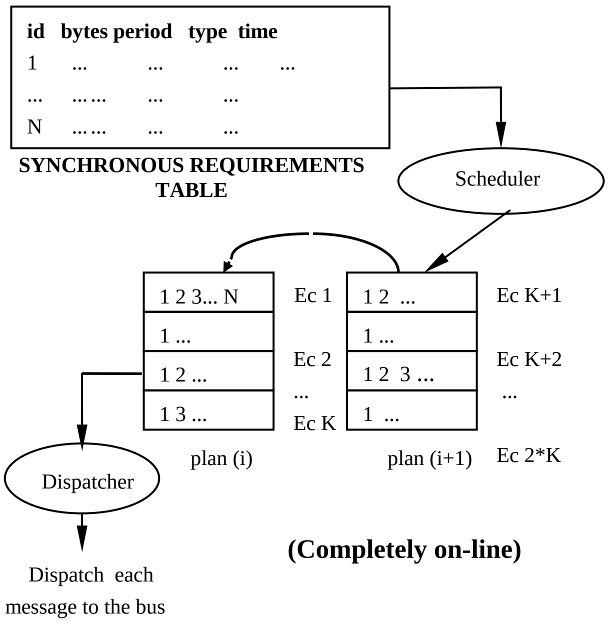

Figure 1. The planning scheduler

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 (fig. 1).

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. The latter is transmitted during the periods of the EC not used by the synchronous messages.

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).

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).

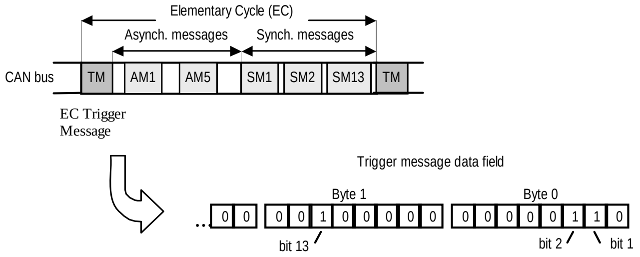

Figure 2. EC Trigger Message data contents.

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. 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. 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.

III. IMPROVING FTT-CAN RESPONSIVENESS

A. Flexibility limits

Once a change request is made concerning the current 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. If the change request is made just before the scheduler starts building the next plan (Fig. 3, marker A), the synchronous transient response time reaches its minimum value (one plan), since the scheduler uses the new requirements immediately. However, if the change request is made just after the scheduler start (Fig. 3, marker B), it will consider the request only in its next instance (one plan after), and the synchronous transient response time becomes two plans long. Thus, the synchronous transient response time when using the SMS alone varies between one and two plans ($LPlan<STRT_{SMS}<2*LPlan$).

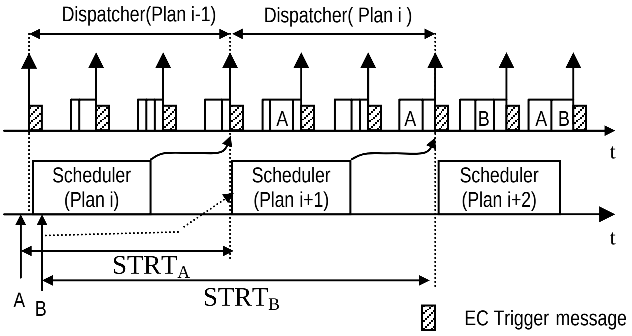

Figure 3. SMS Responsiveness bounds.

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 [3,4]. 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 [4], 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 [4], using this approach the synchronous transient response time can be bounded to the interval: $wcetSch < STRT_{SMS} < LPlan+wcetSch$, where LPlan stands for the plan duration.

B. 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. The proposed way of achieving this, without disturbing the other synchronous messages, 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) is transmitted using the asynchronous message area starting in the EC right after the request. As soon as the dispatcher starts processing the plan in which the new message parameters are reflected (plan i in the example above), the system resumes normal operation, that is, the message is included in the synchronous message area and 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).

If the change to the message set consists only in the addition of a new message, the process described above is adequate. 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) must be eliminated. Those instances still use to the older parameters (before the change) while the updated instances are transmitted in the asynchronous area. The elimination referred above is required to avoid replication of the message production in both synchronous and asynchronous systems.

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.

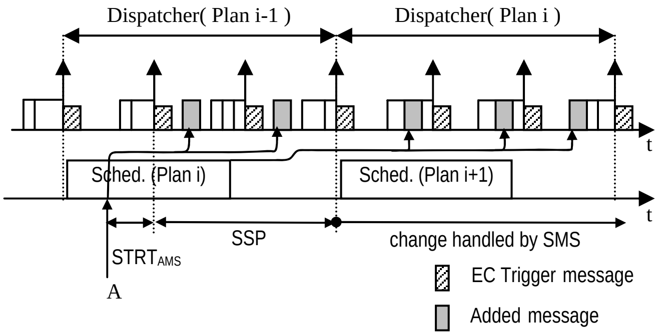

Figure 4. Using the AMS to temporarily convey a new synchronous message.

In fact, its worst case value occurs when the request for changes on an existing synchronous stream is done just after the EC trigger message has been sent. Notice that it would be difficult to eliminate an eventual instance of that stream in the current EC since its production was triggered by the data in the EC trigger message. Thus, $STRT_{AMS} < 1$ EC. Moreover, the following relationship can be established between the STRT with and without the AMS support:

$$STRT_{AMS} = STRT_{SMS} - SSP$$

C. Implementation issues

From the operational point of view, several steps must be performed in order to process the request for a change to the 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.

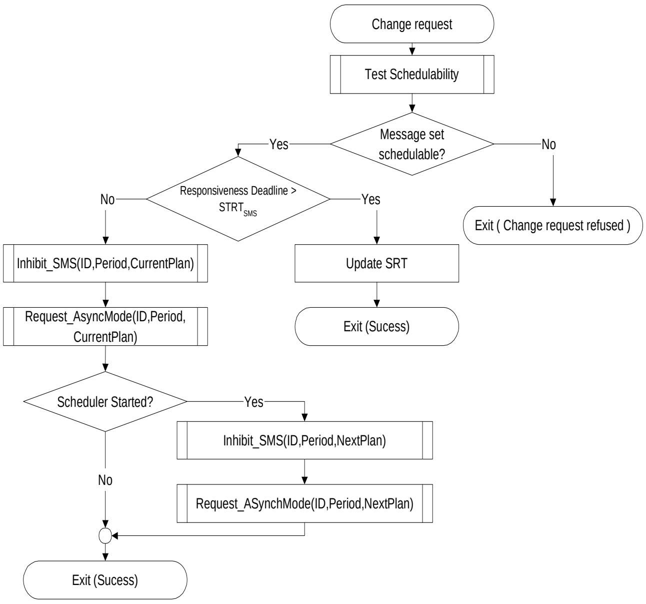

Figure 5. Operational diagram.

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, 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, the change is made in the SRTable. Otherwise a request is made to the dispatcher to eliminate the message from the synchronous message area and notify the producer station to start producing the message using the asynchronous system (AMS). If the scheduler has already started to build the next plan, two state variables are updated in order to allow the dispatcher to know at the beginning of each plan which messages will continue to be produced in the AMS and which will start to be produced by the SMS.

The start and end 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 (start/end). 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 two messages per change request.

IV. GUIDELINES TO ANALYSE SYSTEM PERFORMANCE

During the synchronous support period (SSP), the synchronous messages corresponding to a change request are handled by the AMS under a best effort policy, and will compete for the bus jointly with other asynchronous messages. Despite their higher priority (established by FTT-CAN), the synchronous messages may experience a bounded blocking caused by regular asynchronous ones.

To obtain an adequate behaviour from the AMS during the SSP, in order to timely handle the requests for changes to the synchronous message set, a minimum bandwidth must be allocated to that system. This minimum bandwidth depends on the maximum rate of change requests that the system is expected to handle. This rate determines the maximum number of synchronous and control messages that the AMS might be requested to transmit at any given time. With this maximum number of messages, existing analysis used to study the AMS response time to real-time sporadic messages [6] can be adapted to calculate the required minimum bandwidth. Then, such bandwidth can be allocated during system set-up, at configuration time, through a parameter that limits the maximum duration of the synchronous phase in each EC.

V. 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. Hence, 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 at which the scheduler is started.

Then a method is presented to improve the SMS responsiveness beyond that allowed by managing the plan length and/or the scheduler starting point. It consists on using the asynchronous messaging system (AMS) to temporarily convey the changed message streams until the changes 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 respective new requirements are reflected in the bus traffic, is substantially reduced. Its upper bound is given by one EC (Elementary Cycle) plus the schedulability test execution time (when performed).

Finally, some guidelines are referred that allow to obtain guarantees at pre-runtime concerning the rate of changes to the synchronous message set that can be timely supported. This topic is presently being studied.

REFERENCES

[1] Kopetz, H. Real-Time Systems Design Principles for Distributed Embedded Applications. Kluwer Academis Publishers, 1997

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

[3] Almeida, L., J.A. Fonseca, P. Fonseca. 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, September 1999.

[4] Almeida, L. Flexibility and Timeliness in Fieldbus-based Real-Time Systems. PhD Thesis, University of Aveiro, Portugal, November 1999.

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

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