Adapting the Sampling Period of a Real-time Adaptive Distributed Controller to the Bus Load

Ana Antunes
ESTSetúbal, Instituto Politécnico de Setúbal
Campus do IPS, Estefanilha
2914-508 Setúbal, Portugal
aantunes@est.ips.pt

Paulo Pedreiras, Alexandre Manuel Mota
LSE-IEETA/DET
Universidade de Aveiro
3810-193 Aveiro, Portugal
{pedreiras,alex}@det.ua.pt

Abstract

This paper presents a new method to allow more flexibility on the scheduling of messages across a fieldbus in overload situations. The method proposes the on-line adaptation of the sampling period to the bus load, trying to keep it as low as possible during overloads. The sampling period is allowed to change inside an interval corresponding to 4 to 10 samples per rise time. To asses the solution herein presented, a distributed adaptive control system was implemented in TrueTime, using the FTT-CAN protocol, for the communication infrastructure, a pole-placement controller and a model for the identification of plant parameters that takes into account the network-induced jitter.

1. Introduction

In the last years much work has been developed concerning the issue of real-time control and scheduling co-design. References [1]-[4] present some of that work. Distributed implementations may introduce additional jitter on the control loop due to the need to access the network for exchanging data between diverse system nodes, which, in turn, may lead to a deterioration of the control performance [2], [3]. Some techniques used to reduce the jitter impose heavy constraints to the scheduler, which becomes very rigid [2] and sporadic requests may not be schedulable due to the lack of flexibility imposed by the constraints of the control tasks. On the other hand the pressure to use the same infrastructure to support different control systems rises due to economic reasons [3], [4], a situation that potentially aggravates the network-induced jitter and encourages the use of resource-aware techniques.

Different approaches have been suggested to allow more flexibility. In [2] flexibility to the scheduling process is obtained by relaxing some timing constraints related to the control tasks. Different sets of sampling intervals and time delays are specified a priori, for the same control task, to be chosen from at run time according to the measured time delay. In [3] a feedback scheduler is presented. Here the control tasks inform the scheduler that they intend to do a mode change and the scheduler adjusts the periods of the tasks running to account for that mode change. The reported solutions refer to a set of control tasks executing on the same microprocessor.

This paper reports a distributed control system implemented over the CAN bus using the FTT-CAN protocol where the period of the control task is adjusted on-line in order to allow more flexibility to the bus scheduler.

2. System architecture

To keep the jitter predictable and bounded the network traffic is scheduled using the FTT-CAN protocol [1], [5]. This protocol allows flexibility, timeliness and an efficient combination of time and event triggered traffic. The time-triggered traffic is scheduled on-line and centrally, in a node called master, according to the instantaneous traffic requirements. The timeliness of the time-triggered traffic is guaranteed by means of an on-line admission control mechanism, which only commits change requests on the traffic attributes that do not jeopardize the system schedulability. When using the FTT-CAN protocol the bus time is slotted in Elementary Cycles (ECs) with fixed duration. The nodes are synchronized at the start of each EC by the reception of a message called EC Trigger Message (TM) sent by the master node. Each elementary cycle is divided in two consecutive windows one for the event-triggered and another for the time-triggered traffic. The TM contains information about the messages that are to be produced at each particular EC. Each node in the network has to decode the TM and produce the messages when ordered. The FTT-CAN protocol may include a Quality-of-Service (QoS) manager integrated with the on-line admission control [6]. In this case the message properties include a range of accepted QoS (specified e.g. as a period range) and the QoS manager dynamically grants the best possible QoS according to the overall bus load. This mechanism allows using more

0-7803-9402-X/05/$20.00 © 2005 IEEE

efficiently the bus bandwidth and handling overloads gracefully. The tests presented in the present paper use only time-triggered messages due to limitations of the simulator used.

Figure 1 shows the block diagram of the distributed system. The system is composed of 5 nodes. The master node implements the master functions defined by the FTT-CAN protocol. The sensor node samples the plant and sends the sampled value to the controller node. The controller receives the sampled value and generates the

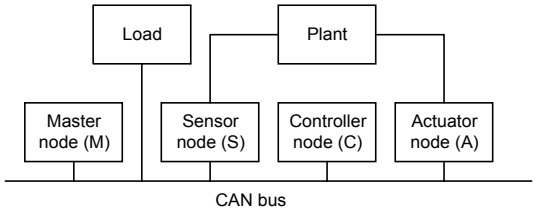


Figure 1. Block diagram of the distributed system.

actuation value that will be sent to the actuator node. The controller node is also responsible for the requests to change the sampling period according to the status of the bus as will be explained latter on. The actuator node receives the actuation value and acts upon the plant. The load node generates load on the bus.

The transfer function of the plant is given in (1).

$$\frac{Y(s)}{U(s)} = \frac{0.5}{s+0.5} \tag{1}$$

The system was simulated using TrueTime, a MATLAB/Simulink based simulator for real-time control systems [3], [7]-[9].

3. The distributed adaptive controller

The controller implemented is a pole-placement adaptive distributed controller. The adaptive controller was implemented using MATLAB inside the TrueTime Kernel of the controller node.

The system identification is based on a model that takes into account the jitter by modelling it as fractional dead-time [10],[11]. This identification model allows a better control performance under jitter conditions as the sampling period approaches the upper bound of the interval defined by the rule of thumb given in [12], when compared with the ordinary model. These results are reported in [13] and [14] were further details on the controller are presented.

The proposed model for the SISO (Single Input Single Output) system is presented in equations (2) and (3) where τ represents the variable delay that can be considered as a dead-time.

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t-\tau) \tag{2}$$

$$y(t) = x(t) \tag{3}$$

For a system of 1st order the discrete transfer function for the model with fractional dead-time is given by (4).

$$G(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 - a_1 q^{-1}} \tag{4}$$

The discrete function for the model was obtained using the parametric model ARX (Auto-Regressive with an eXogenous signal) [15]. The system parameters were estimated using the least squares criterion [15] and a recursive implementation using forgetting factor [15] was adopted to run on-line during the simulation. The sampling period is allowed to change between 0.12s and 0.28s in discrete steps of 20ms.

The regressors for the model are shown in (5).

$$y(k) = \begin{bmatrix} y(k-1) & u(k-1) & u(k-2) \end{bmatrix} \tag{5}$$

The control function uses the pole-placement technique. It allows for the closed-loop response of the system to be totally specified in advance. The closed-loop pole was chosen as α_m=2 and the observer pole as α_o=4. The parameters of the control function were obtained by solving directly the Diophantine's equation. The resulting control function is given in (6).

$$u(k) = t(r_{ef}(k) - a_{obs} r_{ef}(k-1)) - s_0 y(k) - s_1 y(k-1) \ldots$$
$$-(r_1 - 1)u(k-1) + r_2 u(k-2) \tag{6}$$

The control function parameters are computed on-line during the simulation taking into account the changes in the sampling period.

4. Sampling period adaptation algorithm

The sampling period adaptation algorithm is responsible for the adaptation of the sampling period (h) to the network conditions. The sampling period can change inside an interval defined accordingly to the rule of thumb presented in [12], stating that for first order systems the number of samples per rise time (N_r) should be between 4 and 10. The use of the FTT-CAN protocol implies that the minimum time unit, for period specification, is equal to the EC time. The changes introduced in the sampling period at each control cycle are equal to ± one EC time, according to the network status.

It is well known that the control performance is better for shorter sampling periods [3]. Taking that into account, the sampling period adaptation algorithm tries to keep the sampling period as short as possible. The master informs all nodes in the network about the status of the network. It can be "normal" or "overload". The "overload" status can be raised when the master receives a request to schedule a message that it cannot support, since the FTT-CAN protocol always guarantees the schedulability and timeliness of the messages accepted to be scheduled.

When the network status is "overload" the controller requests for the rise of its sampling period in order to allow more flexibility to the scheduler. When the network status returns to "normal" the controller tries to decrease its sampling period to get better control performance. Some precautions should be taken in this last procedure in order to prevent the network to become "overload" due to the changes imposed by the control loop itself. This can be done by allowing the changes to take place only after a certain number of control cycles after the return to the "normal" status of the network. On the other hand the stabilization of the control parameters between changes also leads to the need to impose a certain period of time between consecutive changes of h.

The requests for changing the sampling period are analysed in the master node in order to guarantee that the new message set is still schedulable (on-line admission control, section 2). Upon acceptance from the master the control loop messages' period is changed and the controller node is notified of the change.

The sampling period adaptation algorithm is presented in figure 2.

```
if ovl
  if h<hmax
    h_next = h+ec_time;
  end
else if (h>hmin) && (tmin)
        h_next = h-ec_time;
end
```

Figure 2. Sampling period adaptation algorithm.

5. Experimental results

The system was implemented using the FTT-CAN protocol on the TrueTime simulator. The master node runs the scheduler, the dispatcher and the network handler task. The messages from the control loop to be scheduled are: one messages from the sensor node with the sampled value, two messages from the controller node, one with the actuation value and another with the request to change the sampling period and one message from the master communicating the acceptance of the new sampling period.

At the beginning of each the control cycle the master issues a TM asking for the sensor node to produce the sampled value and to the controller node to produce the request for the sampling period change. The controller receives the sampled value and produces the actuation value. The controller node runs the sampling period adaptation algorithm to calculate the sampling period to be requested to the master for the next control cycle before generating the request message. This calculus is based on the information provided by the master, in each TM, about the status of the network ("normal" or "overload"). Upon acceptance of the request the scheduler will ask for the production of the messages based on a different sampling period and will inform the controller that the new sampling period was accepted.

The admission control algorithm in the master node is not implemented yet and as a result the status of the network is provided by an external sequence that can be changed to test different situations.

For the closed-loop behaviour chosen the limits determined by the application of the rule of thumb [12] are 0.11s when N_r=10 and 0.28s when N_r=4. The EC time is equal to 20ms. The initial sampling period is set to be 0.12s and the maximum value is 0.28s.

The initial table provided to the master node with the message set characteristics is shown in Table 1. Apart from the control loop messages with identifiers 20, 35, 40 and 45 there are 6 load messages with higher priority.

Msg	Local_ID	Period	Size	Offset	Deadline
Load	5	1	3	0	1
Load	9	2	5	0	2
Load	10	4	5	0	4
Load	13	5	4	0	5
Load	14	8	7	0	8
Load	16	8	1	0	8
Req. change h	20	6	2	0	6
Accept h	35	6	2	0	6
Sample	40	6	2	0	6
Actuation	45	6	2	1	7

Table 1. Message set characteristics.

Three tests were made using the system described. In tests 1 and 2 the sampling period is constant and equal to 0.12s and 0.28s, respectively. In test 3 the sampling period is initially set to 0.12s and updated on-line according to the network status. The network status is "overload" from t=9s until t=45s and normal in the remaining instants.

Figure 3 presents the control signals for test 3. The sampling period is shown multiplied by 10 for greater detail. Figure 4 presents the sampling to actuation jitter during test 3.

The criterion used to compare the control performance was the ISE (Integral of the Square Error) criterion. The values obtained for the ISE are shown in Table 2. The ISE was computed from t=4.8s to t=85s. The total bus load for each test is also presented.

Test nº	h	ISE	Bus Load
1	0.12s	3.73	70%
2	0.28s	27.25	66.7%
3	adapting	3.89	69.2%

Table 2. ISE report.

The results show that on-line adaptation of the sampling period is possible without loss of the control action and the control performance is only slightly worst than the one obtained for h constant and equal to 0.12s while allowing greater flexibility to the scheduler during overload conditions. The result is better than the one obtained with h constant and equal to 0.28s.

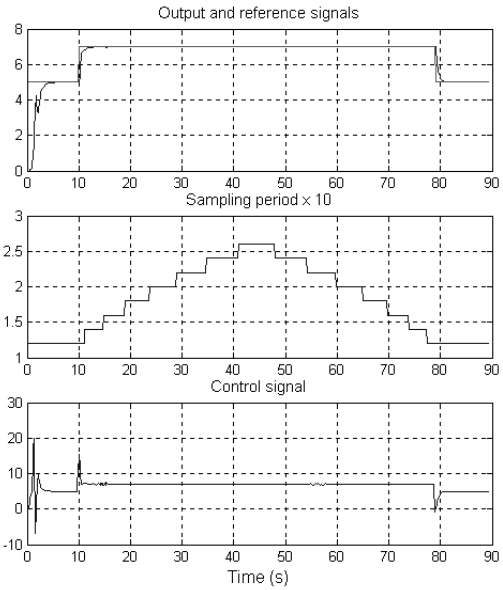


Figure 3. Control signals for test 3.

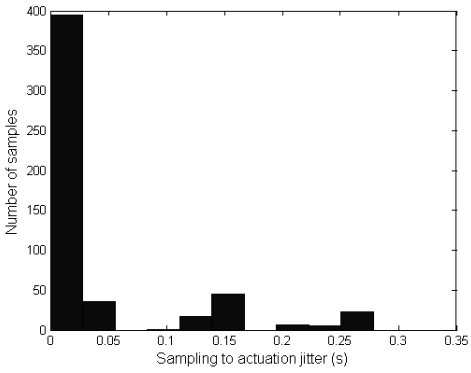


Figure 4. Sampling to actuation jitter for test 3.

As expected the total bus load is reduced in approximately 3% from test 1 to test 3.

6. Conclusions

This paper presents a distributed real-time adaptive control system using a pole-placement controller where the sampling period is adapted on-line to account for possible overload situations in the network. The results show that there is only a small loss in the control performance when comparing to the use of h constant and equal to the smaller value recommended while allowing at the same time more flexibility to the scheduler.

Future work will include the study of strategies to deal with the problems created by the change of h in the transitions between consecutive schedules and analyse the impact of the choice of the EC time in the control performance with adaptation. The on-line admission protocol will also be implemented in the master node.

References

[1] Almeida, L, "Flexibilty and timeliness in fieldbus-based real-time systems", Ph.D. dissertation, Dept. Electrónica e Telecomunicações, Universidade de Aveiro, 1999.

[2] Colom, P.M., "Analysis and design of real-time control systems with varying control timing constraints", Ph.D. dissertation, Universitat Politecnica de Catalunya, 2002.

[3] Cervin, A., "Integrated control and real-time scheduling", Ph.D. dissertation, Lund Institute of Technology, 2003.

[4] Arzen, K.-E., Cervin, A. and Henriksson, D, "Implementation-aware embedded control systems", in D. Hristu-Varsakllis and W.S. Levine (Ed.): Handbook of networked and embedded control systems, 2005. To appear.

[5] Almeida, L., Pedreiras, P. and Fonseca, A., "FTTCAN protocol: why and how", IEEE Transactions on Industrial Electronics, vol.49, nº. 6, 2002.

[6] Pedreiras, P., Almeida, L., "The Flexible Time-Tigerred paradigm: an approach to QoS management in distributed real-time systems", Workshop on Parallel and Distributed Real-Time Systems (WPDRTS), Nice, France, 2003.

[7] Cervin, A., Henriksson, D., Lincoln, B. and Årzén, K.-E., "Jitterbug and TrueTime: analysis tools for real-time control systems", in *Proc. 2nd Workshop on Real-Time Tools*, Copenhagen, Denmark, 2002.

[8] Henriksson, D., Cervin, A. and Årzén, K.-E., " TrueTime: simulation of control loops under shared computer resources", in *Proc. 15th IFAC World Congress*, Barcelona, Spain, 2002.

[9] Henriksson, D. and Cervin, A., "TrueTime 1.2- Reference Manual", Dept. of Automatic Control, Lund Institute of Technology, Sweden, 2004.

[10] Mota, A.M. and Fonseca, J.A., "Dealing with jitter in systems modeling and identification", in *Controlo 2000*, Guimarães, Portugal, 2000.

[11] Mota, A.M., Fonseca, P.M. and Fonseca, J.A., "Systems modeling and identification in CAN based distributed control systems", in *Proc. IFAC Distributed Computer Control Systems*, Sydney, Australia, 2000.

[12] K. J. Åström and B. Wittenmark, *Computer Controlled Systems*, Prentice Hall, 1997.

[13] Antunes, A., Dias, F.M. and Mota, A.M., "Influence of the sampling period in the performance of a real-time adaptive distributed system under jitter conditions", *WSEAS Transactions on Communications*, vol.3, pp. 248-253, 2004.

[14] Antunes, A., Mota, A.M., "Control Performance of a Real-time Adaptive Distributed Control System under Jitter Conditions", Proc. Conf. Control 2004, Bath, UK, 2004.

[15] L. Ljung, *System Identfication- Theory for the user*, 3rd ed., Prentice Hall, 1987.