Dealing with jitter in Systems Modelling and Identification

Alexandre Manuel Mota, José Alberto Fonseca

Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, 3800 Aveiro, Portugal

Email: alex@det.ua.pt, jaf@det.ua.pt

Abstract: Distributed control systems performance can be affected by the occurrence of jitter in the messages that carry relevant data such as sample and actuation variables. This jitter comes from the influence of messages from other sources and thus depends on factors such as the distribution of controller tasks and the medium access control used in the network. In this paper a study of the influence of jitter in system identification is presented. The results are derived in an adverse situation when both sampling and actuation data suffer from jitter. It is shown that using a model that takes in consideration a fractional dead-time with a value equal to the jitter average leads to a much better parameter identification than when the problem is just ignored. - Copyright CONTROLO 2000.

Keywords: Distributed Control Systems, Jitter, System Modelling, Identification.

1. INTRODUCTION

Presently, distributed systems find wide dissemination in embedded control applications, particularly in real-time systems for the automotive and robotics fields. Most of them rely on a fieldbus [1] to interconnect a set of nodes. When periodic variables, such as the ones used in control applications, are to be transmitted, it is possible to impose an average transmission period but, due to the interaction of other periodic, sporadic or aperiodic traffic, it is rather difficult to obtain constant time intervals between successive instances of the same periodic variable. The variation of the variable's periods due to MAC (medium access control) is often called network-induced jitter and may have a negative impact in control loops [2]. Recently, [3] revisits the subject of the degradation of controller performance due to jitter in the sampled and in the actuation variables and [4] studies the problem when a CAN – Controller Area Network is concerned. The need for further research in communication jitter minimisation is also referred in [5]. This subject is also under investigation in the general real-time systems field, e.g. in [6] and in [7] where changes in periods of control tasks are considered.

In this paper the influence of this type of jitter in control actions is discussed and some simulation results are presented.

The occurrence of jitter depends on the architecture of the distributed control systems. This topic is then analysed in paragraph 2. After, in paragraph 3, a solution to model jitter in the sampling signal and in the controller output is presented. Also, it is shown that, for system identification, there is the possibility of reflecting the influence of jitter using a model with fractional dead-time instead of ignoring the problem.

Simulation results with the comparison of the performance of these two types of models show the advantage of the first approach. Finally, in paragraph 4, some conclusions are presented as well as going research work following the same line.

2. IMPLEMENTING DISTRIBUTED CONTROL SYSTEMS

Distributed control systems follow most of the times a generic architecture such as the one depicted in figure 1. There, it is shown just one of the several control loops that may share the same communication infrastructure. This example of control loop includes five blocks: the controller, the sensor and the actuator, which are mandatory but may share the same node (if all share the same node then the system becomes centralised) and the systems identification and the monitoring/supervision node which are optional.

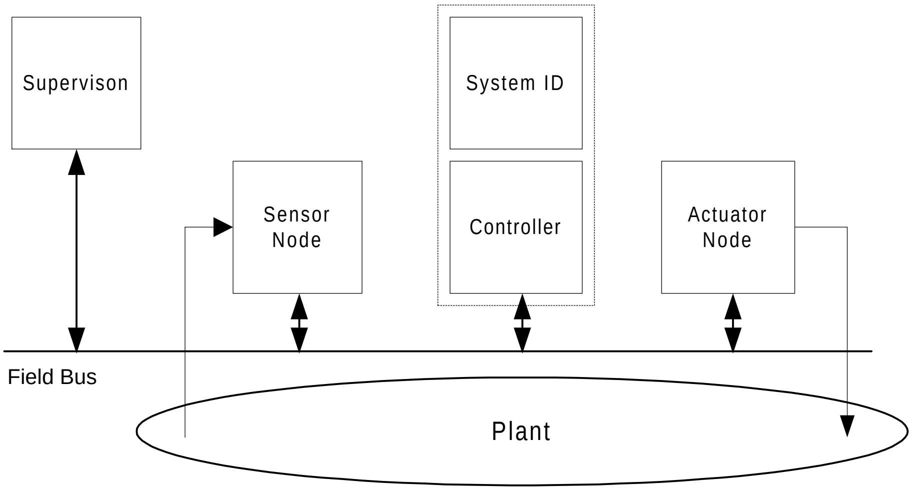

Figure 1 – Generic architecture of a distributed control system.

It should be noticed that each of the nodes integrates a set of subsystems. For example the sensor node includes the sensor itself, an A/D interface and/or input port, a network interface and, almost always, a CPU, often of the microcontroller type. Other nodes such as the controller and actuator will also follow the same architecture. It is obvious that the supervision/monitoring node is most probably based on a PC or PLC and not in an embedded CPU. But in this case it will also operate with other control loops that share the same system.

From now on, the blocks of figure 1 are considered independent, as, for the purposes of this work, this is almost the worst case situation. In fact, this leads at least to the transmission of messages from the sensor to the controller, and from this one to the actuator. These messages will carry the periodic variables used in the controlling process. The system identification block shares a node with the controller, otherwise these two functions would also generate messages between then.

The production of the variables’ values and the transmission of messages on the fieldbus are in general not synchronised. Most of the times, at the sensor node there is a timer generating the sampling instants and, when the data is obtained, the transmission services are called in order to transmit it. Tasks operating in other nodes that need this data, for instance the controller, must wait to its arrival in order to proceed executing. When they produce results, e.g. the actuation variable, these are requested to be transmitted as soon as they are produced. From the transmission point of view this type of operation is called event-triggered because transmission is requested in the sequence of an event not synchronised with the global system operation. Event-triggered operation is the most common approach in current distributed control systems.

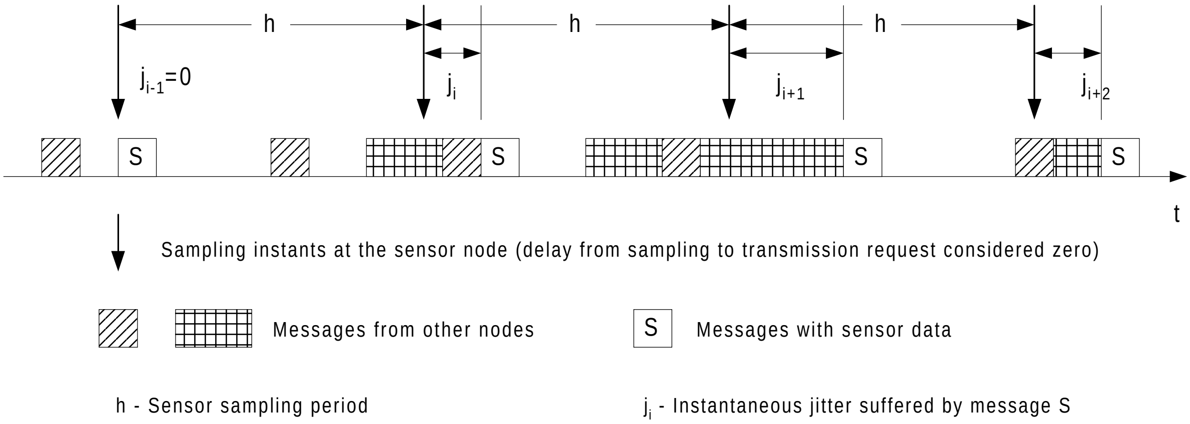

Figure 2 – Network-Induced jitter.

When several control loops and, often, other monitoring or actuating systems are included in the same fieldbus-based distributed system (the word fieldbus is here used in a broad sense of network with appropriate characteristics to handle harsh environments and minimal timing constraints), transmission requests at the node level may not immediately lead to the transmission of the message over the network. The sharing of the network resource is somehow controlled (this is done at the MAC – medium access control level of the OSI Data Link Layer) and thus, when there are different nodes requesting transmission, an arbitration process will decide which one will do it first.

The transmission of a periodic message carrying, e.g., sensor data acquired in specified instants with a constant sampling interval, may then be delayed by influence of a message being transmitted and of other messages waiting for the availability of the bus when these have higher priority from the MAC point of view. Figure 2 shows an example of the effect of unidentified messages in the message with the sensor data. The message is then received with an average periodicity that is, in principle, constant and equal to the initial period. However, the time interval between two consecutive instances of the message changes. This variation in the period is often called network-induced jitter.

In a distributed system, factors such as the Medium Access Control used, the priority of the messages, the message scheduling algorithm (when the definition of the access to the medium is fixed before operation), initial phasing, affect the jitter value. In some cases there are studies already available. This is the case of CAN – Controller Area Network. In [8] it is shown that the delay suffered by a periodic message due to the influence of others and of the medium access arbitration can be modelled by a gamma distribution (figure 3). In this case the message priority will affect the average jitter as well as its dispersion.

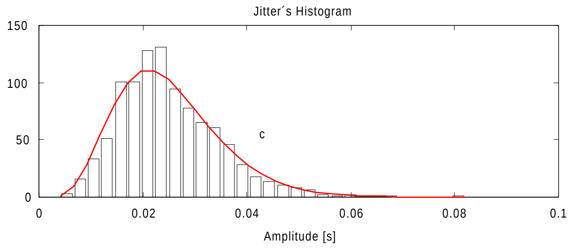

Figure 3 –Jitter Distribution.

3. DEALING WITH JITTER IN SYSTEMS MODELLING AND IDENTIFICATION

The influence of jitter in control systems developed on distributed systems has only received a very scarce attention till now. In our opinion this is due to the fact that the implementation of true distributed control systems is only in its infancy. There is then an extensive field of studies that must be carried on, addressing, on one side, the influence in systems modelling and identification and, on the other side, in the controller performance. Also model-based or model-free controllers will certainly be differently affected by jitter. In this paper the focus is put on a preliminary reasoning on the effect of jitter in modelling and identification processes.

As the concern is on discrete systems, the effect of jitter can be viewed as a perturbation that introduces a variable delay in the reading of the samples and in the actuation signal sent to the plant. The existence of one or both of the situations depends on the distributed control system architecture. These situations are often named read-in and read-out jitter, respectively [3].

In figure 4 a block diagram of a system with read-out jitter is presented. There, it is shown that the discrete signal *u(k)* is the input of a Zero-order-Hold (ZOH) which generates the continuous signal *u(t)*. This one suffers a delay, τ, before it is output to the plant. This delay is variable and thus, jitter will be present in the *u(t- τ)* signal.

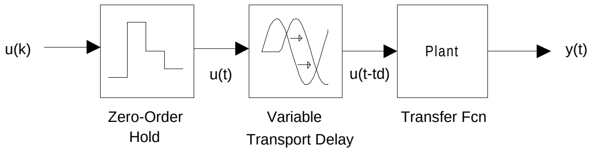

Figure 4 – Diagram of a system with read-out jitter.

Figure 5 now represents a block diagram of a system with read-in jitter. In this case the output of the system, *y(t)*, is delayed by an interval called τ, before the input of the ZOH. Again, this delay is variable and can then be considered jitter.

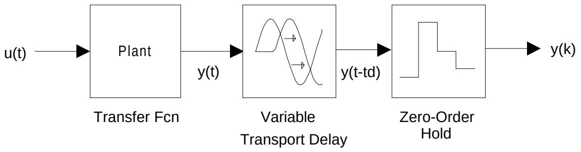

Figure 5 – Diagram of a system with read-in jitter.

Two situations can then be considered when one proceeds to the identification of a system with jitter. The first one is just ignoring it. The second is to try to take it in consideration in the system model. In this last case, a possible model for a SISO system could be:

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

where τ is the variable delay which can obviously be considered as a dead time. As in most distributed systems it is possible to bound this dead time and if this bound is the sampling period value ($h > \tau$), then the discrete model can be [9]:

$$y(kh+h) = \Phi y(kh) + \Gamma_0 u(kh) + \Gamma_1 u(kh-h)$$

where:

$$\Phi = e^{Ah}$$

$$\Gamma_0 = \int_0^{h-\tau} e^{As} ds B$$

$$\Gamma_1 = e^{A(h-\tau)} \int_0^{\tau} e^{As} ds B$$

The discrete transfer function is:

$$G(q) = [1 \quad 0](qI - \Phi)^{-1}(\Gamma_0 + \Gamma_1 q^{-1})$$

This last equation represents the discrete model of a system with fractional dead time which has now a new zero that doesn't exist when τ is zero. It is also clear that Γ_0 and Γ_1 values are dependent on τ.

One hypothesis for using this model is to assume a constant fractional dead time with a value equal to the average jitter. This can be a realistic situation at least for certain networks where time response is a concern (e.g. WorldFIP [10]).

In order to study the validity of this assumption, a simulation using four common control systems $\left(G(s) = \frac{1}{s+1}, \frac{1}{s}, \frac{1}{s^2} \text{ and } \frac{1}{s(s+1)} \right)$ was carried on. For each of them, a system identification with a non-recursive algorithm based on least squares method was done using either the simple model and the one with fractional dead-time. In the simulation the systems were affected by both read-in and read-out jitter with an average around 10% of the sampling interval h.

Figure 6 shows the results of 250 identification tests for the systems. There, it is shown the summation of the square errors between the outputs of the actual system and of the two identified models. It is clear that, when jitter is not taken into account (Model 1), the model identification is poor when compared with the one (Model 2) that considers it as a fractional dead time. The error in the first situation can be at least one order of magnitude greater than in the second one and even much more when the systems show a pole at the origin.

4. CONCLUSIONS AND FUTURE WORK

In this paper the problem of network induced jitter was presented and a preliminary study on its influence in systems identification was briefly discussed. The first results come from the simulation of four simple and typical control systems subject to jitter at the sampling data and at the actuation variable levels. These results show that the use of a discrete model considering a fractional dead-time to represent jitter lead to a much more accurate system identification than if jitter influence is ignored. This is only valid for a non-recursive identification algorithm based on least squares.

To go on with this research line, the influence of different jitter patterns is being considered, namely a wider dispersion, other distributions and average values close to the sampling period. Also, other identification algorithms, e.g. of the recursive type are being verified. Finally, the influence of jitter in model-free control algorithms is also under investigation. It should be also referred that this research work focussed on control issues can be used to tailor the distributed systems infrastructure (architecture, communication model, synchronisation solutions, message and task scheduling, etc.) in order to obtain a better performance in control applications.

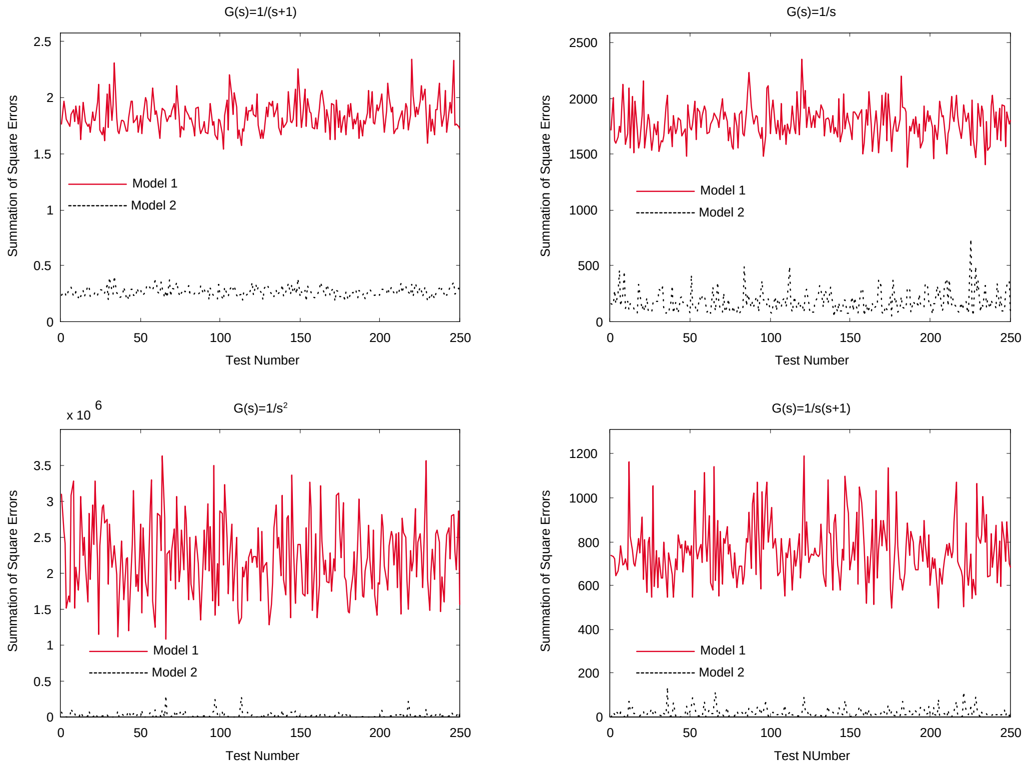

Figure 6 - Results.

REFERENCES

[1] Thomesse , "A Review of the Fieldbuses", Annual Reviews in Control, 22 pp. 35-45, 1998.

[2] Hong, S., "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems", IEEE Trans. Control Syst. Techn., Vol. 3, Nº 2, June 1995.

[3] Stothert, etal "Effect of Timing Jitter on Distributed Computer Control System Performance", Proc. 15 IFAC Workshop DCCS'98 – Distrib. Comp. Control Syst., Sept. 1998.

[4] Juanole, "Modélis. Éval. Protocol MAC du Réseau CAN", École ETR'99, France, Sept. 1999.

[5] Decotignie, "Future Directions in Fieldbus Research and Development", Proc. FeT '99 - Fieldbus Syst. and Appl. Conf., Germ., Sept. 1999.

[6] Cervin, "Improved Scheduling of Control Tasks", Proc. 11th Euromicro Conf. Real Time Systems, June 1999.

[7] Shin, etal "Adaptation and Graceful Degradation of Control System Performance by task Reallocation and Period Adjustment", Proc. 11th Eurom. Conf. Real Time Syst., June 1999.

[8] Fonseca, P. – Modélisation et validation des algorithmes non-déterministes de syncronisation des horloges, Phd Thesis, Universidade de Aveiro, 1999.

[9] Karl Astrom & Bjorn Wittenmark, "Computer Controlled Systems: Theory and Design, Prentice-Hall, 1990.

[10] P. Leterrier, "The FIP Protocol", WorldFIP Europe, France, 1992.