Variable delays and message losses: Influence on the reliability of a control loop

Rony Ghostine, Jean-Marc Thiriet, Jean-François Aubry

Variable delays and message losses: influence on the reliability of a control loop

Rony GHOSTINE[1], Jean-Marc THIRIET[2], Jean-François AUBRY[1]

[1]CRAN - Centre de Recherche en Automatique de Nancy (CRAN UMR CNRS 7039)

ENSEM

2, avenue de la Forêt de Haye

54516 Vandoeuvre lès Nancy

Tel: +33 (0)3 83 59 55 78, Fax: +33 (0)3 83 59 55 65

e-mail: Jean-Francois.Aubry@isi.u-nancy.fr

[2]GIPSA-Lab, Control Systems Dept. (UMR CNRS 5216)

BP 46

38 402 Saint Martin d'Hères cedex, France

Tel: +33 (0)4 76 82 64 14, Fax +33 (0)4 76 82 63 88

Corresponding author: Jean-Marc THIRIET[2]

(jean-marc.thiriet@ujf-grenoble.fr)

Abstract: Today, new technologies (distributed systems, networks communication) are more and more integrated for applications needing to fit real-time and critical constraints. It means that we require more and more to integrate these new technology-based components in systems or sub-systems dedicated to safety or dealing with a high level of criticality.

Control systems are generally evaluated as a function of required performances (overshoot, rising time, response time) under the condition to respect a stability condition. Reliability evaluation of such systems is not trivial, because generally classical methods do not take into account time and dynamic properties which are the bases of control systems.

The methodology proposed in this paper deals with an approach for the dependability evaluation of control systems, based on Monte-Carlo simulation, giving a contribution to the integration of automatic control and dependability constraints.

Keywords: control dependability - reliability – control systems – communication delays – message losses – transient failures – Monte-Carlo simulation

1 Introduction

New technologies, in particular digital microcontrollers and communication networks [1], are now widespread, including strong real-time-constraint or critical systems (ex: X-by-wire systems in vehicles). One of the questions which arise is how it is possible or not to use a new-technology based architecture in a safe or critical system [2]. What is the level of risk of a steering-by-wire car, for instance? In order to answer to this important question, some methodologies or approaches should be used.

How to quantitatively evaluate control system is still an open problem. There have been several investigations in control community; generally control systems are treated as static systems without considering their dynamic aspect [3]. Faults are generally considered as permanent, and system failure can be defined according to the failures of the components taking into account the system architecture. In network-oriented approaches, the failure is generally defined as the non respect of the time of delivery of a message (Quality of service) [4].

Two difficulties can be identified:

- Faults affecting the network may have various effects on the control system, depending on the state of the system, when the failure occurs. For example in a closed loop distributed around a network lost messages in the transient state does not have the same

effect as in the steady state [5] [6]. Another important aspect is the fact that errors can be transient, even fugitive; it is the case with digital-based systems and communication networks. It is consequently pertinent to consider and study the influence of the Quality of Service of the communication network on the system quality control [7].

- In researches dedicated to delayed systems [8], the behaviour of the system is considered generally in the presence of a constant (or bounded) delay in the loop (study of robustness, stability [9]…). In industrial reality, the disturbances of the network are random: the elements connected to the network function in an asynchronous way and several exchanges of messages are necessary during the same sampling time. The consequences are that in the general case an analytical study of the dependability is not easily realizable.

The capacity of the control devices to compensate some consequences of component failures needs to redefine the concept of failures of the whole system: the dependability [10] [11] [12] of the whole system is dependent not only on components of the system and their architecture, but also on the evolving states of the various components and the architecture (dynamic effect). It is so very difficult to get an analytical formal evaluation. To overcome these difficulties, an approach based on simulation is proposed.

Despite of its disadvantages (calculation time [13], convergence of the results), the Monte Carlo approach is used for the statistical evaluation of the reliability parameters thanks to many simulations (or stories), and makes it possible to obtain results.

The paper focuses on the incidence of two types of fugitive failures which can be encountered when using a network, either because of outside disturbances or due to the strategy of the protocol used: the loss of a sample and the delay of a sample. The combined influence of both types of failures is then studied, highlighting the cumulative effects. Indeed

the failures injected independently can have an acceptable effect whereas their simultaneous occurrence can lead to an unacceptable consequence.

The second section of the papers deals with the modelling environment used and the models developed. The third section discusses the influence of the faults, the criteria used, and focuses on the influence of the variable delays and the messages losses, first separately and then combined. The fourth section proposes a frame for the evaluation of the dependability of controlled systems, based on overshoot, response time (not developed in this paper) and stability criteria.

2 Modelling of the control system for dependability evaluation

2.1 Modelling environment

Modelling is based on SANs (Stochastic Activity Networks). SANs [14] are a stochastic generalization of Petri nets. These models permit the representation of concurrency timeliness, fault-tolerance and degradable performance in a single model. Structurally, they consist in activities, places, input gates, and output gates. Activities which are similar to transition in standard Petri nets, are of two types: timed and instantaneous. Timed activities represent activities of the modelled system whose duration impact the performance of the modelled system. Instantaneous activities, on the other hand, represent system activities which occur immediately. Input gates and output gates control the enabling of activities and define the marking changes that will occur when an activity completes.

SAN is defined with the express purpose of facilitating unified Performance/dependability evaluation as well as more traditional performance and dependability evaluation [15]. Dependability evaluation is performed by defining a set of measures in the model. In the context of SPN, these measures are derived from the concept of reward [16].

Our models are developed using Möbius tools [17], a tool that supports the use of SAN. Without entering deeply into details but in order to ease the following parts, some basic explanations about Möbius are now given. Any study in Möbius begins with basic models called atomic models. These atomic models can be joined to define more complex models.

Each atomic model in Möbius is made of places, activities, and gates. These components create the static part of a SAN, the tokens forming the dynamic part. Each component has various attributes (figure 1).

Places: can be simple or extended. In *Möbius,* the concept of extended place is used to represent the coloured places. Each simple place has a name and an initial marking; it is graphically represented as a blue circle. An extended place has a name and a structure: the structure will be the equivalent of the colour. A structure can be formed by the Cartesian product of various fields:

$$\text{Structure} = \{\text{int} \times \text{short} \times \text{double} \times \text{char} \times \text{float}\}$$

Example of a structure frame= {ID (int), value (float), size (int)}: it means that the source whose address *ID* (integer) sends a message on the network; the information sent has the value *value* (float); *size* (integer) is the number of bits of the message.

Activities: an activity can be of two types, instantaneous or timed. An instantaneous activity has a name and a number of cases. If the number of cases is greater than one, a probability is associated to each case. The sum of the probabilities of all the cases must be equal to one. A timed activity has all the attributes of an instantaneous activity plus a field dedicated to the time of activation of the activity. The time is stochastic and can be

defined according to several probability laws (exponential, normal, binomial…). Time is associated to the activities, this corresponds exactly to T-timed Petri nets in [18].

Input gates: Input gates are represented graphically by a red triangle whose top is oriented towards the input place and the base towards the activity. An input gate connects one or more places to only one activity. An input gate has a name and two functions: the validation function which defines the conditions of activation of an activity and a gate function which makes it possible to modify the marking of the input places once the activity has been crossed.

Output gates: Output gates are represented by a black triangle whose the base is oriented towards the activity and the top towards the output place. An output gate has a name and a gate function which makes it possible to modify the marking of the places after the crossing of the activity.

Arcs: Arcs connect input places to input gates, input gates to activities, activities to output gates and output gates to output places.

[Fig. 1. Atomic model under Möbius]

2.2 Modelling of the components of a control system

In digital systems, control algorithms are implemented as programs and data flow distributed on architecture of calculators and networks managed by protocols. Control systems we consider are, by nature, continuous, whereas their control is computerized and thus digital. The knowledge of the process output variables by the control system (system composed of the controller, the actuator and the sensor) is achieved only at specific moments, generally synchronized by a clock, this kind of system are called sampled systems and the

theory is well established. The sampling of a system consists in proceeding periodically to the acquisition of the process outputs and the corresponding update of its input variables. The sensor is the element in charge of the acquisition of the output of the process; this information is sent to the controller which decides the action to achieve (control) to change the states of the process. The controller does not act in general directly on the process, but through an actuator (figure 2).

[Fig. 2. Controlled system]

The network ensures the communication between the various components. The use of a network as well as shared resources (calculators carrying out several tasks) made that the acquisition times of the state of the controlled system and the update of the controls may be not really periodic. The following part shows how one can model such a system to simulate the behaviour, by taking account of the variable delays. In this work, the network mechanisms are not modelled, it is considered that the network is the source of random disturbances, for the Control and the Measurement (see figure 2). The models used here are those of the sensor, the controller, the actuator (the model is not proposed in this paper but is available in [19]), and the process.

2.2.1 Sensor

The sensor role is to send an updated image of the state of the process to the controller. The sensor is placed somewhere on the process to make it possible to take needed measurements. The sensor is a periodical component, synchronised by a clock. At each period (sampling period T_s), the sensor measures information concerning the process and prepares a frame to be sent to the controller through the network. The preparation of the data and the sending of information are done within the T_s time, constant and small enough to be generally considered as being negligible.

The periodicity of the sensor is ensured by the timed activity *period* (Figure 3), it periodically adds all T_s time units a token in *Place1*. The presence of a token in *Place1* validates the *sensor* instantaneous activity. The *sensor* activity will take an image of the state of the system and puts it in the place *sensor_output*. Times of measurement and coding of information are supposed to be very short, for that the *sensor* activity is instantaneous. This activity can be replaced by a timed activity with a fixed time if one wants to take into account a small delay. The extended place *sensor_output* contains a structure of the type frame, this place will be shared with the model representing the network. The presence of a token in this place means that a message is ready to be sent through the network.

The shared place *process* represents the state of the system. The equations describing the evolution of the process are detailed in the part 2.2.4 dedicated to the process.

[Fig. 3. Model of a sensor]

2.2.2 Controller

Controllers are used to control the system in the best way, following performance criteria. Classical performance criteria [20], [21] are:

- precision, materialized for example by a maximum value of the static error,
- speed, materialized in general by a maximum value of the rising time,
- overshoot limitation.

We consider now a PID (Proportional Integral Derivative) controller because it is very classical and widespread in industry.

$$s(t) = K_P.e(t) + K_I \int_0^t e(t).dt + K_D.\frac{de(t)}{dt} \qquad (2.1)$$

K_P is the proportionality factor of the proportional component.

K_I is the coefficient of the integral component.

K_D is the proportionality factor of the derived component.

If PIDs are implemented onto microprocessors, approximations are made to ensure this implementation [22]. The PID algorithm can be programmed in the following form:

$$\begin{aligned} p(i) &= K_P.\big(r(i) - y(i)\big) \\ I(i) &= I(i-1) + b_{i_1}.e(i) + b_{i_2}.e(i-1) \\ D(i) &= a_d.D(i-1) - b_d.\big(y(i) - y(i-1)\big) \\ \Delta u(k) &= u(i) - u(i-1) = \Delta p(i) + \Delta I(i) + \Delta D(i) \end{aligned} \qquad \begin{aligned} \Delta p(i) &= p(i) - p(i-1) \\ \Delta I(i) &= I(i) - I(i-1) \\ \Delta D(i) &= D(i) - D(i-1) \end{aligned} \qquad (2.2)$$

With:

r(i) the setpoint at time i,

y(i) sensor output at time i,

e(i) the static error (difference between the setpoint and output at time i

$$e(i) = r(i) - y(i),$$

b_{i1}, b_{i2}, a_d and b_d are parameters depending on K_I, K_D, time i and i - 1, and the approximation methods, these aspects are quite classical and not further detailed here.

Controller receives the messages sent by the sensor (place *received* figure 4). So the *controller_activity* activity is validated. The crossing of this activity launches the control algorithm coded in the *Output1* output. After the crossing of the *controller_activity* instantaneous activity, a *frame*-type token is put in the *controller_output* place. This token contains three attributes:

- *id* which is the identifier associated with the controller,
- *size* which is the size of the message sent,
- *value* is the setpoint.

[Fig. 4. Model of the controller]

2.2.3 Actuator

The model of the actuator resembles the model of the controller. The complete model is available in [19].

2.2.4 Process

In a networked-control context, the sensor transmits information to the controller which finally transmits the control to the actuator (Figure 5). The only component activated in a strict periodic way is the sensor. It means that the times of transmission can be variable. The controller sends the control each time T_{c_i}, the actuator receives them at times T_{a_i}. The controller and actuator activation times are not periodic anymore and are not known in advance.

Because of these difficulties, we decide to evaluate the actual response (for each sampling time) to a succession of steps representing the variations of the control, using the Laplace-transform formalism.

We suppose that the actuator maintains a constant value until the reception of a new control value. The input signal is displayed in (figure 6). Each a_i represents a control sent by the controller and T_{a_i} represents the time for the application of the control a_i on the process.

[Fig. 5. Component activation times]
[Fig. 6. Control signal]

At *t*, the signal at the input of the process can be written as a sum of steps:

$$e(t) = a_1 U\left(t - T_{a_1}\right) + \left(a_2 - a_1\right) U\left(t - T_{a_2}\right) + \left(a_3 - a_2\right) U\left(t - T_{a_3}\right) + \ldots \quad (2.3)$$

With the step U(t) defined as:

$$U(t) = \begin{cases} 1 \ \ when \ \ t \geq 0 \\ 0 \ \ when \ \ t < 0 \end{cases}$$

$$E(s) = \left(a_1 \frac{e^{-s.T_{a1}}}{s} \right) + \left[\left(a_2 - a_1\right) \frac{e^{-s.T_{a2}}}{s} \right] + \left[\left(a_3 - a_2\right) \frac{e^{-sT_{a3}}}{s} \right] + \ldots \quad (2.4)$$

a_i gives the control value at time T_{c_i} . T_{a_i} is the time when the control value is taken into account by the actuator.

We have:

$$S(s) = E(s).G(s) \qquad (2.5)$$

The Laplace transform of the output S(s) is classically calculated (equation 2.5), from the transform of the input E(s) and the process transfer function G(s).

Case of a second order system:

Now let us consider a system of the second order: G(s) can be written as:

$$G(s) = \frac{K}{(s+x_1).(s+x_2)} = \frac{K_1}{(s+x_1)} + \frac{K_2}{(s+x_2)}$$

which gives:

$$S(s) = \left[\left(a_1 . \frac{e^{-sT_{a1}}}{s} \right) + \left[(a_2 - a_1). \frac{e^{-sT_{a2}}}{s} \right] + \left[(a_3 - a_2). \frac{e^{-sT_{a3}}}{s} \right] + \ldots \right] . \left[\frac{K_1}{(s+x_1)} + \frac{K_2}{(s+x_2)} \right] \qquad (2.6)$$

From this equation one can deduce the output according to time:

Let's consider $\alpha_1 = \frac{K_1}{x_1}$ and $\alpha_2 = \frac{K_2}{x_2}$

$$\begin{aligned} s(t) = {} & \alpha_1 . a_1 \left(1 - e^{-x_1 . t}\right) + \alpha_1 . (a_2 - a_1) \left(1 - e^{-x_1 . (t - T_{a2})}\right) + \alpha_1 . (a_3 - a_2) \left(1 - e^{-x_1 . (t - T_{a3})}\right) + \ldots \\ & + \alpha_2 . a_1 \left(1 - e^{-x_2 . t}\right) + \alpha_2 . (a_2 - a_1) \left(1 - e^{-x_2 . (t - T_{a2})}\right) + \alpha_2 . (a_3 - a_2) \left(1 - e^{-x_2 . (t - T_{a3})}\right) \end{aligned} \qquad (2.7)$$

Let us remind that the sensor measures information periodically. At the time T_s, the actuator applies the following control:

$$s(T_s) = \alpha_1.a_1\left(1 - e^{-x_1.T_s}\right) + \alpha_2.a_1\left(1 - e^{-x_2.T_s}\right) \quad (2.8)$$

At the time $2.T_s$, there are two possibilities:

1) The sensor does not receive a new control before the time $2.T_s$, so the delay is higher than the sampling period, in this case the previous equation remains valid.

2) The actuator receives a new control before the time $2.T_s$. In this case the output can be calculated by the expression:

$$s(2.T_s) = (s_1 - a_1.\alpha_1).e^{-(x_1.T_s)} + a_2.\alpha_1 - (a_2 - a_1).\alpha_1.e^{-(x_1.(2.T_s - t_2))} + (s_2 - a_1.\alpha_2) \times e^{-(x_2.T_s)} + a_2.\alpha_2 - (a_2 - a_1).\alpha_2.e^{-(x_2.(2.T_s - t_2))} \quad (2.9)$$

With the same process, at the time $i.T_s$ the actuator receives a new control between the time $(i-1).T_s$ and $i.T_s$, the output is obtained by:

$$s(i.T_e) = (s_1 - a_{i-1}.\alpha_1).e^{-(x_1.T_s)} + a_i.\alpha_1 - (a_i - a_{i-1}).\alpha_1.e^{-(x_1.(i.T_s - t_i))} + (s_2 - a_{i-1}.\alpha_2).e^{-(x_2.T_s)} + a_i.\alpha_2 - (a_i - a_{i-1}).\alpha_2.e^{-(x_2.(i.T_s - t_i))} \quad (2.10)$$

These equations are implemented in the model representing the sensor (output of the *sensor* activity), in this case we considered useless to add another model to represent the process, since generally a sensor is associated only to one process. If it is needed, this separation can be easily applied. Let us note that this calculation is only made once every T_s time units (period of the sensor), since the output is not determined on a continuous basis. We

calculate the output of the system each sampling time by taking account of the variable delays which vary from one sampling period to another.

The transfer function which will be used in the following is drawn from the work of Aström and Wittenmark [23]. This example is used since several works are available in the literature from this example, in particular in the field of Networked Control Systems, so this example can be considered as a benchmark.

$$G(s) = 1000 / s.(s + 1)$$

The interest is that this system is known. It can be shown that this system become unstable when the delay is greater or equal to 48% T_s. This important figure will be considered in the next parts.

The final model of the whole system (control system + process) can be obtained by joining the models of the four basic elements described above.

3 Influences of the faults

The goal of our study is to analyze the influence of the transient faults of the network and their consequence on the reliability of the control system. The permanent faults are not taken into account in this study.

3.1 Influence of the variable delay

To illustrate the influence of the variable delay, let us consider the example of Aström and Wittenmark presented in the end of the previous section. We know that a constant delay

equal to 48% of the sampling period on each message transmitted through the network makes the system unstable.

Let's consider first the case when the delay is always lower than the limit. This delay is random according to a uniform law in an interval from time ranging between 0 and 40% of the sampling period. To study the influence of the variable delay on the performances, we repeated 1000 tests of the response to the step by simultaneously introducing a random delay on the delivery of information between sensor and controller on the one hand and controller and actuator on the other hand.

[Fig. 7. Distribution of the overshoot, the response time at 10% and the rising time in the presence of variable delays (the x-axis represents the value of the parameter of performance and the y-axis represents the number of stories which reached the value given in X-coordinate)]

Figure 7 shows the distribution of the parameters of performances (overshoot, response time and rising time) in the presence of a variable delay. Vertical thin black lines represent the value of the concerned parameter for a constant delay (40% T_s, 20% T_s, 0% T_s).

We can see that the influence of the variable delay acts on the overshoot which remains bounded by two values which correspond to the minimal and maximal constant delays. A large number of stories (400 stories) have an overshoot which approaches a particular value 1.11474 which corresponds to the value of the overshoot in the case of an average constant delay (20% T_s), which is quite normal. The response time takes a value among five particular class values.

For the rising time on the contrary, the variable delays improve the rising time, nearly 90% of the stories have a rising time of 0.06 for a rising time of 0.07 in the case of a minimal fixed delay. Considering the system is digitalised, the output of the system is measured only at the moments of activation of the sensor, which explains why the values of the rising time are multiples of the sampling period. For the 1000 stories no case of instability appeared. This can

be explained by the fact that all the delays are lower than the worst case (48% of the sampling period).

The following test (figure 8) aims at seeing the influence of the variable delay on the stability of the system. The delays are distributed according to a uniform law in a time interval ranging between 20% and 60% of the sampling period.

[Fig. 8. Evolution of the output in the presence of a variable delay ranging between 20% and 60% of the sampling period]

Figure 8 superimposes the outputs for a test of 1000 stories. It is noticed here that the disturbance of the system is very strongly accentuated with the increase in the variable delay range. We can also notice some cases where the system has trend to instability. It is interesting to notice that even if the system has a trend to instability, it does not become unstable, even if certain delays exceed the value of 48% T_s. That indicates that the system is robust with delays larger than the maximum delay, if the frequency of occurrence of these delays is sufficiently weak. This is an interesting result which shows that the system is robust to some over-delays, so what is important here is not only the value of the delays, but also the distribution of the delays during time.

3.2 Influence of message losses

We consider now that any information exchanged between the components can be lost before being received by the receiving station. We distinguish two types of messages: measurement and control.

[Figure 9: Response of the system to a reference step: example of the loss of the third sample]

Figure 9 shows the response of the system to a reference step, when there are some information losses. One can notice that the influence of a loss during the same time of treatment is not identical, according to whether it relates to measurement or control. In the first case, the controller does not receive information and does not calculate a new control. In the second case, a control is elaborated but it is not sent to the actuator.

Figure 10 shows the distribution of performance parameters in the presence of probabilities on the messages (for each probability: 1000 stories of response to a step).

[Figure 10: Distribution of the overshoot, the response time to 10% and the rising time for loss probabilities 0.01, 0.02, 0.03 (the x-axis represents the value of the parameter of performance and the y-axis represents the number of stories which evolved to the value given in X-coordinate). Overshoot is expressed as 1 for normal value with no overshoot (1.1 means 10 % overshoot); Response time and rising time are expressed in ms]

The vertical thin black lines represent the value of the parameter concerned if there are no losses. It is noticed that in the majority of cases, the performances parameters are not affected and that the values approach the values obtained without loss. The results show that quality (in particular for the response time and the rising time) in the presence of loss can be better than in the ideal case (no loss). Although in the majority of cases the overshoot did not change, we can observe some situations where the overshoot is more than 160% of the setpoint; these situations correspond to the losses of the messages in the transient stage: that can be dangerous, when these overshoots are non tolerable by the system.

3.3 Superposition of variable delays and losses of the messages

This paragraph studies the simultaneous influence of the variable delays and the losses of the messages. The delay is drawn by chance according to a uniform law in a time interval ranging between 0 and 40% of the sampling period. And the probability of loss of the messages is fixed at 0.01.

[Fig. 11. Distribution of the overshoot, the response time at 10% and the rising time in presence of variable delays and the messages losses (x-axis represents the value of the parameter of performance and y axis represents the number of stories which evolved to the value given in X-coordinate)]

The results of the superposition of the two types of faults resemble those of the variable delay with a small degradation over the overshoot and the response time. The overshoot can be larger than 1.34263 (value obtained for a maximum constant delay). As in the case of a variable delay only, a large number of stories (370 stories) have an overshoot which approaches 1.11474 (overshoot obtained for a constant delay with 20% T_e). For the response time, the superposition of the two types of faults gives new values which do not appear when each case is considered separately. These results show the difficulty in evaluating the influence of the superposition of the two types of faults.

This part showed the relationship between disturbing events with a communication point of view, and the evolution of the performance parameters, with a control point of view.

4 Towards a study of reliability

We can define the mission of the system as being the control of a physical variable with a certain quality of control. The quality of control is defined by certain criteria: limited overshoot, minimal rising and response times, and no trend to instability. Any behaviour of the system which does not respect the specifications must be regarded as a failure of the system. The considered failures are probabilistic, so the failure of the system itself is probabilistic and can be put in relation to the concept of reliability. This may be justified by the fact that in many industrial systems, an excessive value of a physical variable may be dangerous for a part of the system and then impact the reliability of the whole system (for example over current in an electric drive, over torque on a shaft, over temperature in a semi conductor, etc.). We define the "failure by a criterion" as being the probability that this criterion is not respected, for example the failure by overshoot means that the overshoot is too high. The system will be considered as failed when at least one criterion (overshoot, response time, rising time) fails.

Criteria of evaluation

In order to obtain significant evaluations, it is necessary to produce several simulations to obtain a degree of confidence of the estimated variables. Let us remind that with Monte Carlo approach, it is not guaranteed to have the exact values of the parameters to be evaluated. In fact, the method guarantees only the probability that the number of stories N ensures a result with a given precision.

To define the confidence in the results, two parameters are used: a precision interval and a percentage of values which belongs to this interval. Each time a new history i (i being the number of the history) is launched, a new average value V_i is determined and the

difference $d_i=V_i - V_{i-1}$ is calculated. If d_i belongs to the precision interval, the number of histories belonging to the precision interval increases. If the percentage of histories belonging to the precision interval reaches a given threshold, the simulation stops.

For example an accuracy of 5% of the precision interval and 95% of the percentage of membership requires that 95% of the calculated d_i be lower than ±5% of the last average.

In the next sub-sections are studied the cases of the failure by overshoot and the failure by stability. Depending on the system, other cases can be studied.

4.1 Failure by overshoot

The failure by overshoot consists in fixing a maximum gap D_{ov} between the output and the ideal response to a step. When there are transient faults, if at a given time the overshoot is greater than the set-point D_{ov}, we consider a failure by overshoot. We can physically justify this choice since in many industrial applications the overshoot of certain variables can be dangerous.

4.1.1 Influence of a variable delay on the failure by overshoot

We repeated the tests of response to the step (history) by simultaneously introducing a random delay on the delivery of information between sensor and controller on the one hand and controller and actuator on the other hand. For each simulation, we detect if the criterion of overshoot is satisfied or not. With each new history, we calculate the average number of times

where the criterion is not satisfied (overshoot higher than the threshold). Simulation is stopped when we obtain a difference between two average successive values lower than ±5% of the last average, and this at least in 90% of the stories.

[Fig. 12. Probability of failure by overshoot in the presence of variable delays, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure]

Figure 12 illustrates the relation between the variable delay which expresses the quality of service of the network and the failure by overshoot, representative of the quality of control. The curves show the change of the probability of failure by overshoot according to the D_{ov} variation. The three curves correspond to three classes of variable delay. The delay is uniformly distributed between 10% and 30% of the sampling period for the first case, between 15% and 35% for the second case and between 20% and 40% for the third case. These values are taken to illustrate the influence of the variable delay. They are however realistic, for the first case, if the communications protocol manages the access in a priority way, we can for example imagine a situation where:

- a high priority message which is always sent before the messages relating to the application, introducing a delay of 10% of the sampling period (125 bits for a rate of 125 000 bit/s)
- after this delay the network will be temporarily unreachable for one random time between 0% and 20%.

Other causes can generate this type of delay: for example in non deterministic open networks where the traffic can vary in a random way. The best way of representing these delays is more accurately to represent the traffic and the unavailability of the network (transient failure).

4.1.2 Influence of losses on the failure by overshoot

The same step for the evaluation of the probabilities of failures is applied to estimate the influence of the losses of the messages. The tests of response to the step (history) are repeated by simultaneously introducing probabilities of losses on the delivery of information between sensor and controller on the one hand and controller and actuator on the other hand. The Monte Carlo simulation is used for the evaluation of the probability of failure by overshoot and the same criterion of stop is used as in the case of the variable delays (5% of precision and 90% of percentage of membership).

[Fig. 13. Probability of failure by overshoot in the presence of losses of information, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure]

Figure 13 shows the probability of failure as a function of the threshold constraint for various probabilities of loss. These results show the existence of a relation between the probability of the losses of a message and the failure by overshoot of the system.

As we pointed out concerning the variable delay, we can wonder about the origin of the losses to characterize their probability of occurrence according to the environment and the traffic circulating on the network. We can for example take into account the probability of having two lost consecutive messages (this possibility is plausible in the case of wireless networks).

4.1.3 Influence of the superposition of variable delays and loss of messages on the failure by overshoot

The two previous paragraphs considered only one source of fault and made it possible to highlight the relation between the source of fault and the failure by overshoot. The following part aims to highlight the simultaneous influence of these two sources on the failure by overshoot.

[Fig. 14. Probability of failure by overshoot in the presence of the losses and the variable delays, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure]

The delay is uniformly chosen between 10% and 30% of the sampling period and the probability of loss is fixed at 0.01. On figure 14, the three curves correspond to the three cases: 1) variable delay 2) losses of messages 3) superposition delay plus losses of the messages.

A probability of loss of messages of 0.01 on a network is a rather high probability, in practice this probability is smaller; while a delay varying uniformly between 10% and 30% is more realistic. With these values we find that the overshoot is much more sensitive to the variation of the delay than to the loss of the messages (the probability of exceeding of 10% the setpoint is about 0.7 for a variable delay and of 0.009 for a probability of loss of 0.01). The results of simulation show that there are cases where the system is reliable with respect to a single source of failure but not for the superposition of two different sources.

4.2 Failure by stability

The purpose is here to identify situations where the disturbances have a trend to make the system unstable [24]. To define the failure by stability, we supervise the successive peaks of the response to the level in the presence of these transient faults. If we observe peaks of increasing amplitude, this is a situation of trend to instability.

[Figure 15: Trend to instability: example of an output in Volts as a function of the time in ms]

More precisely, the criterion is as follows: if we measure three successive amplitudes of peaks (in absolute value) increasing, the system is declared in failure by stability.

[Table 1. Probability of failure by stability]

Table 1 shows the results obtained for the probability of failure by stability. The results are given for a delay varying uniformly between 20% and 60% of the sampling period. In the general case, a first study should be achieved before to decide the unacceptable number of peaks, this parameter depending on the context of the application. In the present study, a succession of three peaks with increasing amplitude was considered.

4.3 Evaluation of reliability

According to the definition of reliability given previously, the system fails if one criterion is not satisfied. To determine the reliability:

- the probability of failure for each criterion is first calculated, under specified conditions (delay, probability of loss, or the superposition of both)

- reliability according to our definition is then determined: the calculation of reliability is done in two ways: by the theorem of Poincaré knowing the probabilities of failures for each criterion or direct simulation on the system.

To calculate reliability it is necessary to fix the thresholds for each parameter. In this paragraph the study is limited to the overshoot and the trend to instability. One fixes D_{ov} at 70%. For stability the same criterion that in the previous sub-section is used (three increasing peaks).

The failure of the system will take place if a failure by stability or a failure by overshoot arises:

$$P_{failure\_system} = P\left(def\_overshoot \cup def\_stability\right)$$

$$= P_{def\_overshoot} + P_{def\_stability} - P\left(def\_overshoot \cap def\_stability\right) \qquad (4.1)$$

The probability of each of these three terms can be calculated from simulations on the SAN models (see section 2.1). For example to calculate the probability of the third term it is

enough to add two places *failure_stability* and *failure_overshoot* and to put a token in the place *failure _overshoot* after the occurrence of a failure by overshoot and a token in the place *failure _stability* in the case of a failure by stability. The probability of the term $P(_{def\_overshoot} \cap _{def\_stability})$ will be the probability of having a token in each of the two places.

Table 2 gives the evaluation of these terms for an accuracy of 5% and one percentage of membership of 90%.

[Table 2. Evaluation of the probabilities of failure]

From these probabilities, reliability can be calculated with equation 4.1 (theorem of Poincaré).

Reliability can also be evaluated directly in the following way: for each history we test wether at least one criterion is not satisfied, if it is the case, a counter concerned with unreliability *ndef (T)* which represents the number of stories where the system was found in a failing state, is incremented. Reliability is estimated by the following relation:

$$R(t) = 1 - \frac{n_{déf}(t)}{N} \qquad (4.2)$$

Where $n_{def}(t)$ is the number of stories which led to a failure state,

N is the total number of simulated stories,

R(t) is the reliability.

Table 3 compares the results of evaluation of reliability by the two methods.

[Table 3. Results of evaluation of reliability by the two methods]

Values on the variations are always lower than the precision defined at the simulation stage (5%).

To obtain the results of section 4, four simulation campaigns (a campaign gathers several studies) were launched:

- campaign_1: for the evaluation of the influence of the variable delay.
- campaign_2: for the evaluation of the influence of the losses of messages.
- campaign_3: for the evaluation of the influence of the superposition of the variable delay and losses of messages
- campaign_4: for the direct evaluation of reliability.

The first three campaigns make it possible to evaluate the probabilities of failure by overshoot, response time and stability under various conditions (variable delay, losses, superposition). The fourth campaign makes it possible to evaluate directly reliability (section 4.3). All the campaigns can be carried out in parallel on the same machine. The time of simulation is long but acceptable: as an example in order to determine the reliability in the last paragraph, 1 158 035 stories were necessary before the stop of simulations (equivalent to 3 hours of simulation).

5 Conclusion

The work presented here is an approach for the evaluation of reliability of Networked Control System (NCS). The approach focuses in particular on transient faults which appear with digital networked components. Introduction of control loops, with some tolerant-fault mechanisms, can also improve the actual reliability of the system, but at the same time

complexifies the evaluation. When a control loop is introduced in a system, the objective is to obtain a new system defined by performance objectives: stability, response time, overshoot, static error… The presence of fugitive failures will bring more or less strong degradations of the performances being likely in certain cases to cause irreversible damages (damage or destruction of components, rejections of effluents, etc). It is important thus to force a limit not to be exceeded for the transient disturbances. The problem of reliability is thus to calculate the probability that the performances of the system remain within acceptable limits beyond whose the risk is intolerable. As it is proposed by [25], reliability becomes a vector of probabilities in order that each performance indicator remains within tolerable limits. For each probability, disturbing events or the combinations of disturbing events (cuts) can be defined. It is however not possible to provide an analytical expression of these probabilities because they depend not only on the probabilities of occurrence of the fugitive failures but also both on the thresholds fixing the limit of acceptability and on functional dynamics of the system. This is why we proposed a simulation approach.

First, it is intended to identify the incidence of two types of fugitive failures: the loss of a sample and the delay. The step response of the system is studied by Monte-Carlo simulation, by random injection of a loss or random variation of the delays [26]. We use acceptable performance threshold detectors on rising time, response time, overshoot, trend to instability. The criterion for stopping simulation is related to the probability of membership with a certain confidence interval on the result. Finally, both disturbances are used together, highlighting cumulative effects. Because the system is dynamic, the probability of non failure of the system is not the product of the probabilities of non occurrence of the failures.

This work does not intend to solve completely the problem of the evaluation of the reliability of Networked Control Systems (NCS), the resolution of this problem being not trivial. We propose a method and the tools associated to approach this evaluation by simulation and thus to bring a help to the systems design satisfying with critical requirements on certain performance parameters. Thanks to Monte Carlo simulation, the approach suggested made it possible to quantify the influence of the transient faults on reliability, and so, the influence of the quality of service of the communication function on the quality of control of the system.

One of the most interesting aspects to develop for the assistance to the design of NCS is undoubtedly the search for importance factors which are specific to these systems. In "classical" reliability, importance factors define the sensitivity of the reliability of the system to certain factors related on the component or the structure. This work showed that reliability depends also on functional parameters (pole and zeros of the transfer function…), for example it will be interesting to study the sensitivity of the reliability to a controller coefficient, in a case where the increase can improve the dynamics of the system but degrade its reliability; in this case a compromise must be required.

References

[1] Hoppe T., Kiltz S., Dittmann J. Security threats to automotive CAN networks – practical examples and selected short-term countermeasures, Reliability Engineering & System Safety, In Press, Accepted Manuscript, Available online 5 July 2010

[2] Aldemir T., Guarro S., Mandelli D., Kirschenbaum J., Mangan L.A., Bucci P., Yau M., Ekici E., Miller D.W., Sun X., Arndt S.A., Probabilistic risk assessment modeling of digital instrumentation and control systems using two dynamic methodologies, Reliability Engineering & System Safety, Volume 95, Issue 10, Pages 1011-1039, October 2010.

[3] Moncelet G., Christensen S., Demmou H., Pauldetto M., Porras J. Dependability evaluation of a simple mechatronic system using coloured Petri nets In: Workshop on Practical Use of Coloured Petri Nets and Design CPN. p.189-198. Aarhus University, Aarhus, Denmark, 1998.

[4] Portugal P.J., Carvalho A. A Stochastic Petri Net Framework for Dependability Evaluation of Fieldbus Networks – A Controller Area Network (CAN) Example. International IEEE Conference in Mechatronics and Robotics – MECROB, 2004.

[5] Jumel F., Thiriet J.M., Aubry J.F., Malasse O. Towards an information-based approach for the dependability evaluation of distributed control systems", 20th IEEE Instrumentation and Measurement Technology Conference (IEEE/IMTC'2003), Vail (Colorado, United States), 20-22nd May 2003, pp. 270-275.

[6] Barger P., Thiriet JM, Robert M. – Dependability study in distributed control systems integrating smart devices, Low Cost 2004, Ottawa (Canada), June 2004, pp. 79-84.

[7] Ligusova J., Thiriet J.M., Ligus J., Barger P. Effect of Elements Initialization in Synchronous Networked control System to Control Quality" - RAMS/IEEE conference 2004 (Annual Reliability and Maintainability Symposium), Los Angeles, January 2004, 6 pages (ISBN: 0-7803-8216-1).

[8] Richards J.P., Time Delay Systems: An overview of some recent advances and open problems. Automatica. Vol (39), No. 10, p:1667-1694, October 2003.

[9] Walsh G.C., Ye H., Bushnell L. Stability analysis of networked control systems. Proceedings of the 1999 American control conference. Vol (4), pp.2876-2880, San Diego, CA, 1999.

[10] Cauffriez L., Ciccotelli J., Conrard B., Bayart M. Design of intelligent distributed control systems: dependability point of view, Reliability Engineering And System Safety. Vol (4), pp. 19-32, 2004.

[11] Brissaud F., Barros A., Bérenguer C., Charpentier D. Reliability Study of an Intelligent Transmitter, 15th ISSAT International Conference on Reliability and Quality in Design, USA, 2009.

[12] Levitin G., Hausken K. Intelligence and impact contests in systems with redundancy, false targets, and partial protection, Reliability Engineering and System Safety, Volume 94, Issue 12, December 2009, Pages 1927-1941.

[13] Champagnat R. Supervision des systèmes discontinus: définition d'un modèle hybride et pilotage en temps-réel, Thèse de Doctorat de l'Université Paul Sabatier de Toulouse, 1er octobre 1998.

[14] Moghavar A., Meyer J.F. Performability modeling with stochastic activity network. In proc of the 1984 real-time systems Symp. p:215-224, Austin, TX, USA, 1984.

[15] Sanders W.H., Meyer J.F. Stochastic activity networks: formal definitions and concepts, Lectures on formal methods and performance analysis: first EEF/Euro summer school on trends in computer science, pp 315-343. Springer-Verlag New York, Inc., New York, NY. 2002.

[16] Malhotra, M. , Trivedi K. Dependability Modelling Using Petri-Nets, IEEE Transaction on reliability, Vol. 44, No. 3, pp. 428-440, 1995.

[17] Deavours D.D., Clark G., Courtney T., Dalys D., Derisavi S., Doyle J.M., Sanders W.H., Webster P.G. The Moebius framework and its implementation. IEEE Trans. On Soft. Eng, Vol. 28, No 10, pp. 956-969, 2002.

[18] David R., Alla H. Discrete, Continuous, and Hybrid Petri Nets, Springer, 2004.

[19] Ghostine R. - Influence of transient faults onto reliability of Networked Control Systems (Influence des fautes transitoires sur la fiabilité d'un système commandé en réseau) – PhD report, 12 June 2008, Nancy-Université, CRAN [in French].

[20] Aström K.J., Hagglund T. Automatic tuning of simple regulators. 9th IFAC WorId Congress., Budapest, Hungary, 1984, p.267-272.

[21] Smith C.L.. Digital Computer Process Control. Intext Educational Publishers, San Francisco, 1972.

[22] Ravichandran C.S., Subha Rani S. Manikandan T. Designing of PID Controller for Discrete Time Linear System Using Balanced Approach Reduced Order Model. American Journal of Applied Sciences. Vol (4)-3, p:155-159, 2007.

[23] Aström K.J. , Wittenmark B. Computer-controlled systems, theory and design. Prentice hall, Englewood cliffs, New Jersey 07632, 1990.

[24] Babak A.S. Stability of networked control systems in the presence of packet losses. In: Proceedings of the IEEE Conference on Decision and Control. p:676-681, USA, 2003.

[25] Hongbin L., Qing Z.. Cut . Tie Set Method for Reliability Evaluation of Control Systems. American Control Conference. USA, 2005.

[26] Ghostine R., Thiriet J.M., Aubry J.F. Dependability evaluation of networked control systems under transmission faults. 6th IFAC on fault detection, supervision and safety of technical processes, SafeProcess. p.1129-1134, Beijing (China), August 29 September 2006.

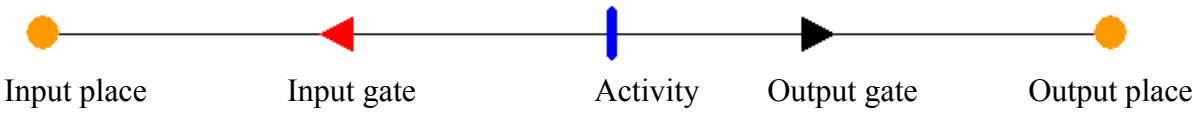

Figure 1: Atomic model under Möbius

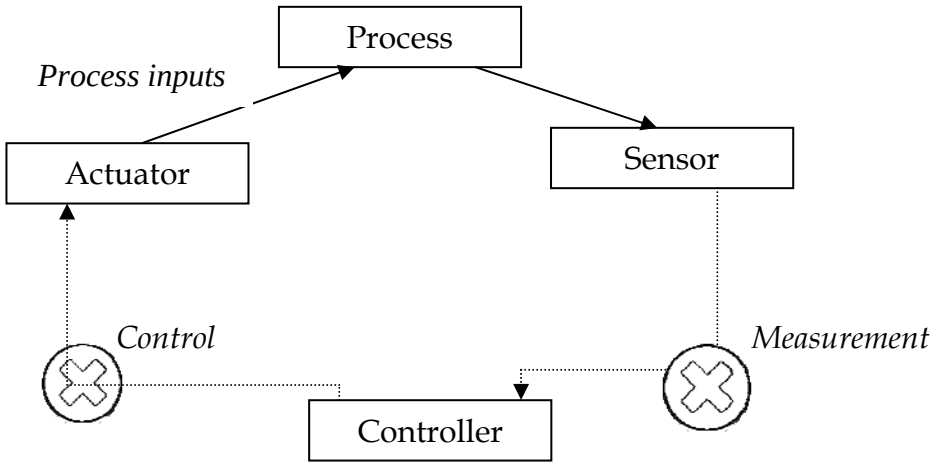

Figure 2: Controlled system

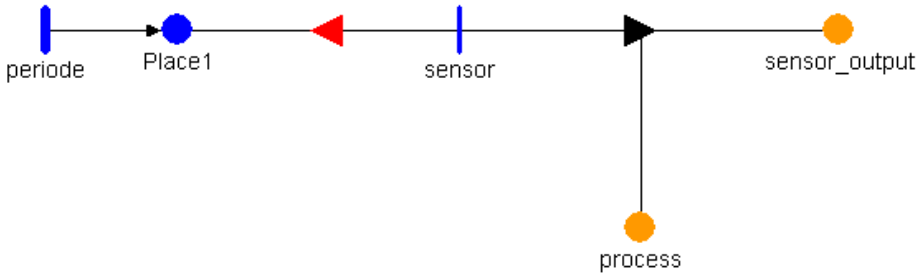

Figure 3: Model of a sensor

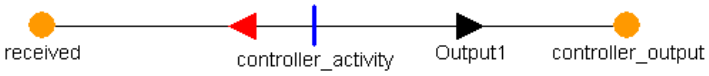

Figure 4: Model of the controller

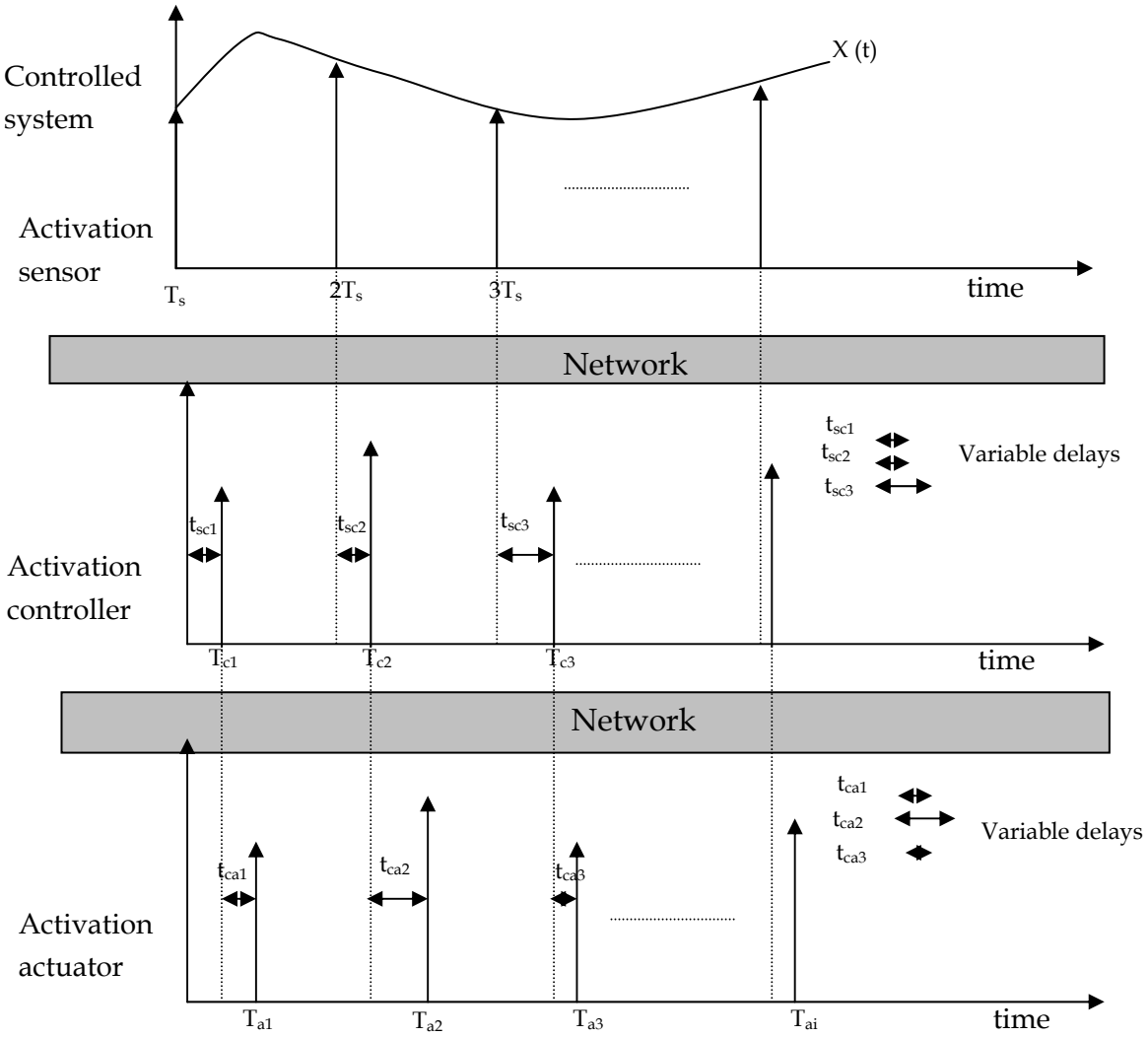

Figure 5: Component activation times

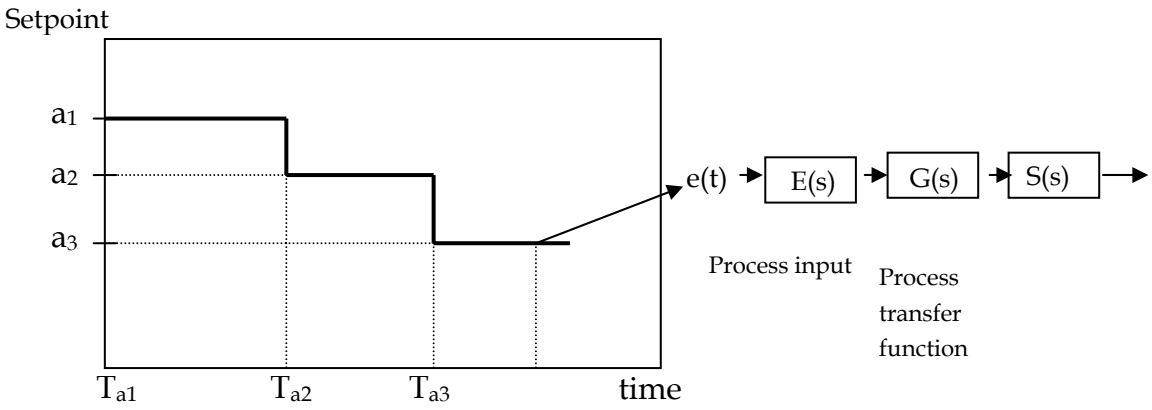

Figure 6: Control signal

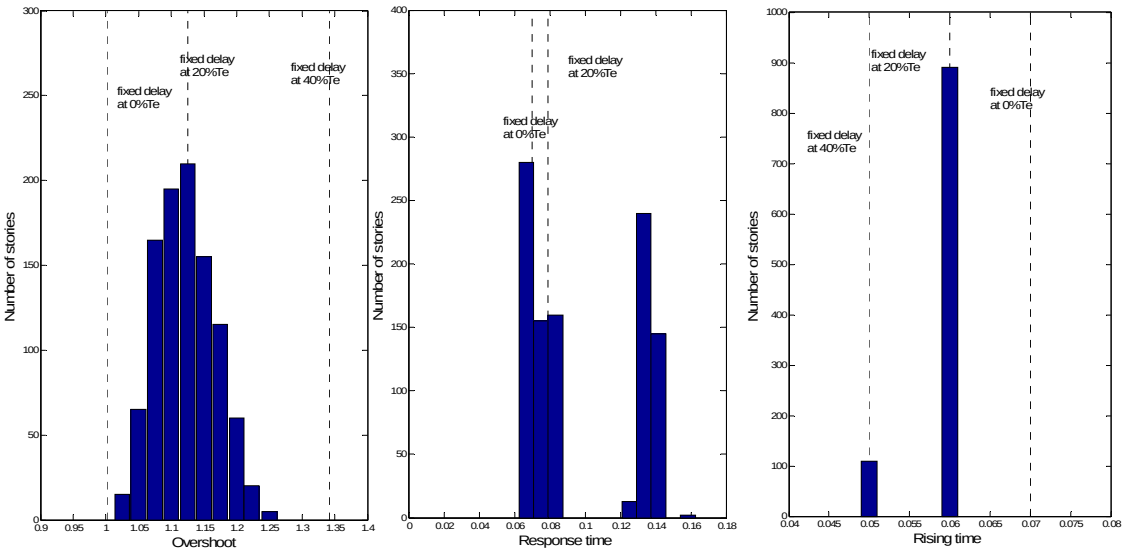

Figure 7: Distribution of the overshoot, the response time at 10% and the rising time in the presence of variable delays (the x-axis represents the value of the parameter of performance and the y-axis represents the number of stories which reached the value given in X-coordinate)

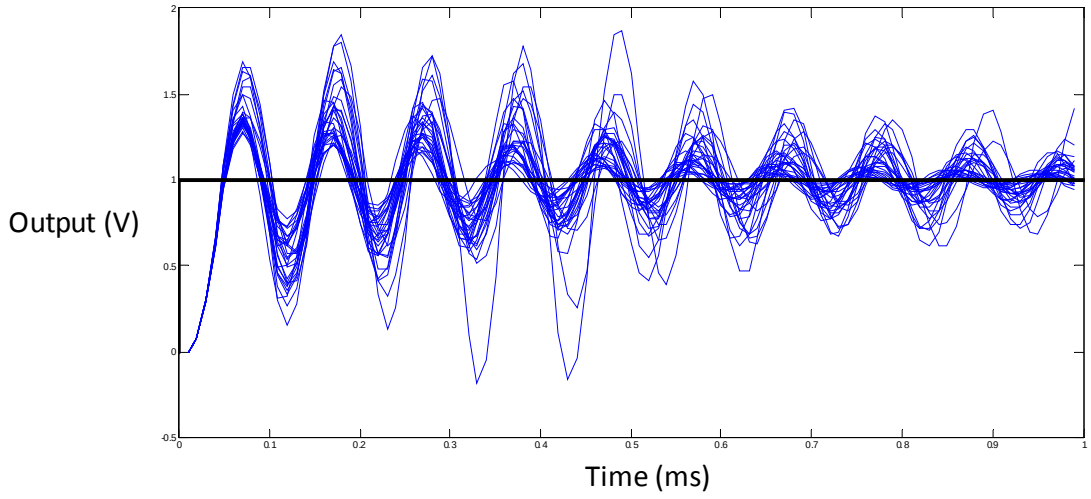

Figure 8: Evolution of the output in the presence of a variable delay ranging between 20% and 60% of the sampling period

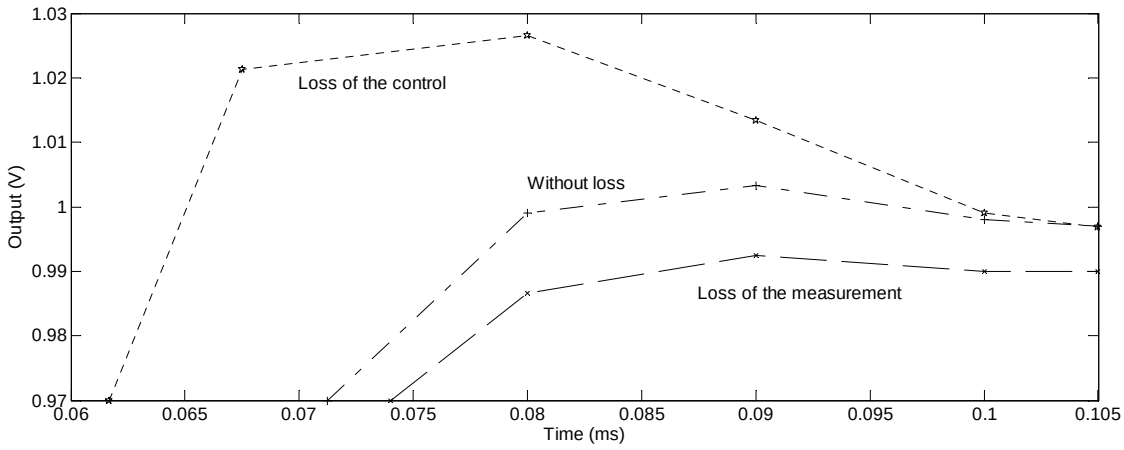

Figure 9: Response of the system to a reference step: example of the loss of the third sample

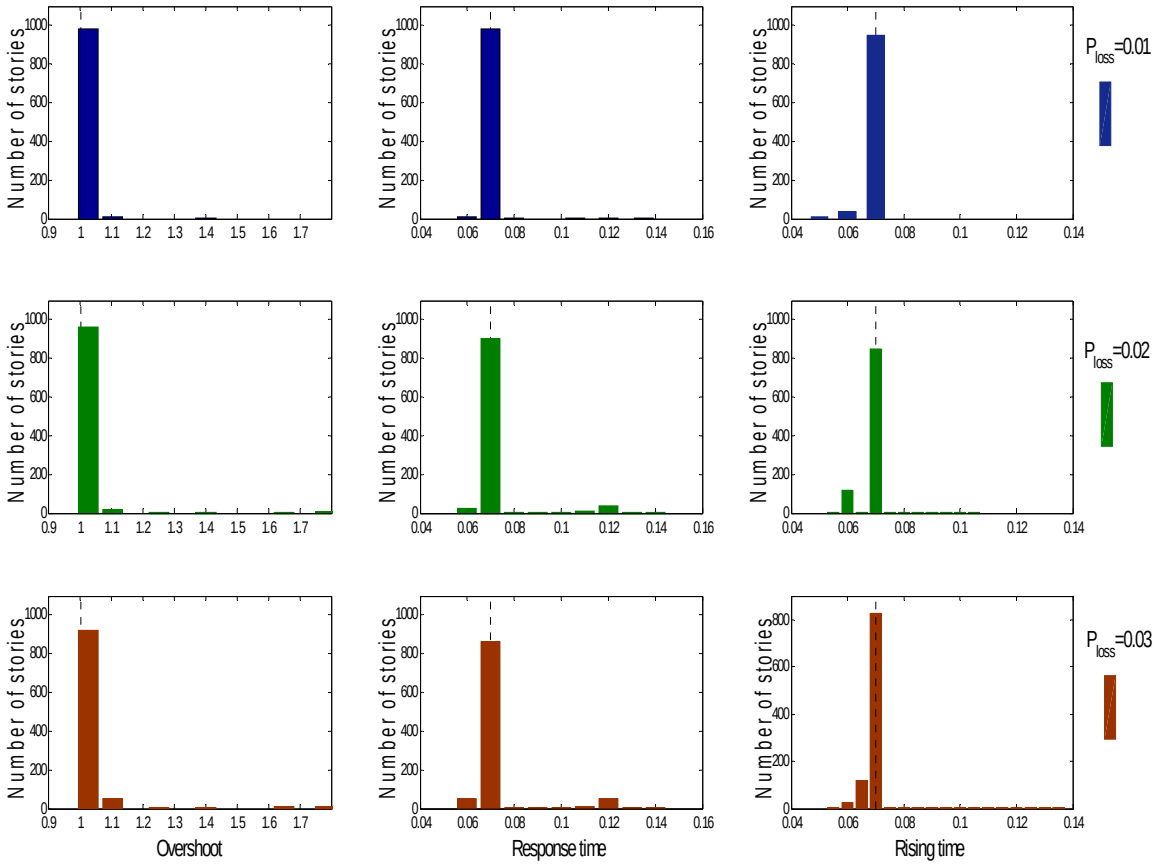

Figure 10: Distribution of the overshoot, the response time to 10% and the rising time for loss probabilities 0.01, 0.02, 0.03 (the x-axis represents the value of the parameter of performance and the y-axis represents the number of stories which evolved to the value given in X-coordinate). Overshoot is expressed as 1 for normal value with no overshoot (1.1 means 10 % overshoot); Response time and rising time are expressed in ms

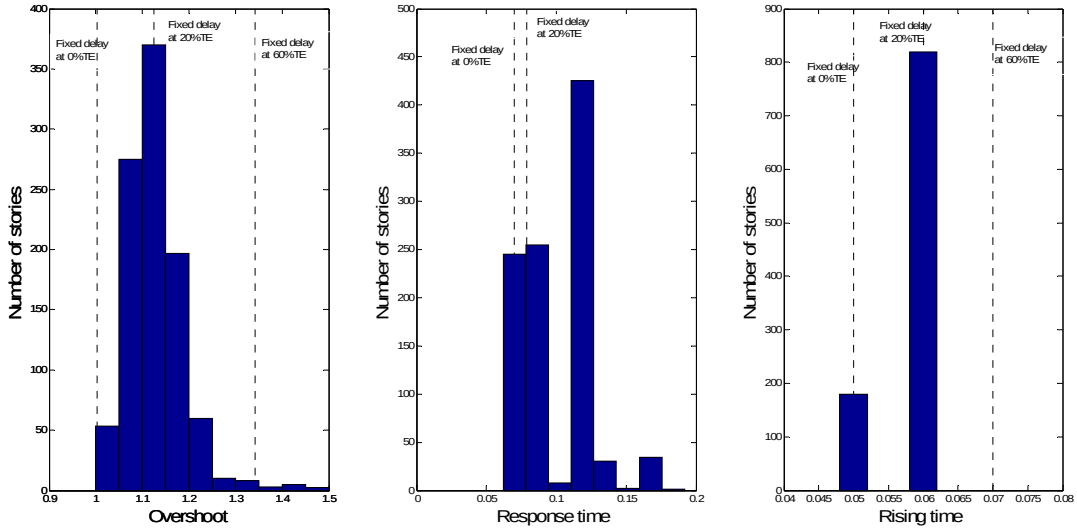

Figure 11: Distribution of the overshoot, the response time at 10% and the rising time in presence of variable delays and the messages losses (x-axis represents the value of the parameter of performance and y axis represents the number of stories which evolved to the value given in X-coordinate)

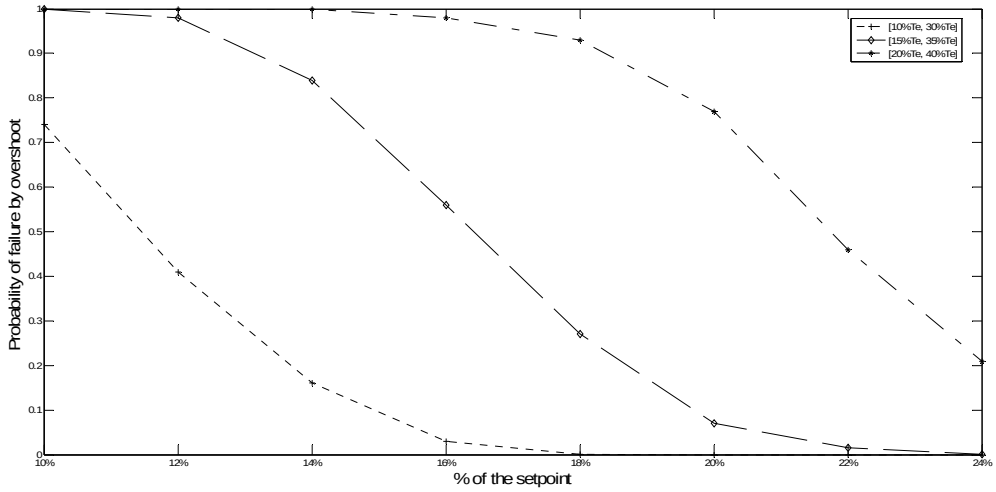

Figure 12: Probability of failure by overshoot in the presence of variable delays, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure

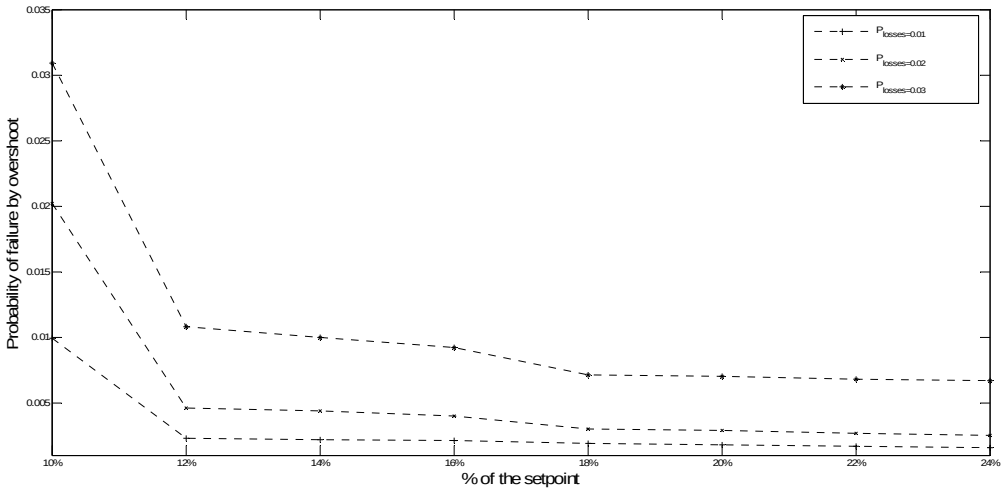

Figure 13: Probability of failure by overshoot in the presence of losses of information, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure

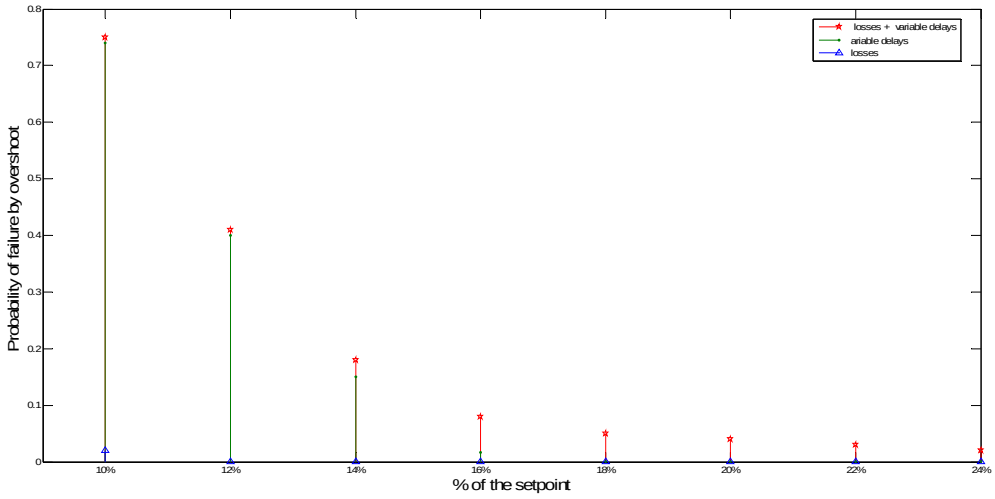

Figure 14: Probability of failure by overshoot in the presence of the losses and the variable delays, the x-axis represents the constraint of D_{ov} threshold (expressed in % of the setpoint), the y-axis represents the value of the probability of failure

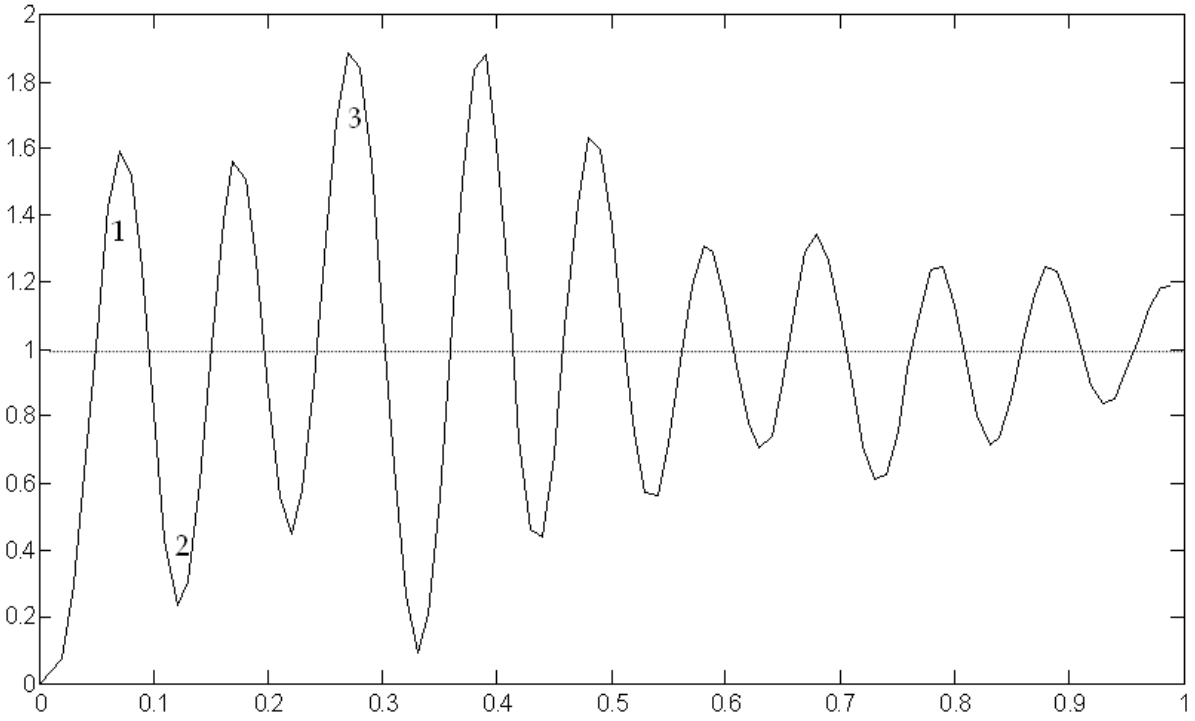

Figure 15: Trend to instability: example of an output in Volts as a function of the time in ms