Optimal Control with Packet Drops in Networked Control Systems

Flavia Felicioni, François Simonot, Françoise Simonot-Lion, Ye-Qiong Song

▶ To cite this version:

HAL Id: inria-00431406

https://hal.inria.fr/inria-00431406

Submitted on 29 Nov 2009

OPTIMAL CONTROL WITH PACKET DROPS IN NETWORKED CONTROL SYSTEMS

Flavia Felicioni*, François Simonot**, Françoise Simonot-Lion***,
Ye-Qiong Song***

* FCEIA – UNR, Universidad Nacional de Rosario, Rosario Argentina (flaviaf@fceia.unr.edu.ar)
**Nancy Université, IECN UMR 7502, B.P. 239, F-54506 Vandoeuvre-lès-Nancy Cedex France
(Francois.Simonot@esstin.uhp-nancy.fr)
*** Nancy Université, LORIA UMR 6508,
Campus Scientifique - BP 239 - 54506 Vandoeuvre-lès-Nancy CedexFrance
({simonot,song}@loria.fr)

Abstract: The stability and performance of a networked control system are strongly influenced by the network delay and packet drops. In this paper, we consider that late arrived sampling data are dropped, so that we only focus on the analysis of the impact of packet drop sequences on the control loop stability and performance. For any dropping sequence specified by *(m,k)*-firm model, and considering a simple mono-variable linear system with a proportional controller and zero control action in case of sampling data drop, we derived the stability conditions based on the upper bound of the plant state variance. It has been shown that the stability only depends on the values of *m* and *k* but not the pattern of the dropping sequence. In case of network overload, this gives much freedom to actively dropping some packets while still keeping the system stable. An analytic method to determine the optimal control gain for any given packet drop pattern is also derived, providing thus a guideline for optimal control and network resource scheduling co-design. Copyright © 2009 IFAC

Keywords: Networked control system, packet drop, (m,k)-firm, stability, performance optimization

1. INTRODUCTION

With the progress of networking technology, more and more control systems are now linked by networks, forming a research topic named Networked Control Systems (NCS) where one or more feedback loops are closed via a shared communication network (Antsaklis and Baillieul 2007). In such systems, the measurements quantized by the sensors are sent to the controller over the network link. After the computation of the controller based on these measurements, the control output is then sent to the actuators that may also be via the network.

Unlike fieldbus design approaches in the past that mainly focused on providing to applications with real-time QoS (Quality of Service) guarantee from network point of view, the design of NCS focuses on the QoC (Quality of Control) optimization from the application point of view. For NCS design, the challenge is how to optimize on the one hand the QoC with the presence of the network induced delay and packet loss, and on the other hand the share of the limited network resources (e.g. network bandwidth). The underlying network of a NCS could be either a fieldbus (e.g. CAN Controller Area Network) or a general-purpose network (e.g. Ethernet, WiFi, or even Zigbee/IEEE802.15.4). As the

network link is often shared among different applications and control loops, the network overload may occur and must be correctly handled. Otherwise excessive network delay and/or packet loss may lead to bad QoC or even to system instability.

For dealing with network overload, three control and implementation (e.g. task and packet scheduling) co-design approaches are available. One is to dynamically change the priority of packets in order to favour the data transfer of some critical control loops to the detriment of the other less critical applications. This approach can be easily implemented on the prioritized bus like CAN (Juanole and Mouney 2007) but its generalization to other kinds of networks is far from straightforward. The second approach consists in dynamically changing the control loops' sampling periods by sending less frequently the sampling data to reduce the network load. This approach has been proposed mainly for optimally sharing the processor on which several controllers are executing (Seto et al. 1996), (Eker et al. 2000), (Simon et al. 2005). However its implementation for the network resource sharing could be difficult for general purpose network. Note that this approach has been proved effective for certain particular networks, as FTT-Can (Antunes et al. 2007). For instance, at the network overload, the period change decision must still be transmitted to the sensor nodes via network, implying complicated mechanisms. The third approach consists in selectively dropping sample data within a certain tolerated limit during the resource overload period, while still keeping the desired

QoC or graceful degraded QoC (Ramanathan 1999), (Hadjicostis and Touri 2002), (Bommannavar and Basar 2008). Among these three approaches, it is clear that changing packet priority does not reduce the total network load but only re-allocates more network resources to the most important control loops and less resources to the others. While changing the period or dropping packets are more radical solutions to the workload reduction.

In this work, we adopt the third approach since comparing to changing the sampling period it appears easier to implement and more efficient than that of the priority change. However the following problems must be carefully addressed before adopting this approach. The first problem is the choice of the dropping sequence when selective dropping should be implemented. What is the impact of the dropping sequence on the QoC? The second problem is how to design optimal control law when a dropping sequence is effective.

In what follows, we describe in section 2 the NCS architecture we considered, the packet dropping policy based on (m,k)-firm model and then give the analytic relationship between the packet drop sequence and the control system and state what are the considered stability criterion and performance optimisation objective. We particularly show in section 2.3 how the network-induced delay can be incorporated into the discrete-time system model under our "zero control" action assumption. This allowed us to only focus on the study of the data drops. Section 3 states the stability condition under any given dropping sequences. As the dropping sequence is repeated periodically of k-periodic, the traditional stability in variance (Aström and Wittenmark, 1997) does not exist when the horizon tends to infinite, so the upper bound of the variance is used instead to assess the system stability. The new metric allows us to upper bound the system output variance. We also give the method to determine the optimal gain for any given dropping sequence. Section 4 provides a numerical application of our analytic results. Through this example we show two points: 1) our analytic results can be easily applied to compute the allowed (m,k)-firm dropping patterns while upper bound the plant state variance; 2) the default value of the controller gain is a tight upper bound of the optimal gain. Section 5 discusses the possible implementation plans of our approach and Section 6 summarizes the new contributions of this work.

2. SYSTEM MODEL AND PROBLEM STATEMENT

2.1 Network Control System (NCS) architecture

The NCS model shown in Fig. 1 consists of a continuous plant (linear, mono-variable) and a digital controller connected onto a networked architecture. The output of the plant is sampled periodically (period h) and has to be transmitted immediately through a network (see signal y_i in Fig. 1). This transmission introduces a delay. We suppose that the closed loop system is controllable if this delay is less than a given value δ. A correct sample, for this plant and this

controller, is a sample that can be transmitted in less than δ to the controller.
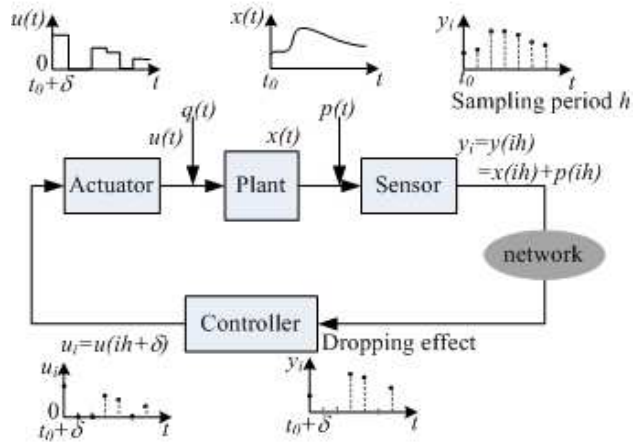

Fig. 1. System architecture

Therefore, as soon as the network is overloaded and is no more able to transmit in time ($< \delta$) the sampled data, the policy that we propose consists in dropping the packet that contains this data since late arrived data is useless for such kind of applications. Note that this strategy, with respect to priority adjustment, allows to keep always an amount of bandwidth for other data flows sharing the network.
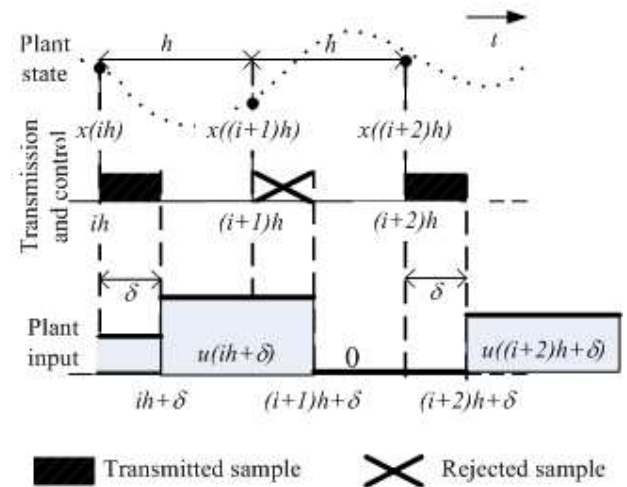
Fig. 2. Algorithm of the controller under sample loss.

We assume that the controller node is synchronized with the sensor one (this service is provided in certain network protocol). The signal management in the closed loop system is: a) clock-triggered sampling of plant output (period h); b) clock-triggered controller (periodic task, period h) executing a constant delay control law under a constant skew δ (a similar policy is proposed in Liou and Ray, 1991); and c)

event-driven actuator, updating the plant-inputs as soon as the controller output becomes available.

If a sensor sample is dropped, it is recognized as vacant sampling by the controller (Halevi and Ray 1988) and the actuator applies the "zero control" action to the plant. Fig. 2 illustrates both cases. Applying a "zero control" signal to the actuator in case of absence of sampling data is justified by the fact that it would cost the least amount of control energy among all possible control actions (Hadjicostis and Touri 2002), (Imer et al., 2006), (Sadjadi 2003). Of course another way is also possible which consists in applying the "last available control". This last way is equivalent to the zero-order hold (ZOH) action in continuous time but may lead to more control energy.

2.2 Sample Drop Policy

As explained in the former section, the approach considered in this paper to solve network congestion is to selectively drop some instances of systems outputs. Some solutions exist that discard data packets in order to reduce the effective utilization of the system. Among these solutions, the technique often mentioned is *(m,k)*-firm policy (Ramanathan, 1999) (Quan and Hu, 2000). The *(m,k)*-firm model requires that at least m out of any *k* consecutive packets must be delivered by the network. In (Jia et al., 2005) and (Jia et al. 2007) was first studied how to choose the value of *k* and the rate of *m* over *k* so that the system remains stable for mono- and multiple-variable linear systems. The impact of the packet drops distribution on the QoC has been evaluated but without giving a general relationship.

In this paper, we continue to adopt the *(m,k)*-firm based dropping policy and give in section 3 the general analytic relationship between a packet drop sequence and the QoC.

2.3 System Model

The first order plant is described by the discrete time dynamics ($f(ih)$ is noted f_i in the following):

$$x_{i+1} = \alpha x_i + \beta u_i + \beta q_i, \beta \neq 0$$
and
$$y_i = x_i + p_i$$
(1)

where x_i represents the plant state, u_i its input; q_i and p_i, are white noises, assumed to be independent and with zero mean and whose variances will be noted σ_q^2 and σ_p^2. Particularly p_i models the quantization noise in an additive way and q_i represents the actuator noise.

Considering a proportional controller $u_i = \gamma \Pi_i y_i$, where Π_i models if the plant state sample is transmitted to the controller or not ($\Pi_i = 1$ if the ith sample is transmitted and 0 otherwise). Π_i is called *(m,k)*-pattern.

Then the closed loop model is:

$$x_{i+1} = (\alpha + \beta \Pi_i \gamma)x_i + \beta(\Pi_i \gamma p_i + q_i)$$
(2)

We consider that the delay between the time of each plant output sampling and the time of the corresponding actuation instant is always equal to δ (see section 5 for the possible implementation of this assumption). In order to compensate this skew (figure 3), we use a Prediction-Based Model (Marti and Velazco, 2007) that computes the control signal u, with the estimated state vector at the actuation instant $ih + \delta$.
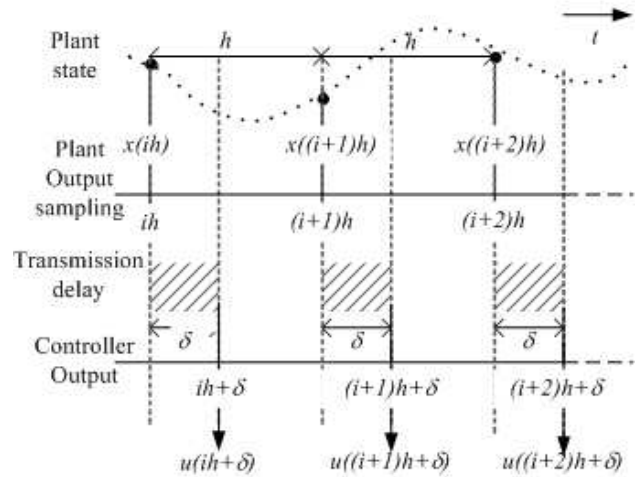

Fig. 3. Impact of the transmission delay.

During the interval (ih , $ih + \delta$), the plant (linear) evolves and the estimated state vector at $ih + \delta$, is $x(ih + \delta) = e^{\alpha \delta} x(ih)$.

If we multiply the terms of the equation (2) by the constant $e^{\alpha \delta}$, we obtain:

$$e^{\alpha \delta} x((i+1)h) = e^{\alpha \delta}(\alpha + \beta \gamma \Pi_i)x(ih)$$
$$+ e^{\alpha \delta}\beta(\Pi_i \gamma p(ih) + q(ih))$$

and the obtained equation shows that:

$$x((i+1)h + \delta) = (\alpha + \beta \gamma \Pi_i)x(ih + \delta)$$
$$+ e^{\alpha \delta}\beta(\Pi_i \gamma p(ih) + q(ih))$$

Then, without loss of generality we can work directly with the model (2) where the noises are $e^{\alpha \delta_c} p_i$ and $e^{\alpha \delta_c} q_i$ (same properties as p_i and q_i).

2.4 Stability conditions and optimisation issues

First we need to identify the condition on the closed loop control that ensures the stability of an unstable plant under output and process noises and subject to a specific dropping strategy of the sampled plant output. As specified in section 2.2, we apply a proportional controller with a gain γ and the transmission of plant state samples are controlled thanks to a

(m,k)-firm policy. For any sequence of *k* samples, only *m* are transmitted according to a predefined *(m,k)*-pattern. So, the packet sequence delivery is *k*-periodic.

From control theory, the control design goal for a system under noise (or cost criterion) is to determine the controller gain which minimizes the variance of the output (see for example, Aström and Wittenmark, 1997). In this paper, we consider an infinite horizon and propose to guarantee the closed-loop stability by calculating the gain γ in order that $\sup_{i} var(x_i)$ should be finite. This property allows taking into account all the future of the system.

Moreover, finding an optimal controller means to determine the parameters of the system that makes this limit as small as possible. More precisely, the optimisation issue consists in finding:

- the parameters of the dropping policy: *m*, *k* and the *(m,k)*-pattern,

- the gain γ of the controller.

The purpose of the next section is to provide a mathematical support for the specification of the parameters.

3. ANALYTICAL APPROACH

In the following, we note $v_i = var(x_i)$. The assumptions made on the noises p_i and q_i, show that:

$$v_{i+1} = a_i v_i + b_i, \ v_0 = 0, \ i \geq 0 \tag{3}$$

where: $a_i = (\alpha + \beta\gamma\Pi_i)^2$ and $b_i = \beta^2(\sigma_q^2 + \gamma^2 \sigma_p^2 \Pi_i)$

We denote $A = a_0 a_1 ... a_{k-1}$. Bearing in mind that $\Pi_i = 0$ for *k-m* terms and $\Pi_i = 1$ for *m* terms, it is readily seen that:

$$A = (\alpha + \beta\gamma)^{2m} \alpha^{2(k-m)}$$

This shows that *A* depends only on *m* and *k* and not of Π.

We set $B_i = b_{i+k-1} + \sum_{j=i}^{i+k-2} b_j a_{j+1}...a_{i+k-1}, 0 \leq i < k$ and $B = \max_{0 \leq i < k} (B_i)$.

In fact, taking account of the *k*-periodicity of a_i and b_i, we have:

$$B_{i+1} = b_{i+k} + \sum_{j=i+1}^{i+k-1} b_j a_{j+1}...a_{i+k}$$

$$= b_i + a_i \sum_{j=i+1}^{i+k-1} b_j a_{j+1}...a_{i+k-1}$$

$$= b_i + a_i (B_i - b_i a_{i+1}...a_{i+k-1})$$

$$= b_i(1-A) + a_i B_i$$

Then B_i can be computed by the recurrence relation:

$$B_{i+1} = a_i B_i + b_i(1-A), \text{ for } i \geq 0,$$

with $B_0 = \sum_{j=0}^{k-1} b_j a_{j+1}...a_{k-1}$.

We obtain the following result.

Theorem 1.

1) If $A \geq 1$, the sequence (v_n) is unbounded.

2) If $0 \leq A < 1$, then the sequence (v_n) is bounded and:

$$\sup_{n \geq 0} v_n = \frac{B}{1-A}$$

For short, the system is stable if and only if $0 \leq A < 1$

Sketch of proof. From equation (3), taking into account the *k*-periodicity of a_i and b_i, it can be seen that for $0 \leq i < k$ and $n \geq 0$:

$$v_{i+(n+1)k} = A v_{i+nk} + B_i \tag{4}$$

1) If $A \geq 1$, the equality (4) shows that $v_{nk} \geq v_0 + nB_0 = nB_0$. We note that $B_0 = \sum_{j=0}^{k-1} b_j a_{j+1}...a_{k-1} \geq b_{k-1} \geq \beta^2 \sigma_q^2 > 0$; therefore, $v_{nk} \geq n\beta^2 \sigma_q^2$ showing that the sequence (v_n) is unbounded.

2) If $0 \leq A < 1$, from equation (4), we can deduce that $v_{i+nk} = \frac{B_i}{1-A} + A^n (v_i - \frac{B_i}{1-A})$, for $0 \leq i < k$ and $n \geq 0$. Therefore $\lim_{n} v_{i+nk} = \frac{B_i}{1-A}$.

It can be shown that $v_i \leq \frac{B_i}{1-A}$ for any *i*. Due to the *k*-periodicity of B_i, we have $v_{i+nk} \leq \frac{B_i}{1-A}$ and therefore $\sup_{n} v_{i+nk} = \frac{B_i}{1-A}$. Finally, $\sup_{n} v_n = \frac{B}{1-A}$.

$\square$

Let us consider now the optimisation problem. For this purpose, given a *(m,k)* dropping policy (m, k, Π), the optimisation problem is to find γ that minimizes $M(\gamma) = \sup_{n \geq 0} v_n(\gamma)$, bearing in mind that

$A = (\alpha + \beta\gamma)^{2m}\alpha^{2(k-m)}$, under the constraint $A < 1$. This is equivalent to:

$$|\gamma - \gamma_0| < \frac{|\alpha|^{1-\frac{k}{m}}}{|\beta|} \quad \text{where } \gamma_0 = -\frac{\alpha}{\beta}$$

A way to evaluate the optimum can be achieved thanks to an iterative algorithm that computes $M(\gamma)$ for γ from

$\gamma_{\min} = \gamma_0 - \dfrac{|\alpha|^{1-\frac{k}{m}}}{|\beta|}$ to $\gamma_{\max} = \gamma_0 + \dfrac{|\alpha|^{1-\frac{k}{m}}}{|\beta|}$ by step equal to d.
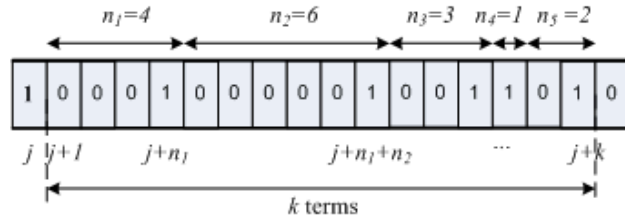
This can be complex because of the computation of B_i. Therefore, we propose a convenient and simple upper bound for $\inf_{\gamma} M(\gamma)$.

Π being given, if $\gamma = \gamma_0 = -\dfrac{\alpha}{\beta}$, then $A(\gamma_0) = 0$.

Corollary .

$$M(\gamma_0) = \frac{\alpha^{2N}}{(\alpha^2 - 1)}(\beta^2\sigma_q^2 + (\alpha^2 - 1)\sigma_p^2) - \frac{\beta^2\sigma_q^2}{(\alpha^2 - 1)}$$

where $N - 1$ is the length of the longest run of 0 in the pattern Π (see Fig. 4).

$N = \max(n1, n2, n3, n4, n5) = 6$

Fig. 4. Example of pattern Π.

We notice that $M(\gamma_0)$ depends only on the longest run of 0 in the pattern Π, N.

Sketch of proof.

It has been seen before that $B_{i+1} = a_i B_i + b_i(1 - A)$ where a_i, b_i, B_i and A depend on γ. Since $A(\gamma_0) = 0$, it turns out that $B_{i+1} = a_i B_i + b_i$. Setting $c = \beta^2\sigma_q^2 + \alpha^2\sigma_p^2$, this relation shows that the sequence is strictly increasing for (see Fig. 4):

- $j \le i < j + n_1$ with $B_j = c$,

- $j + n_1 \le i < j + n_1 + n_2$ with $B_{j+n_1} = c$,

- ...,

- $j + n_1 + ... + n_{m-1} \le i < j + n_1 + n_2 + ... + n_m = j + k$ with $B_{j+n_1+...+n_{m-1}} = c$.

Consequently,

$$M(\gamma_0) = \max_{j \le i < j+k} B_i = \max(B_{j+n_1-1}, B_{j+n_1+n_2-1}, ..., B_{j+k-1}).$$

Denoting $N = \max(n_1, n_2, ..., n_m)$, it is seen that $M(\gamma_0) = x_N$ where x_N is the Nth iteration of the relation $x_{i+1} = \alpha^2 x_i + \beta^2\sigma_q^2$ with the initial condition $x_1 = c$. It is well known that:

$$x_N = (\alpha^2)^{N-1}c + \left(\frac{(\alpha^2)^{N-1} - 1}{(\alpha^2) - 1}\right)\beta^2\sigma_q^2$$

which entails the result.

$\square$

```
/* Calculation of M(γ), for γ, m, k and Π
/* Input data
/*  - α, β : characteristic of the plant
/*  - σ²_p, σ²_q :noises on plant output and input
/*  - γ: controller gain
/*  - m,k,Π
/* Result
/*  - R = M(γ)

M(α, β,σ²_p , σ²_q ,γ, m,k,Π)
begin
  for J in 0..k-1 loop
    a(J) :=( α + β * γ *Π(J)*( α + β * γ * Π(J));
    b(J) :=β * β *( σ²_q + γ * γ * σ²_p * Π(J)) ;
  end loop
  A :=exp(2*m*Log(|α+β*γ|)+2*(k-m)*Log(|α|));
  pa :=1 ;
  B(0) :=b(k-1)*pa ;
  for I in k-2..0 loop
    pa :=pa*a(I+1) ;
    B(0) :=B(0)+b(I)*pa ;
  end loop
  Bmax :=B(0) ;
  for J in 0..k-2 loop
    B(J+1) :=a(J)*B(J)+b(J)*(1-A) ;
    if B(J+1)>Bmax then
        Bmax :=B(J+1)
    end if
  end loop
  R :=Bmax/(1-A);
  return(R);
end
```

Fig. 5. Algorithm for $M(\gamma)$ evaluation.

Obviously, $\inf_{\gamma} M(\gamma) \le M(\gamma_0)$. One the one hand, the corollary shows that it's possible to compute easily a value of γ that respects the stability condition but on the other hand, we would like to know if the bound $M(\gamma_0)$ is tight. This will be investigated in the next section that presents an experiment.

4. NUMERICAL APPLICATION

We apply this approach on a classical case study. We consider a plant described by equation (1) and whose parameters are:

- $\alpha = 3$ and $\beta = 1$,
- under the noises p and q, whose variances are $\sigma_p^2 = 1$ and $\sigma_q^2 = 1$.

In order to avoid a too large dispersion of the plant state, we impose the variance of the plant output to be bounded by a given value $v_{lim} = 1000$. So we have to ensure that the gain γ of the controller is such as $M(\gamma) < v_{lim}$.

The control architecture is distributed as shown in Fig. 1.

We consider that the dropping policy is handled by a dedicated admission controller implemented in the sensor or in a router. The available memory in this device allows representing a pattern whose size is less or equal to 10 ($k \le 10$).

Furthermore, we impose to limit the network bandwidth used by this application by ensuring $\frac{1}{k} \le \frac{m}{k} \le 0.5$. The computation of $M(\gamma) = \sup_{n \ge 0} v_n(\gamma)$ is done thank to the algorithm presented in Fig.5.

The result we obtained is given in Table 1. The two first columns give the (m,k) constraint: m, k, and the pattern Π. The third and fourth columns provide γ_{opt} that minimises the limit of the plant output variance for this constraint and $M_{opt} = M(\gamma_{opt})$ (see Fig. 6 for the algorithm providing γ_{opt}). The column 5 contains $M(\gamma_0)$ and in order to estimate, for this case study, if γ_0 is a tight upper bound of γ_{opt}, we provide two information: the first one ($n_<$) gives the number of γ tested in $[\gamma_{\min}, \gamma_{\max}[$ such as $M(\gamma) < M(\gamma_0)$, the second one (ε) furnishes an indicator on the relative distance between $M(\gamma_0)$ and $M(\gamma_{opt})$ and is evaluated as $\varepsilon = \dfrac{M(\gamma_0) - M(\gamma_{opt})}{M(\gamma_{opt})}$.

```
/* Calculation of  γ_opt such as M( γ_opt )= inf M( γ )
                                             γ
/* -Input data
/*  * α, β : characteristic of the plant
/*  * σ_p², σ_q² :noises on plant output and input
/*  * γ: controller gain
/*  * limits on k, m/k and v_lim
/* -Result
/*  * γ_opt, M( γ_opt ), M( γ_0 ) for each admissible Π
begin
  for each (m,k) such as k = 2..10 , m/k ≤ 0.5   loop

    γ_min = γ_0 − |α|^(1−k/m)/|β| ;   γ_max = γ_0 + |α|^(1−k/m)/|β|
    d = ( γ_max − γ_min )/100 ;
    γ_opt = γ_min ; M_opt = M( γ_opt ) ;
    for each g in [ γ_min +d ... γ_max  [ , step d loop
      M= M( γ ) ;
      if (M< v_lim ) then
        if (M< M < M_opt ) then
          M_opt = M ; γ_opt = γ ;
        end if
      end if
    end loop
    print(m,k,Π,  γ_opt , M_opt , M( γ_0 ));
  end loop
end
```

Fig. 6. Evaluation of $\gamma_{opt}, M(\gamma_{opt}), M(\gamma_0)$ for each (m,k) constraint that is admissible.

The smallest limit of the plant state variance is obtained for a $(1,2)$ constraint whose pattern is $\Pi = "10"$. In this case, the best gain is $\gamma_{opt} = -2{,}966667$ and $M(\gamma_{opt}) = 90{,}11$. Note that, for this pattern, on 100 regularly spaced values of γ, 10 of them are such that $M(\gamma) < M(\gamma_0) = M(-3) = 91$. Fig. 7 shows the curve $M(\gamma)$ for this configuration.

The largest limit of the variance corresponds to a $(5,10)$ pattern, $\Pi = "1110100100"$. The optimal gain is $\gamma_{opt} = -2{,}9933$ and $M(\gamma_{opt}) = 819{,}70$. $M(\gamma_0) = 820$. Only 2 values of γ are such as $M(\gamma) < M(\gamma_0)$ and ε is smallest than $0{,}00001$ (see Fig. 8).

For this example, the value of $\gamma_0 = \dfrac{-\alpha}{\beta} = -3$ appears to provide a tight upper bound of $\inf_{\gamma} M(\gamma)$.

Table 1. γ_{opt} , $M(\gamma_{opt})$ **according to** m, k, Π

(m,k)	Π	γ_{opt}	M_{opt}	$M(\gamma_0)$	$n_<$	ε
(1, 2)	10	-2,9667	9,01E+01	9,10E+01	10	1,0%
(1, 3)	100	-2,9956	8,19E+02	8,20E+02	3	0,1%
(2, 4)	1100	-2,8933	7,86E+02	8,20E+02	26	4,3%
(2, 5)	10100	-2,9692	8,13E+02	8,20E+02	15	0,9%
(3, 6)	110100	-2,9667	8,12E+02	8,20E+02	11	1,0%
(3, 6)	110010	-2,8600	7,75E+02	8,20E+02	32	5,8%
(3, 7)	1100100	-2,9954	8,19E+02	8,20E+02	2	0,1%
(3, 7)	1010100	-2,9676	8,12E+02	8,20E+02	14	1,0%
(3, 8)	10100100	-2,9968	8,19E+02	8,20E+02	3	0,1%
(4, 8)	11100100	-2,9933	8,20E+02	8,20E+02	2	0,1%
(4, 8)	11010100	-2,9667	8,12E+02	8,20E+02	10	1,0%
(4, 8)	11010010	-2,9667	8,12E+02	8,20E+02	11	1,0%
(4, 8)	11001010	-2,8400	7,71E+02	8,20E+02	35	6,4%
(4, 9)	110100100	-2,9949	8,19E+02	8,20E+02	1	0,1%
(4, 9)	110010100	-2,9696	8,12E+02	8,20E+02	10	0,9%
(4, 9)	110010010	-2,9949	8,19E+02	8,20E+02	1	0,1%
(4, 9)	101010100	-2,9696	8,12E+02	8,20E+02	12	1,0%
(4,10)	1100100100	-2,9962	8,19E+02	8,20E+02	2	0,1%
(4,10)	1010100100	-2,9962	8,19E+02	8,20E+02	2	0,1%
(5,10)	1110100100	-2,9933	8,20E+02	8,20E+02	2	0,1%
(5,10)	1110010100	-2,9733	8,13E+02	8,20E+02	9	0,9%
(5,10)	1110010010	-2,9933	8,20E+02	8,20E+02	2	0,1%
(5,10)	1101100100	-2,9933	8,20E+02	8,20E+02	2	0,1%
(5,10)	1101010100	-2,9667	8,12E+02	8,20E+02	10	1,0%
(5,10)	1101010010	-2,9667	8,12E+02	8,20E+02	10	1,0%
(5,10)	1101001100	-2,9667	8,12E+02	8,20E+02	11	1,0%
(5,10)	1101001010	-2,9667	8,12E+02	8,20E+02	11	1,0%
(5,10)	1100110010	-2,8933	7,86E+02	8,20E+02	27	4,3%
(5,10)	1100101010	-2,8333	7,69E+02	8,20E+02	37	6,6%

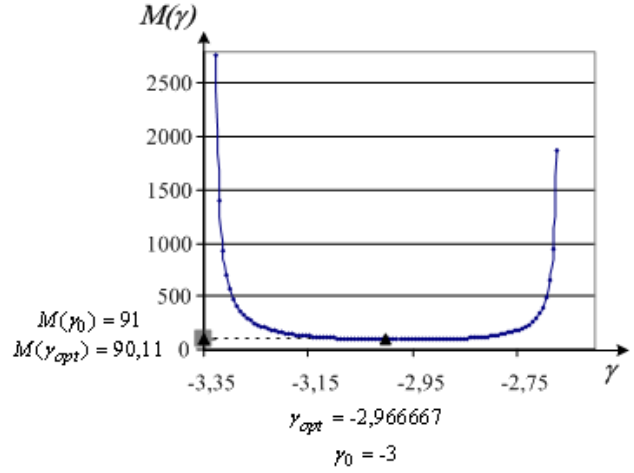condition should also be implemented to know the condition to decide the packet dropping.

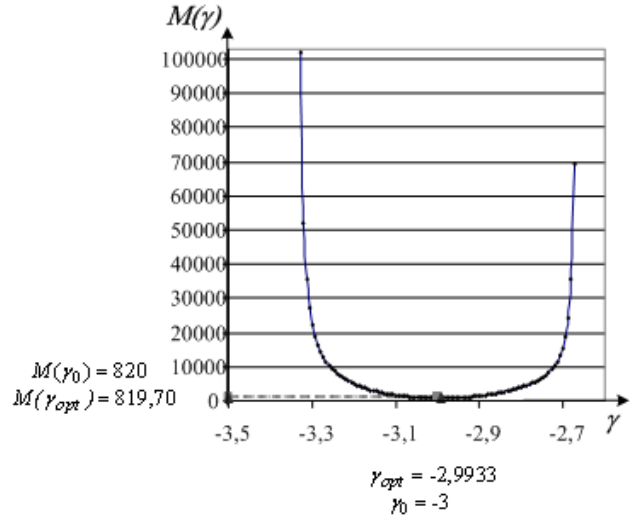Fig. 7. $M(\gamma)$ under a (1,2) constraint ($\Pi = "10"$).

Fig. 8. $M(\gamma)$ under a (5,10) constraint ($\Pi = "1110100100"$).

5. IMPLEMENTATION ISSUES

As stated above, the goal of our communication and control co-design approach is to optimize both the network bandwidth utilization and the QoC.

The approach we proposed consists in selectively dropping the sampling data packets to reduce the network load while still maintain an acceptable QoC for the application. In fact, as we have shown in section 2 and 3, for a given sampling period, the control loop can tolerate a certain consecutive data packet drops, i.e. the longest run of *0* in the *(m,k)*-pattern. To achieve the selective data dropping in case of network overload, a dropping mechanism should be implemented either in the network and/or at sender nodes (e.g. the sensor node in our control loop), and a trigger

For the synchronous system considered in Fig. 2, a simple way to if the network is in overload situation or not is the use of a timer at the sender node (sensor) by setting the timeout value to $\delta - \tau_{sc}$. Where τ_{sc} is the expected packet transmission delay from the sensor node and the controller node plus a secure margin (e.g. mean value + the variance). This delay can be either estimated on-line (similar to the round trip time estimation in TCP) on fixed off-line when it is possible. If a sample data packet is not transmitted in time (i.e., before $\delta - \tau_{sc}$), it indicates that the network is in congestion and we should reduce the workload by sending the sampling data packet according to a pre-configured (m,k)-pattern (could be implemented using a *k*-bits register). The system passes thus from the normal mode to a degraded mode. The system can try to return back to the normal mode after a fixed period or

by observing that the delay before sending the data packet is much lower than $\delta - \tau_{sc}$.

6. CONCLUSIONS

In this paper we decided to choose a simple mono-variable linear system with a proportional controller to illustrate the approach of control and communication co-design. We assumed "zero control" action in case of sampling data drop. So for dealing with network congestion we proposed to selectively drop sampling data according to *(m,k)*-firm model.

Following new results are presented: 1) We proved (Eq. 3) that the network-induced delay can be incorporated into the discrete-time system model under our "zero control" action assumption, allowing thus the system stability analysis by only considering the data drops. 2) When periodic dropping pattern is applied to the system output sampling data, classic plant state variance does not converge to a value. In this case the upper bound of the variance can be used instead. 3) We derived that the system stability (upper bound of the variance) only depends on the value of *m* and *k* but not *(m,k)*-pattern (theorem 1). Moreover, we identified a gain γ_0 that can easily be computed (even on line) and that provides a tight approximation of the best upper bound of the variance; this gain only depends on the longest run of *0* in the *(m,k)*-pattern (corollary).

Although a very simple system has been chosen in this paper, it is worth noting that the principle of this approach can also be applied to multi-variable linear systems. For instance in (Jia et al. 2007) and (Felicioni et al. 2006), a two-variable system has been considered and ZoH has been assumed, the stability as well as the QoC under (m,k)-firm dropping policy have been investigated using LQ cost function. The results are also encouraging.

Our future work aims at extending this approach to handling multiple control loops sharing a same network (wired or wireless) where the design of packet scheduling and selective dropping should significantly contribute to optimizing the resource utilization while still keeping the system stable and with a graceful QoC degradation during network congestion situations.

REFERENCES

Antsaklis P. and J. Baillieul (editors), "Technology of betworked control systems", *Proceedings of the IEEE, Jan. 2007*, VolL.95, No.1.

Antunes A., P. Pedreiras, L. Almeida and A. Mota, "Dynamic rate and control adaptation in networked control systems", *5th International Conference on Industrial Informatics*, Vol 2, June 2007, Vienna, Austria.

Aström, J. and Wittenmark, B., (1997). *Computer-controlled Systems: Theory and Design*. Prentice-Hall International. Inc, second edition.

Bommannavar P. and T. Basar, "Optimal control with limited control actions and lossy transmissions", *47th IEEE CDC*, pp2032-2037, Cancun, Mexico, Dec. 9-11, 2008.

Eker J., P. Hagander, and K. E. Årzén, "A feedback scheduler for real time control tasks", *Control Engineering Practice*, pages 1369-1378, 2000.

Felicioni F., N. Jia, Y.Q. Song and F. Simonot-Lion. "Impact of a (m,k)-firm Data Dropouts Policy on the Quality of Control". *6th IEEE International Workshop on Factory Communication Systems*, Turin, Italy, 2006.

Hadjicostis C. and R. Touri Feedback Control Utilizing Packet Dropping Network Links Proc. of the *IEEE Conference on Decision and Control*, 2002, pp: 1205—1210.

Halevi, Y., & Ray, A., 1988. Integrated communication and control systems: Part I—analysis. *Journal of Dynamic Systems, Measurement, and Control* 367–373.

Imer O. C., S.Yükselb y T. Başarb. Optimal control of LTI systems over unreliable communication links. *Automatica*. 42: 1429 – 1439 (2006).

Jia N, Y.Q. Song and F. Simonot-Lion, "Graceful Degradation of the Quality of Control through Data Drop Policy", *European Control Conference 2007 (ECC'07)*, Kos, Greece, July 2007.

Jia N, Y.Q. Song and R.Z. Lin. "Analysis of Networked Control System with Packet Drops Governed By (m,k)-firm Constraint". The *6th IFAC FET'05*, Mexico, 2005.

Juanole G. and G. Mouney, "Using an hybrid traffic scheduling in networked control systems", Proceedings of *European Control Conference 2007*, Kos, Greece, July 2007.

Liou, L. W. and Ray, A. A stochastic regulator for integrated communication and control systems: II-formulation of control law", Journal of the ASME , 113: 612–619, 1991.

Martí P., M. Velasco (2007) Toward Flexible Scheduling of Real-Time Control Tasks: Reviewing Basic Control Models. *HSCC 2007*: 710-713.

Quan, G., and X. Hu, Enhanced Fixed-priority Scheduling with (m,k)-firm Guarantee, Proc. Of *21st IEEE Real-Time Systems Symposium*, pp.79-88, Orlando, Florida, (USA), November 27-30, 2000.

Ramanathan P., "Overload management in Real-Time control applications using (m,k)-firm guarantee", *IEEE Transactions on Parallel and Distributed Systems*, 10(6) :549-559, Jun 1999.

Sadjadi B. Azimi, Stability of networked control systems in the presence of packet losses, Proceedings of the *42nd IEEE conference on decision and control* (2003), pp. 676–681.

Seto D., J. P. Lehoczky, L. Sha and K. G. Shin. "On task schedulability in real-time control system", Proceedings of the *17th IEEE Real-Time Systems Symposium*, pages 13-21, Washington, DC, Dec 1996.

Simon D., D. Robert, and O. Sename, "Robust control/scheduling co-design : application to robot control", In *RTAS'05*, San Francisco, CA, USA, Mar 2005.