

## Gain-Scheduling Multi-Bit Delta-Modulator for Networked Controlled System

Ivan Lopez, C. T. Abdalla, Carlos Canudas de Wit

► **To cite this version:**

Ivan Lopez, C. T. Abdalla, Carlos Canudas de Wit. Gain-Scheduling Multi-Bit Delta-Modulator for Networked Controlled System. European Control Conference, ECC'07, Jul 2007, Kos, Greece. pp.s/n. hal-00201789

**HAL Id: hal-00201789**

**<https://hal.archives-ouvertes.fr/hal-00201789>**

Submitted on 2 Jan 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Compensation Schemes for a Delta-Modulation-Based NCS

I. Lopez\*, C.T. Abdallah\*\*

Electrical & Computer Engineering Department  
MSC01 1100

1 University of New Mexico  
Albuquerque, NM 87131-0001

{ilopez, chaouki}@ece.unm.edu

C. Canudas de Wit

Laboratoire d'Automatique de Grenoble  
UMR CNRS 5528  
ENSIEG-INPG

B.P. 46, St. Martin d'Hères 38402 France

carlos.canudas-de-wit@inpg.fr

**Abstract**— We analyze in this paper stabilization issues for a Networked Control System that uses a Delta-Modulator Scheme within the encoder/decoder structures. We also analyze the packet-loss issue, and determine a maximum allowable number of consecutive bits lost while keeping stability. We then design a compensation scheme for re-synchronizing the encoder and decoder, after a bit is lost in a network without acknowledgment signals. We finally present a compensation scheme that ensures stability after a pre-determined number of bits is lost. Examples and simulations are provided to demonstrate the results.

## I. INTRODUCTION

In recent years, much research and development have been expanded in the Networked Control Area. Because of the absence of dedicated communication protocols for control purposes, new issues arise and new tools are needed to deal with networked control systems. In particular, the communication channel between the plant and the controller may no longer remain unmodelled, since it can only carry data at a finite rate, and the conventional assumption of infinite capacity of the channel no longer holds. In addition to suffering from both packet losses and quantization effects, the finite data rate limitation forces us to determine the usefulness of the number of bits [7].

Several recent papers have focused on the quantization problem caused by the limited channel rate, see for instance [12], [6], [11], [4], [9] and [1]. Such research has focused on developing a new theory of control under communications constraints. Other researchers have focused their efforts on implementing low data rates systems for control purposes. Specially, in [2] a differential coding with a Delta-Modulation ( $\Delta - M$ ) scheme was used since such a scheme provides the simplest form of differential coding. This translates into a low cost design, since the  $\Delta - M$  algorithm is a simple two-level dynamic quantizer.

Let us recall (see for example [4] and [11]), the minimum

\* The research of this author is partially supported by CoNaCyt.

\*\* The research of this author is partially supported by NSF award CNS 0626380 under the FIND initiative.

required rate for stabilization is given by

$$R > \sum_{i=1}^n \log_2(\lambda_i) \quad (1)$$

where  $\lambda_i$  are the eigenvalues of an open-loop linear system. Thus, if we consider the scalar discrete-time systems, the  $\Delta - M$  algorithm is limited to stabilize linear systems of the form

$$x(k+1) = ax(k) + bu(k) \quad (2)$$

with  $|a| \leq 2$ . In this paper, we first analyze modifications for the  $\Delta - M$  algorithm proposed by [2] in order to stabilize systems with  $|a| > 2$ . This is motivated by the previously mentioned cost issues associated with the simple  $\Delta - M$  schemes.

We also consider the packet-dropping problem to analyze the recovery of synchronization and stabilization issues. This analysis is innovative since previous work on the subject have dealt with the limited-rate and the packet losses separately. Recent works considered packet losses but assumed unlimited channel rate, see [3] and [8], while research dealing with limited-rate channels have not included packet losses. Few authors have considered both packet loss and limited capacity. In one notable exception, the authors derived the minimum channel capacity when an erasure channel is present, see [10].

Our results show that the maximum number of packets that can be lost sequentially depends on the region where a certain estimation error  $\tilde{x}_k = x(k) - \hat{x}(k)$  lives. Using this fact, we redesign the  $\Delta - M$  scheme used in [2] so the system can handle at least a minimum number of packet losses.

We emphasize that while we consider scalar systems in our current analysis, the generalization to non-scalar systems will be part of our future work.

## II. PROBLEM SETUP

We consider the same system described in [2]. To get a better understanding of our proposed scheme, we first analyze the details and limitations of the original scheme. In what follows, we assume that:

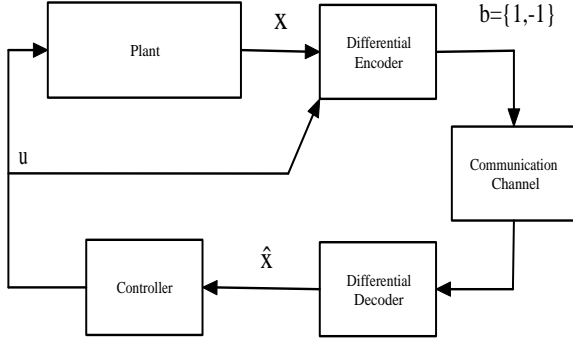


Fig. 1. Closed loop system

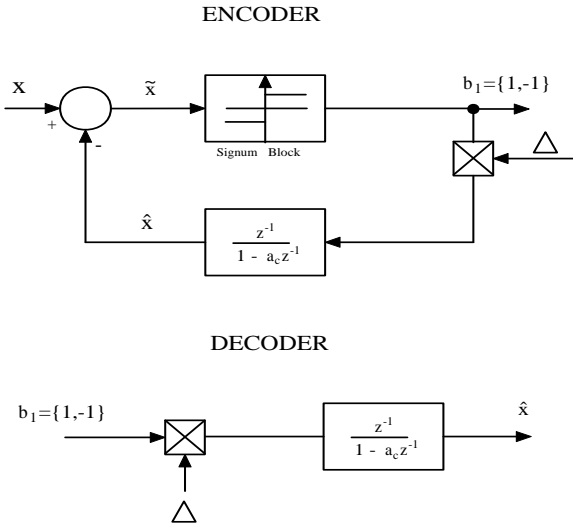


Fig. 2. Closed loop system

- The transmitted bit at time  $k$  is  $b_k \in \{-1, 1\}$ .
- Only encoder-to-decoder information transmission is allowed.
- No packets are lost (this assumption will be relaxed later).
- All the elements in the loop, including the communication channel, are noiseless.

The block diagram used in this work is shown in figure (1).

The plant is modeled as a scalar discrete linear time invariant system,

$$x(k+1) = ax(k) + bu(k) \quad (3)$$

$$u(k) = -k\hat{x}(k). \quad (4)$$

In the original  $\Delta-M$  scheme of [2], the encoder and decoder shown in figure (2) may be described by:

$$\hat{x}(k+1) = a_c \hat{x}(k) + \Delta \text{sgn}(\tilde{x}(k)) \quad (5)$$

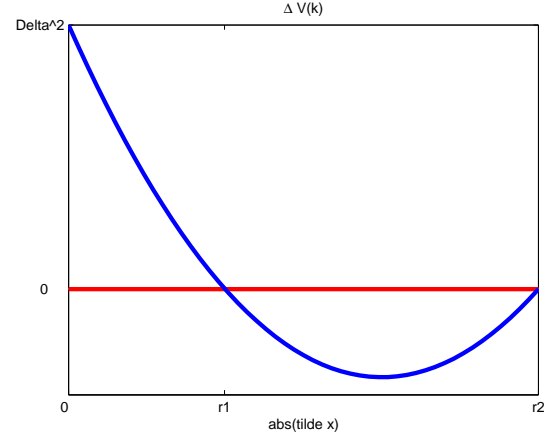


Fig. 3.  $\Delta V(k)$

Combining equations (3) and (5) we obtain the following dynamics for the system and the error:

$$\begin{aligned} x(k+1) &= a_c x(k) + b k \tilde{x}(k) \\ \tilde{x}(k+1) &= a \tilde{x}(k) - \Delta \text{sgn}(\tilde{x}(k)) \end{aligned}$$

Therefore, the stability of the overall system may be analyzed from that of the coding error  $\tilde{x}(k)$ . When  $V(k) = \tilde{x}^T \tilde{x}$  is chosen as a Lyapunov function candidate, [2] shows that the change  $\Delta V(k)$  is given by

$$\Delta V(k) = \begin{cases} \leq 0 & \text{if } |\tilde{x}(k)| \leq r_1 \\ < 0 & \text{if } r_1 < |\tilde{x}(k)| < r_2 \\ \geq 0 & \text{if } r_2 \leq |\tilde{x}(k)| \end{cases} \quad (6)$$

where  $r_1 = \frac{\Delta}{a+1}$  and  $r_2 = \frac{\Delta}{a-1}$ .

The region where  $\Delta V(k) < 0$ , i.e.,  $r_1 < |\tilde{x}(k)| < r_2$  will be denoted by  $R_1$ . We already know from (1) that the rate of the channel will limit the absolute value of  $a$ , but it is important to study what happens if we try to stabilize systems with  $|a| > 2$ . This analysis will provide the impetus for redesigning the Delta Modulator Scheme.

### III. STABILITY ANALYSIS FOR SYSTEMS WITH $a > 2$ .

In [2] it was shown that the expansion of  $\Delta V(k)$  is given by the following equation:

$$\Delta V(k) = (a^2 - 1) \tilde{x}^2(k) - 2a\Delta \text{sgn}|\tilde{x}(k)| + \Delta^2 \quad (7)$$

If we study the plot of this function in figure (3) in the range  $0 \leq |\tilde{x}(k)| \leq r_2$ , we notice that the maximum value of  $\Delta V(k)$  occurs at  $|\tilde{x}(k)| = 0$ . Now, if for some  $k_f$  we have  $|\tilde{x}(k_f)| = 0$ , then  $\tilde{x}(k_f + 1) = -\Delta$  and this may land  $\tilde{x}(k_f + 1)$  outside the region  $R_1$ . Once there, the error state will not return to the region of attraction since the Lyapunov function outside  $R_1$  is increasing.

Let us examine carefully when such an event takes place. We know that  $r_2 = \Delta/(a-1)$ , so that if  $\Delta > \Delta/(a-1)$ , the error  $\tilde{x}(k)$  will be ejected from the region of  $r_1 < |\tilde{x}(k)| < r_2$  and can never return to it. We then note that is  $|a| > 2$

when the inequality  $\Delta > \Delta/(a-1)$  is satisfied. Therefore, for  $|a| > 2$  it is not possible to stabilize the system as predicted by the minimum rate given by equation (1). Although this result was expected, this analysis allows us to think of ways to solve the problem for systems with  $|a| > 2$ .

#### IV. COMPENSATION SCHEME

To overcome the limitations imposed by the scheme of the previous section, we obviously need to increase the bit-rate in the closed-loop. This can be done either by sending the bits in less time (i.e. by increasing the sampling time) or by increasing the number of bits in the same time period. Let us analyze each of these two options separately.

##### A. Increasing sampling time

Let us assume that the discrete-time linear system was obtained by discretizing a continuous-time system. The continuous-time, linear, time-invariant system has the form

$$\dot{x} = fx + gu \quad (8)$$

Then, the discretized system is given by:

$$x(k+1) = e^{fT}x(k) + g \int e^{a\eta} d\eta u(k) \quad (9)$$

If we allow the sampling time to be increased, we can move the discrete pole to the desired position  $1 \leq a \leq 2$ . This will be accomplished if

$$e^{fT} < 2, \text{ or equivalently } T < \frac{\ln(2)}{f}$$

##### B. 2-Bit-Delta-Modulation Scheme

On the other hand, if we can not increase the sampling time, to overcome the problem of increasing the rate, we may try sending more information (bits) across the channel. In [4], it was shown that using a DPCM communication scheme (that is a generalization of the Delta Modulation Scheme) instead of a 1-bit quantizer may solve the problem. Thus, a multilevel quantizer is used, where the number of levels is determined by the relation  $r_{min} = \log_2(a)$ .

Here, a different idea is proposed to conserve the general structure of a Delta modulator. We propose to exchange the single modulator,  $\Delta_1$ , for 2 modulators,  $\{\Delta_1, \Delta_2\}$  as shown in figure (4).

The idea behind our proposed scheme is to use a comparator that determines whether we are inside or outside the region  $|\tilde{x}(k)| < r_2$ . If we are inside this region, we use the  $\Delta_1$  modulator, and switch to the other modulator,  $\Delta_2$ , once outside. The second modulator allows to increase the region of attraction as will be explained latter. The **sign** function will give us one bit, and the comparator will give us one extra bit that will allow us to handle more instability in the system, i.e. to stabilize systems whose eigenvalues are greater than 2 in magnitude.

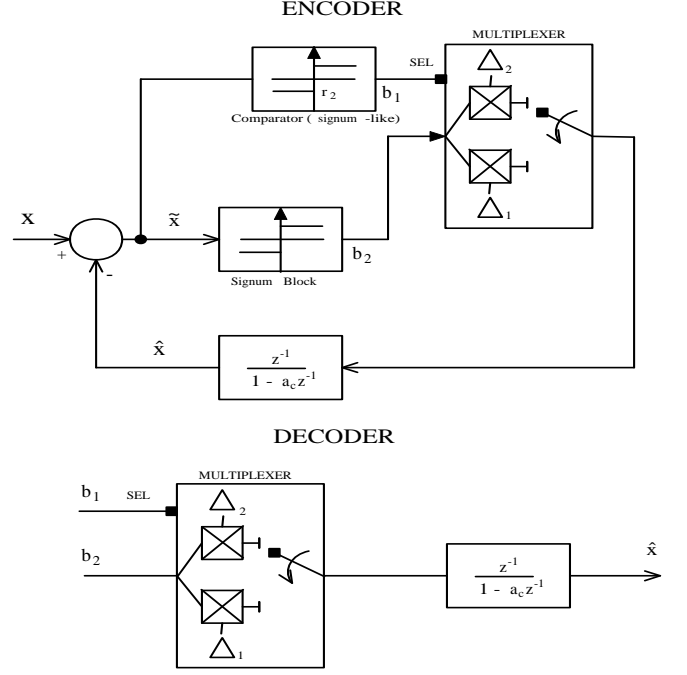


Fig. 4.  $\Delta V(k)$

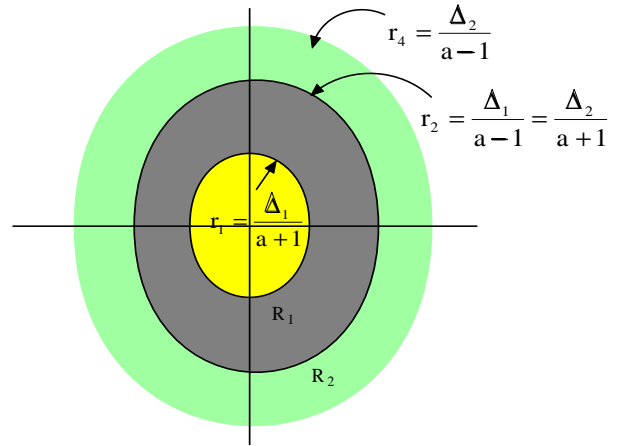


Fig. 5.  $\tilde{x}(k)$  region

#### V. DESIGN OF A 2-BIT-DELTA-MODULATION SCHEME

Let us explain in greater detail the proposed 2-Bit-Delta-Modulator. First, we build a Delta Modulator similar to [2]. This will give the region  $R_1$  where  $\Delta V(k) < 0$ . As shown before, the problem for systems with  $|a| > 2$  arises when  $|\tilde{x}(k)| = 0$  since  $\tilde{x}(k+1)$  will be strictly outside the region  $R_1$ . That is exactly where the second modulator is activated. This second modulator is chosen with a  $\Delta_2$  that creates a second region in which  $r_3 < |\tilde{x}(k)| < r_4$  and where we enforce  $r_2 = r_3$ . We denote this second region by  $R_2$  as shown in figure (5). This selection will imply that  $\Delta_2 > \Delta_1$ .

**Remark 5.1:** Although we are dealing with an scalar system, in figure (5) we chose two-dimensional balls, to

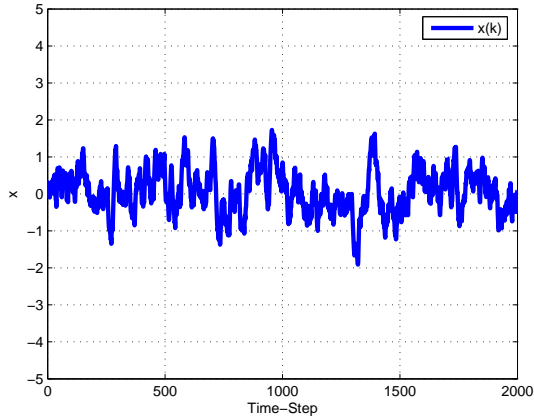


Fig. 6. NCS response with 2-Bit-Delta Modulation Scheme and  $a = 2.15 \leq 3$

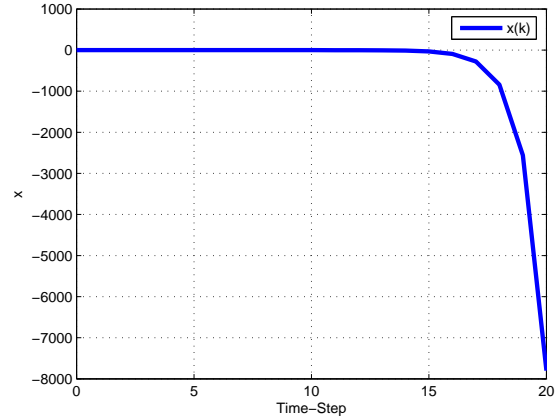


Fig. 7. NCS response with 2-Bit-Delta Modulation Scheme and  $a = 3.05 > 3$

better illustrate the proposed concepts.

Reviewing our Lyapunov analysis, we know that if the initial condition is such that  $|\tilde{x}(0)| < r_2$  then  $\Delta V(k) < 0$ . There will however be a moment when  $|\tilde{x}(k)|$  will be less than  $r_1$  and, eventually, when it is near 0 it will be ejected out of  $r_2$  if  $|a| > 2$ . The comparator in figure (4) then provides the signal to start using the second modulator, which will make  $\Delta V(k) < 0$  in the region  $r_3 \leq |\tilde{x}(0)| \leq r_4$ . To force  $r_2 = r_3$  we enforce

$$\Delta_2 = \frac{a+1}{a-1} \Delta_1. \quad (10)$$

With this relation we know that the region where  $\Delta V(k)$  is negative will be within  $r_1 < |\tilde{x}| < r_4$  where  $r_3 = \frac{1}{a+1} \Delta_2$ ,  $r_2 = \frac{1}{a-1} \Delta_1$ ,  $r_3 = r_2$  and  $r_4 = \frac{1}{a-1} \Delta_2 = \frac{a+1}{(a-1)^2} \Delta_1$ .

Now, if we analyze the case where  $\Delta V(k)$  reaches its maximum, i.e., when  $|\tilde{x}(k)| = 0$ , we see that  $|\tilde{x}(k+1)| = \Delta_1$ . If we compare this jump with the border  $|\tilde{x}(k)| = r_2 = \Delta_1 \frac{a+1}{(a-1)^2}$  we see that

$$\Delta_1 > \Delta_1 \frac{a+1}{(a-1)^2} \quad (11)$$

$$a > 3. \quad (12)$$

In other words, by modifying the modulation scheme and using a second bit, we can now have handle systems with an unstable eigenvalue  $|a| \leq 3$ . Note however that the 2-bit  $\Delta$ - $M$  modulation scheme is inefficient, since with 2 bits we should be able to handle eigenvalues  $|a| \leq 4$  as established in equation (1).

#### A. Example 1

To test the modified scheme we present the following system that cannot be stabilized with the original Delta-Modulation scheme of [2].

$$\begin{aligned} x(k+1) &= 2.1x(k) + u(k) \\ u(k) &= -1.15\hat{x}(k) \end{aligned}$$

Assume that  $\Delta = 0.2$ ,  $x_0 = 0.12$ . Therefore,  $r_1 = 0.0645$ ,  $r_2 = 0.1818$ ,  $r_4 = 0.5124$  and  $r_1 < |\tilde{x}_0| < r_2$ . In the simulation shown in figure (6) we see that the system has been stabilized.

#### B. Example 2

We see now that as predicted for  $|a| > 3$  the system can no longer be stabilized by the 2-Bit-Delta-Modulation. The system that was simulated is

$$\begin{aligned} x(k+1) &= 3.05x(k) + u(k) \\ u(k) &= -2.1\hat{x}(k) \end{aligned}$$

The result is shown in figure (7).

## VI. PACKET-LOSSES ISSUE

The previous sections dealt with a communication channel that does not suffer any packet losses. This however is a common problem in packet-based networks, and is dealt with using a variety of approaches (re-transmission for example). In our particular case when we talk about packet losses, we mean bit losses since the information send through the channel is a bit and not a set of bits with header (commonly known as packet). One question that naturally arises is how robust is the Delta-Modulation scheme in the face of lost bits. By lost bits we study the case where the bits transmitted from the encoder through the channel does not reach the decoder.

For our analysis, we will consider a UDP-like channel, i.e., we do not allow any acknowledgement packet flowing from the decoder back to the decoder. The reason for this choice is that UDP-like channels have been preferred by several experiments to avoid long, and potentially destabilizing delays, see [?] and [5].

The first effect of dropping a bit, even before considering its impact on stability and performance, is the loss of synchronization between the encoder and decoder. Recalling

(see equation (5)) that both encoder and decoder use a predictor that updates with the bit transmitted. Therefore, a dropped bit will cause the encoder and decoder to lose synchronization, since there is no acknowledgment signal that allows the encoder to know that a bit has not reached its intended destination.

We note here that in our considered networked control system configuration, the plant and encoder are physically collocated and as such, the encoder has access to the control input,  $u(k)$ , every time. This fact has been pointed in [11] to be enough for conserving the equimemory property of the encoder and decoder, i.e, it can guarantee their synchronization, even in the absence of an acknowledgement signal. To use this advantage we have to modify the structure of the encoder and decoder with some extra logic. We then introduce a new notation:  $\hat{x}_e$  for the encoder estimator and  $\hat{x}_d$  for the decoder estimator and, if there is no packet losses, their dynamics will be given by

$$\hat{x}_e(k+1) = a_c \hat{x}_e(k) + \Delta \text{sgn}(\tilde{x}_e(k)) \quad (13)$$

for the encoder and

$$\hat{x}_d(k+1) = a_c \hat{x}_d(k) + \Delta \text{sgn}(\tilde{x}_e(k)) \quad (14)$$

for the decoder, where the  $\text{sgn}()$  function in both equations is actually the transmitted bit  $b$ .

Let us suppose that at some instant,  $k$ , the transmitted bit is lost. The encoder prediction will evolve following equation (13). However, since the bit with the information of  $\text{sgn}(\tilde{x}_e(k))$ , never reaches the decoder, the decoder estimator will not follow anymore equation (14) and will evolve differently than the encoder following the next equation

$$\hat{x}_d(k+1) = a_c \hat{x}_d(k) \quad (15)$$

Therefore, the control law that will be applied in the next sampling instant is given by

$$u(k+1) = -k \hat{x}_d(k+1) \quad (16)$$

So the synchronization between encoder and decoder is lost and has to be recovered. According to the approach given in [10], we propose to modify the encoder as follows. At instant  $k+1$ , the encoder compares the control signal  $u$  that was received from the controller, with the expected control signal  $u_e = -k \hat{x}_e$ . If there were no lost bits,  $u$  will be equal to  $u_e$ , since the estimates  $\hat{x}_e$  and  $\hat{x}_d$  will be equal. However, if the bit is lost, then  $u(k)$  will be different from  $u_e(k)$  and that will trigger a reset action for the encoder estimator. The reset action will consist on the following: before the encoder generates the next estimation,  $\hat{x}_e(k+1)$ , it will replace the current value of  $\hat{x}_e(k)$  (which was previously calculated using the whole information that did not arrive to the decoder) by the value given by  $a_c \hat{x}_e(k-1)$ . This replacement expression has the same value that the decoder calculated previously because of the bit lost. The idea works because before the first bit is lost the estimations  $\hat{x}_e$  and  $\hat{x}_d$  are equivalent. Then,

after doing the replacement, the encoder calculates the next prediction  $\hat{x}_e(k+1)$ . In summary, both encoder and decoder will be once more in synchronization and proceed thereafter considering a fact that  $\hat{x}_e = \hat{x}_d$  and denote it as  $\hat{x}$ .

We have to comment that we are assuming noiseless elements between the controller and the actuator so we can use the equality between  $u$  and  $u_e$  without major concerns. We can add, however, some robustness in the case of the presence of some noise between the controller and actuator by considering the following compensation.

When a packet is lost we can note from equations (14) and (15), that the difference between the expected signal  $u_e$  and the received  $u$  is given by  $|k\Delta|$ . So we can place a threshold in the comparison of  $u_e$  and  $u$ : if  $|u - u_e| < \left|\frac{k\Delta}{2}\right|$  we consider them as equal, i.e., the packet arrived to the decoder. If  $|u - u_e| > \left|\frac{k\Delta}{2}\right|$  then we consider that the packet was lost. Under this logic, any additive noise with magnitude strictly less than  $\left|\frac{k\Delta}{2}\right|$  will not cause problems.

While this solves the synchronization problem when bits are lost, we have yet to analyze what happens to the stability of the closed-loop system. When one bit is lost, we need to alter the prediction form of the encoder as explained above in order to regain synchronization. This however implies that the error  $\tilde{x}(k)$  is not longer given by  $\tilde{x}(k+1) = a\tilde{x}(k) - \Delta \text{sgn}(\tilde{x}(k))$ , but instead by:

$$\begin{aligned} \tilde{x}(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= ax(k) - bk\hat{x}(k) - a\hat{x}(k) - bk\hat{x}(k) \\ &= a\tilde{x}(k) \end{aligned} \quad (17)$$

This last expression can be easily generalized to  $l$  consecutive bit losses as

$$\tilde{x}(k+1) = a^l \tilde{x}(k) \quad (18)$$

We know, from section II that the stability region is limited on the outside by  $r_2 = \frac{\Delta}{a-1}$ . This limit allows us to determine the number of consecutive bits that may be lost before losing stability. In fact, by setting  $|\tilde{x}(k)| \leq r_2$ , from equation (18) and the expression for  $r_2$  we obtain

$$l \leq \left\lfloor \frac{\log_2 \left( \frac{\Delta}{(a-1)|\tilde{x}(k)|} \right)}{\log_2(a)} \right\rfloor \quad (19)$$

where  $\lfloor \cdot \rfloor$  is the floor function. We see that the allowable number of lost future bits depends on the current error  $\tilde{x}(k)$ . This implies that there is a region within the region  $R_1$  that does not allow ANY bit losses. This region is given by  $\tilde{x}(k) > \frac{\Delta}{a(a-1)}$ . In summary, the Delta-Modulator system in the original design can no longer guarantee stabilization for the whole of region  $R_1$  when bits are lost. We present next some simulations that show the behavior of both the error and the state when bits are lost.

#### A. Example 3

Consider the system given by

$$\begin{aligned} x(k+1) &= 1.5x(k) + u(k) \\ u(k) &= -0.8\hat{x}(k) \end{aligned}$$

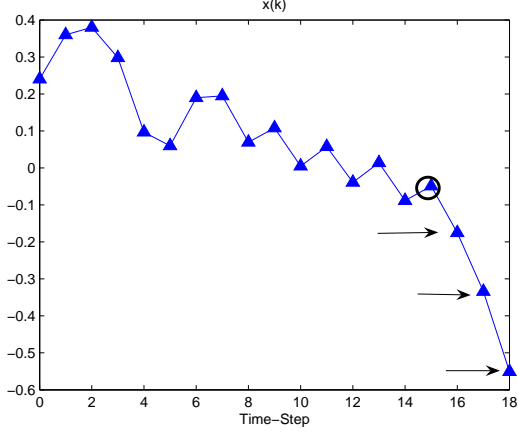


Fig. 8.  $x(k)$  evolution when 3 packet losses occur from  $k = 15$  to  $k = 17$

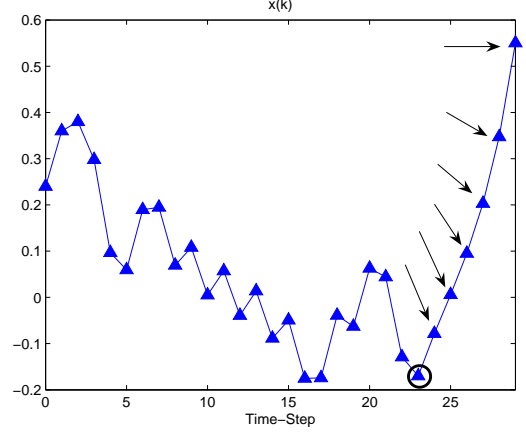


Fig. 10.  $x(k)$  evolution when 6 packet losses occur from  $k = 24$  to  $k = 29$

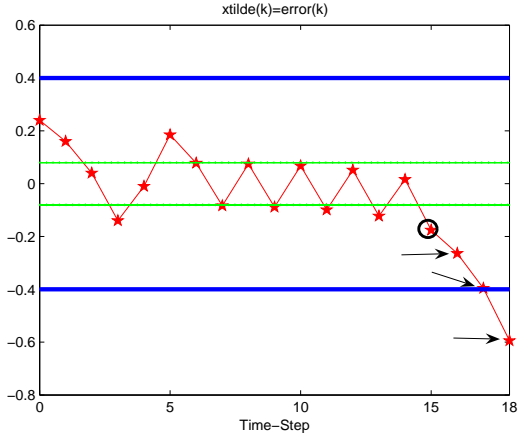


Fig. 9.  $\tilde{x}(k)$  evolution when 3 packet losses occur from  $k = 15$  to  $k = 17$

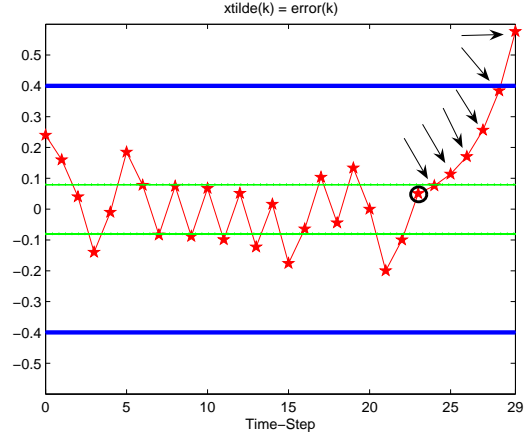


Fig. 11.  $\tilde{x}(k)$  evolution when 6 packet losses occur from  $k = 24$  to  $k = 29$

Suppose  $\Delta = 0.2$ , then  $r_1 = 0.08$ ,  $r_2 = 0.4$  and suppose  $x(0) = 0.24$ . Now let suppose that in the time interval  $0 \leq k \leq 15$  no bits are lost. The error at  $k = 15$  is  $\tilde{x}(k) = -0.1761$ , i.e, it is within the region  $R_1$  and suppose now that 3 consecutive bits are lost. Equation (19) gives us a maximum of 2 consecutive bit losses before we leave the stability region. In fact, we see in figure (9) that if 3 bits are lost  $\tilde{x}(k)$  goes outside  $-r_2$  and, therefore, outside region  $R_1$ . We also show the evolution of the state  $x(k)$  in figure (8).

#### B. Example 4

We want to clarify that equation (19) is actually valid for all the region  $|\tilde{x}(k)| < r_2$  and not just for the stability region. In this example we consider the same system as before but we assume that no bits are lost in the time interval  $0 \leq k \leq 23$ . Therefore, at  $k = 23$  we have  $\tilde{x}(k) = 0.05059 < r_1$ , i.e, we are in the interior ball where the change in the Lyapunov function is positive. But still, equation (19) predicts that more than 5 consecutive bit losses will cause the error to

reach the instability region. Figures (10) and (11) illustrate this.

#### VII. COMPENSATION FOR PACKET-LOSSES

We know from the results in [10] that in the case of noisy channels (for example, an erasure channel), the rate, or more accurately the capacity of the channel is no longer limited by  $R = \log_2(a)$  but instead by  $R = \frac{\log_2(a)}{\gamma}$ , where  $\gamma$  is the probability that the packet arrived. In other words, we have to increase the rate of communications in order to guarantee stabilization of the system.

For the purpose of our design, we will assume that  $1 \leq a \leq 2$  and study whether the 2-Bit-Delta-Modulator scheme can help with the lost packets issue. The reason of this assumption is that we want to use the extra rate provided by the second bit to compensate for lost bits instead of accommodating more unstable systems. We want to clarify that our “packet” is now composed by the concatenation of the two bits,  $b_1$  and  $b_2$ , that are sent through the channel.

Let us use the same 2-Bit-Delta-Modulator Scheme of

figure (4). If  $\tilde{x}(k) \in R_1$ , the multiplexer selection input,  $b_1$ , will choose to multiply by  $\Delta_1$ . If we are outside the region  $R_1$ ,  $b_1$  will choose to multiply by  $\Delta_2$ . Recall from section V that  $\Delta_2 > \Delta_1$ .

Obviously, this scheme only gives guarantees if we assume that no packets are dropped when  $\tilde{x}(k)$  is outside  $R_1$  (actually, the scheme tolerates some packet drops outside  $R_1$  that will be dependent of  $\tilde{x}$  as we will see later). Let us then establish the following condition for our design: the maximum consecutive number of packets that can be lost when we are in ANY subregion within  $R_1$  is  $\beta$  where  $\beta \in \mathbb{N}$ . With this condition, we can guarantee that in any subregion  $R_1$ ,  $\beta$  packets lost may be tolerated, but for some of these subregions even more packets may be handled.

As we see in the examples, for any error in the region  $R_1$ , the worst case scenario in terms of packet losses is when  $\tilde{x}(k)$  is “near”  $r_2$ . Let us therefore quantify how “close” must be  $\tilde{x}(k)$  to  $r_2$  in order to go outside the region after  $\beta$  lost packets. From equation (18) this is given by:

$$\begin{aligned} a^\beta |\tilde{x}(k)| &< \frac{\Delta_1}{a-1} \\ |\tilde{x}(k)| &< \frac{\Delta_1}{a^\beta (a-1)} \end{aligned} \quad (20)$$

Therefore, the region

$$R_p = \left\{ \tilde{x} : \frac{\Delta_1}{a^\beta (a-1)} < |\tilde{x}(k)| < \frac{\Delta_1}{(a-1)} \right\} \quad (21)$$

is the one where  $\beta$  packet losses will force the system into instability. Moreover, the extremes cases occur when  $\tilde{x}(k) = \frac{\Delta_1}{a-1}$ , denoted  $\tilde{x}_{sup}$ , and  $\tilde{x}(k) = \frac{\Delta_1}{a^\beta (a-1)}$ , denoted  $\tilde{x}_{inf}$ .

If either of these two extreme cases occur, then after  $\beta$  packet losses we will be in

$$\tilde{x}(k+\beta) = \begin{cases} \frac{a^\beta \Delta_1}{a-1} & \text{for } \tilde{x}_{sup} \\ \frac{\Delta_1}{a-1} & \text{for } \tilde{x}_{inf} \end{cases}$$

But this will imply that the 2-Bit Delta Modulator will change the value of  $b_1$  and use the product with  $\Delta_2$ .

Now, because we assume that  $\beta$  is the maximum number of packets that can be lost consecutively, then we know that  $b_1(k+n+1)$  and  $b_2(k+n+1)$  will arrive to the decoder and this will also switch to  $\Delta_2$ . Recalling from section V, the value of  $\Delta_2$  that can match the regions of the two modulators is given by equation (10). Now, if the state is located in the worst part of the region,  $\tilde{x}(k) = r_2$ , and we want to have  $\beta$  consecutive packet losses without encountering instability, we must guarantee that  $a^\beta r_2 \leq r_4$ . That is equivalent to

$$a^\beta \frac{\Delta_1}{a-1} \leq \frac{(a+1)\Delta_1}{a-1}$$

From this last inequality, the maximum number of packets loss is given by

$$\beta = \left\lfloor \frac{\log_2\left(\frac{a+1}{a-1}\right)}{\log_2(a)} \right\rfloor \quad (22)$$

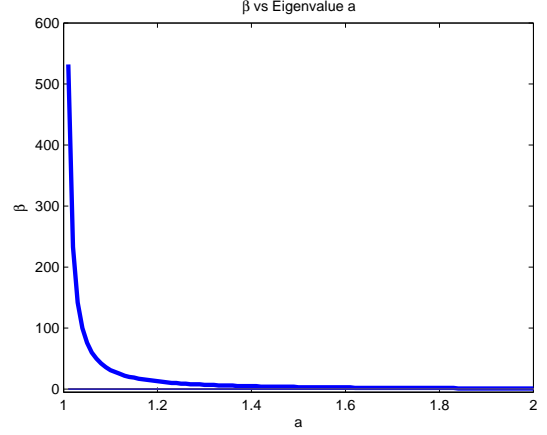


Fig. 12.  $\beta$  versus Eigenvalue  $a$

Equation (22) provides the maximum number of consecutive packets that can be lost when we are in the worst case (subregion of  $R_p$ ) without losing stability when using the 2-Bit Delta Modulator Scheme. Obviously, this is an inverse relation of the eigenvalue of the system: the number  $\beta$  of packets that we can afford to lose, is small for  $a$  approaching 2 (faster dynamics) and is large for  $a$  approaching 1 (slower dynamics). This can be seen in the figure (12). It is important to note that because  $|a| \leq 2$  the only way that we can get out of region  $R_1$  is due to packet losses. Moreover, if we are within  $R_1$ , the system may lose up to  $\beta$  packets without going unstable. We know that after  $\beta$  packet losses, the system may end up in the region  $R_2$ . In  $R_2$  however, we can no longer guarantee that any more lost packets are tolerated. The number of lost packets that may be handled will obviously vary, according to  $\tilde{x}(k)$  and may be determined by an equation similar to (19) as follows

$$l_2 \leq \left\lfloor \frac{\log_2\left(\frac{\Delta_2}{(a-1)|\tilde{x}(k)|}\right)}{\log_2(a)} \right\rfloor \quad (23)$$

This will force the system to return to the region  $R_1$  in order to guarantee that  $\beta$  packets may be lost again without losing stability.

#### A. Example 5

Consider the system given by

$$\begin{aligned} x(k+1) &= 1.2x(k) + u(k) \\ u(k) &= -0.8\hat{x}(k) \end{aligned}$$

Suppose  $\Delta_1 = 0.2$ ,  $\Delta_2 = 2.2$  then  $r_1 = 0.0909$ ,  $r_2 = 1$ ,  $r_3 = r_2 = 1$ ,  $r_4 = 11$  and suppose  $x(0) = 0.547$ ; therefore,  $R_1 : r_1 \leq \tilde{x}(0) = 0.547 \leq r_2$  and  $R_p : 0.093 \leq |\tilde{x}| \leq 1$ .

According to equation (22), the  $\beta$  number of packets that can be lost is 13 in any subregion of  $R_1$ . We assume that starting at  $k = 0$  the system loses its first packet and consecutively loses one packet per sampling time until  $k = 12$  (13 lost packets in total). The system then loses no



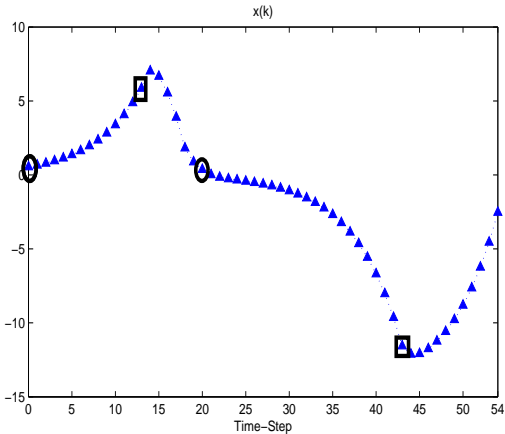


Fig. 13.  $x(k)$  evolution when 13 packets are lost from  $k=0$  to  $k=12$  and then 22 packets lost from  $k=20$  to  $k=42$

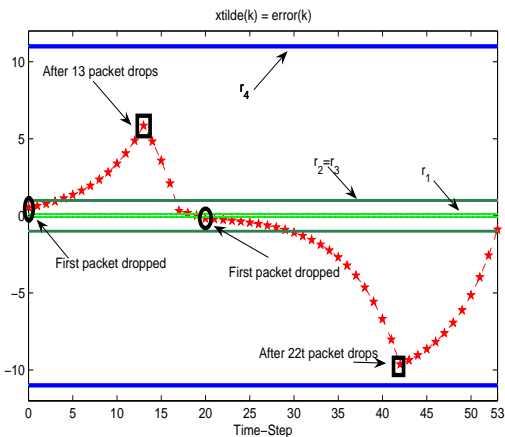


Fig. 14.  $\tilde{x}(k)$  evolution when 13 packets are lost from  $k=0$  to  $k=12$  and then 22 packets lost from  $k=20$  to  $k=42$

more packets until  $k=20$ , where  $\tilde{x}(k) \approx 0.02 < 0.093 < 1$  (less than  $r_2$  and out of subregion  $R_p$ ) where it starts losing a packet per sampling time until  $k=42$ , i.e., 22 packets in total, then it continues its operation without suffering any more lost packets.

The behavior of the system, (both the state  $x(k)$  and the error  $\tilde{x}(k)$ ) is illustrated in figures (13) and (14). The circles indicate the instants when the packet losses start, and the rectangles indicate when the packet losses have ended. In figure (14) the limits for region  $R_1$ ,  $\pm r_1$  and  $\pm r_2$ , are shown as well as the regions added by the second packet,  $\pm r_3 = \pm r_2$  and  $\pm r_4$ . We see that  $\beta = 13$  is not the maximum for subregions that are different from  $R_p$ .

### VIII. CONCLUSIONS AND FUTURE WORK

This paper has provided extensions to previous results that make it possible to stabilize scalar systems with eigenvalues greater than 2 in magnitude. We then presented a new design of a 2-Bit Delta-Modulator-Like encoder/decoder

scheme that keeps the simplicity and desirable characteristics of the 1-Bit scheme.

We have also included the effects of packets lost in the channel and showed how to re-synchronize the encoder and decoder. We determined the number of consecutive packets that can be lost before going into instability and finally, we presented a similar 2-Bit Delta-Modulator-Like encoder/decoder scheme that allows us to handle a specific and predetermined number of lost packets.

In the future, we plan to extend these results to multidimensional systems, and to include time-delays in the channel. The idea of the 2-Bit Delta Modulator may also be extended for the M-Bit case in order to control systems with arbitrary magnitude eigenvalues  $a$  and/or to allow for a larger number of dropped packets.

### REFERENCES

- [1] R. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45(7):1279–1289, July 2000.
- [2] C. C. de Wit, F. Rubio, J. Fornes, and F. Gomez-Estern. Differential coding in networked controlled linear systems. *Proceedings of the American Control Conference*, 2006.
- [3] V. Gupta, D. Spanos, B. Hassibi, and R. Murray. On lqg control across a stochastic packet-dropping link. *Proceedings of the American Control Conference*, 2005.
- [4] J. Hespanha, A. Ortega, and L. Vasudevan. Towards the control of linear systems with minimum bit-rate. *Proceedings of the Int. Symp. on the Mathematical Theory of Networks and Systems*, 2002.
- [5] I. Lopez, J. Piovesan, C. Abdallah, D. Lee, O. Martinez, M. Spong, and R. Sandoval. Practical issues in networked control systems. *Proceedings of the American Control Conference*, June 2006.
- [6] S. Mitter. Control with limited information: the role of systems theory and information theory. *IEEE Information Theory Society Newsletter, Eur. J. Control*, 7:1–23, 2001.
- [7] G. Nair and R. Evans. Communication limited stabilization of linear systems. volume 1, pages 1005–1010, Dec 2000.
- [8] L. Shi, M. Epstein, and R. Murray. Towards robust control over a packet dropping network. *International Symposium on Mathematical Theory of Networks and Systems*, 2006.
- [9] L. Shi and R. Murray. Towards a packet-based control theory - part i: Stabilization over a packet-based network. In *Proceedings of the American Control Conference*, volume 2, pages 1251–1256, June 2005.
- [10] S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Transactions on Automatic Control*, 49(7):1196–1201, 2004.
- [11] S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004.
- [12] W. Wong and R. Brockett. Systems with finite communication bandwidth-part ii: Stabilization with limited information feedback. *IEEE Transactions on Automatic Control*, 44(5):1049–1053, May 1999.