INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2002; **00**:1–6 *Prepared using* rncauth.cls *[Version: 2002/11/11 v1.00]*

Rate Limited Stabilization: Sub-optimal Encoder-Decoder Schemes

I. Lopez Hurtado † *and C.T. Abdallah ‡

Department of Electrical & Computer Engineering,University of New Mexico

SUMMARY

In this paper, we extend results from packet-based control theory and present sufficient conditions on the rate of a packet network to guarantee asymptotic stabilizability of unstable discrete LTI systems with less inputs than states. Two types of Network Control Systems are considered in the absence of communication delays, then for one of the two types, the case of a constant time delay is discussed. Examples and simulations are provided to illustrate the results. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: Limited Data Rates, Networked Controlled Systems, Quantization, Linear Systems.

1. Introduction

Feedback control systems wherein the control loops are closed through a real-time network are called networked control systems (NCS) [17], [18]. The primary advantage of a NCS is that a reduced of number system components and connections are used resulting in easier maintenance and diagnosis of the system. However, when controlling across networks, several assumptions of classical theory of control may need to be revisited. For example, the delay from the sensor to the controller may be time-varying or random, and similarly for the delay from the controller to the actuators. This specific issue has been analyzed in several works, for example:[4], [7], [3] and [8]. New complications may therefore arise because the sensed data and the control signals are no longer connected directly through a "dedicated wire", but rather through a data network which has a finite bandwidth (or a finite data rate), and which may also be shared by many other systems. In recent years, much research has been expanded in the area of NCS and because of the benefits of remote industrial control, several reliable protocols have been developed for real-time control purposes. Meanwhile, computer networking technologies have witnessed incredible advances in recent years. With the decrease in cost, the increase in performance, and with the steady investment in infrastructure, the Internet has

*Correspondence to: Department of Electrical & Computer Engineering, Room 333, MSC01 1100, 1 University of New Mexico, Albuquerque, NM 87131-0001, USA.
†Email: ivlopez@unm.edu
‡Email: chaouki@ece.unm.edu

Contract/grant sponsor: First author is partially supported by Conacyt; second author is partially supported by NSF award under the FIND initiative:; contract/grant number: CNS 0626380

Received 15 January 2008

in fact become a suitable network for industrial control applications. In 1999, Wong and Brockett [19] considered a feedback system communicating through a digital channel with finite capacity, and since asymptotic stability was deemed unrealistic, the concept of containability was introduced. Mitter [6] and collaborators have contributed to the development of a new theory that matches classical control theory with traditional information theory, [2], [16], [15] and [10]. In [16], an efficient encoder-decoder scheme is proposed to guarantee stabilization of a class of discrete linear time-invariant (DLTI) system using the minimum rate imposed by the Data Rate Theorem [16]. Reference [5] presented an encoder/decoder scheme that also achieved the minimum data rate while also considering packet losses. Similarly, reference [7] presents an encoder-decoder scheme that deals with uncertainty in the plant model. It is clear in all of these schemes that the cost of reducing the data rate is the complexity in the algorithm and the computational power required for the encoding/decoding operations. There may however be situations where simpler algorithms are preferred, at the expense of having a higher data rate. The purpose of this paper is to provide such simple encoder/decoder schemes that are easy to implement while requiring a higher data rate in order to guarantee asymptotic stability.

The first scheme that we present is based on ideas proposed in [13], [14] and [12]. The authors of those papers considered a general DLTI system $x(k+1) = Ax(k) + Bu(k)$ and found a sufficient rate for exponential stabilization of an unstable plant of order n, under the rather limiting assumption that the system has n inputs (where n is the number of states) and an invertible input matrix B. The work addressed finite rate issues, packet dropping, as well as uncertainties in the plant model. Moreover, the authors assumed the existence of a truncation-based encoder/decoder without providing the specific structure of this encoder/decoder.

We extend the results of [12] to the case of discrete-time, linear, time-invariant systems with m inputs such that $m < n$, where n is the order of the system. We also relax the condition of the invertibility of the B matrix, and extend the stabilizability results to systems with a constant time-delay induced by the sensor-to-controller network. Moreover, we present an easily implementable encoder/decoder structure. As was considered in [12], we discuss two types of network control systems: one that includes a network between the sensors and the controller, and another that models two networks in the loop, one between the sensors and controller, and another between the controller and the actuator.

Finally, we propose a new encoder/decoder scheme that is more complex but that uses a lower rate than the truncation-based encoder/decoder. The new scheme is also less complex, but requires a higher rate than the ones presented in [16], [5] and [7].

2. Problem Setup

We consider the two configurations for the packet-based network control system presented in [12]. The first system, is referred to as *Network Control System Type I*, has a rate of R_{p1} packets/sample-time. This packet based network accomodates a packet size of D_{Max} bits used for data (although the protocol information requires extra bits in the packet, it is not needed for this analysis). Let us consider the discrete LTI system shown in Figure 1 and described by

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where A is $n \times n$, B is $n \times m$ and $u(k)$ is $m \times 1$.

The second type of packet-based network, referred to as *Network Control System Type II*, consists of the same discrete LTI system given by equation (1), but with the addition of a second network between the controller and the actuator with rate R_{p2} as shown in Figure 2.

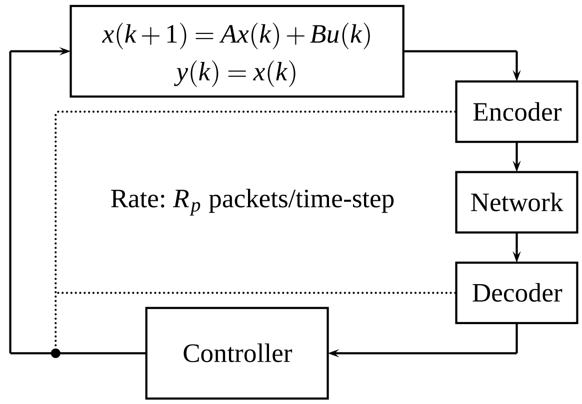

Figure 1. Closed-loop network control system: Type 1

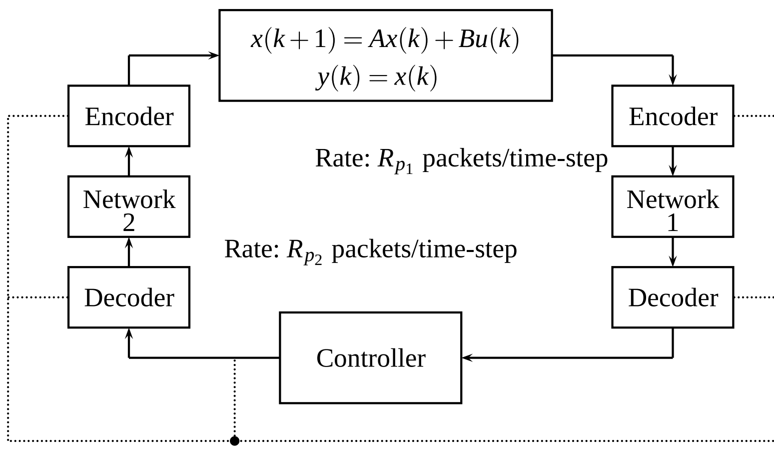

Figure 2. Closed-loop network control system: Type 2

From here on, the following notations are adopted. The **log** function is base 2, the norm symbol ($\|.\|$) denotes the Euclidean norm and $\lceil . \rceil$ is the **ceil** function. In addition, we use the variable μ to denote the controllability index which for multivariable linear systems [1] is defined as the least integer k such that

$$\operatorname{rank} \begin{bmatrix} B| & AB| & \ldots | & A^{k-1}B \end{bmatrix} = n. \tag{2}$$

We assume that the controller does not saturate, and that the packet-network does not drop packets nor is it subjected to disturbances (noise). For both NCS types, we assume that the states may be measured. We also assume equimemory of the encoder and decoder so that the decoder knows exactly the encoding scheme used by the encoder at all times. This last assumption is explained in Section 3 where we present the encoder and decoder schemes. Finally, we assume that both the encoder and decoder, know a value $L_0 > 0$ such that $\|x(0)\| < L_0$ and that both have access to the control signal or can compute it (this is represented by a dotted line in Figures 1 and 2).

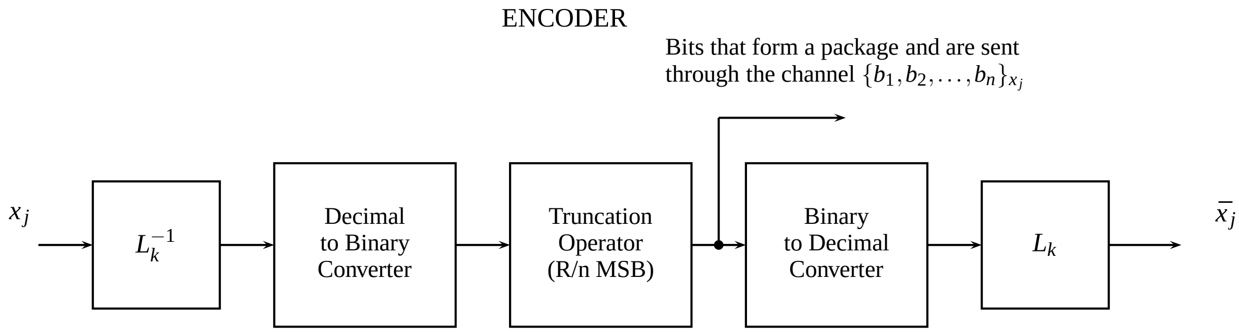

Figure 3. Encoder Scheme

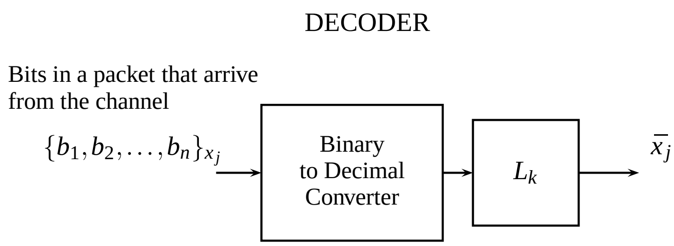

Figure 4. Decoder Scheme

3. Encoder-Decoder Design

Several approaches for the design of an encoder/decoder scheme were presented in previous works. Most of them are based on some type of predictor that emulates the evolution of the plant state and the difference between this prediction and on the actual measure of the plant, i.e., the error. The quantized error is sent through the channel, then decoded at the receiver and used to obtained an approximation of the state, which is used to generate the control signal. In our case however, we send a quantized version of every state component rather than the error using a modified version of the encoder/decoder scheme proposed in [11]. Figures 3 and 4 illustrate our scheme which is described next in detail. At the first instant, $k = 0$, the sensor measures the state exactly. Since we assume that both the encoder and decoder know L_0, each component x_j of the measured state is divided by L_0 which gives a number x_j/L_0 that is less than or equal than 1 in magnitude. We assume for now that x_j/L_0 is less than one and positive (in Section 4 we will comment how to solve if it is exactly 1 or negative). The encoder converts this number to its binary representation and keeps only the r_j most significant bits (MSB). This truncated version is labeled as $\left(\frac{x_j(0)}{L_0}\right)_t$. The quantity r_j will be later calculated in Section 4. The decimal representation of these r_j bits is multiplied by L_0 resulting in an estimate $\bar{x}_j(0) = \left(\frac{x_j(0)}{L_0}\right)_t L_0$ which is stored in the encoder. By grouping in a vector the j truncated state components we obtain the state estimate $\bar{x}(0)$. The bits in each truncated state component form a packet (or packets depending on D_{Max}) that is sent through the channel. On the receiver side, the decoder receives a packet (or packets) and separates the bits that correspond to each state component. It then converts the binary representation of the bits received into a decimal representation and multiplies by L_0 which gives the value $\bar{x}_j(0)$. This should the same value stored in the encoder and, therefore, the equimemory property between encoder and decoder is preserved. Since the control signal at time $k=1$ only depends on $\bar{x}(0)$, we can show that at time $k=1$, $x_j(1)$, is bounded as follows. Using the triangle inequality and matrix norm properties we have:

$$\begin{aligned}\|x(1)\| &\leq \|Ax(0)+Bu(\bar{x}(0))\| \\ &\leq \|A\|\|x(0)\|+\|Bu(\bar{x}(0))\| \\ &\leq \|A\|L_0+\|Bu(\bar{x}(0))\| \\ &= L_1.\end{aligned}$$

Since the control algorithm is predefined, the encoder and decoder can both calculate this value L_1 right after they have calculated the value $\bar{x}(0)$. The stored L_1 will then be used at instant $k=1$ to keep the ratio $|x(1)/L_1| \leq 1$. By carefully examining the above steps, we obtain the following difference scalar equation to bound the norm of each state component:

$$L_k = \|A\|L_{k-1} + \|Bu(\bar{x}(k-1))\| \tag{3}$$

Since the equation above only depends on the terms L_{k-1} and $\bar{x}(k-1)$, all signals needed to compute this equation are available at the encoder and the decoder. We also note that since A is unstable, the $\|A\| > 1$ and L_k will become unbounded, but we will show in section 4 that L_k will only grow μ time-steps, before it is reset to a new starting value for another μ time-steps.

4. Results

4.1. Network Control System: Type I

In the case of NCS Type I, the state vector $x(k)$ is given by

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}. \tag{4}$$

We assume below that $x_j(k) > 0$, $\forall j$ since the sign of each state component may later be accounted for by adding n extra bits to the rate (one extra bit per state component). We then obtain the following binary representation of $\frac{x(0)}{L_0}$ at the encoder side:

$$\frac{x(0)}{L_0} = \begin{bmatrix} \frac{x_1(0)}{L_0} \\ \frac{x_2(0)}{L_0} \\ \vdots \\ \frac{x_n(0)}{L_0} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{\infty} \alpha_{1i} 2^{-i} \\ \sum_{i=1}^{\infty} \alpha_{2i} 2^{-i} \\ \vdots \\ \sum_{i=1}^{\infty} \alpha_{ni} 2^{-i} \end{bmatrix}; \tag{5}$$

where $\alpha_{ij} \in \{0,1\}$. This binary representation is truncated keeping only the r_j most significant bits for state component x_j. The truncated representation is given by:

$$\left(\frac{x(0)}{L_0}\right)_T = \begin{bmatrix} \left(\frac{x_1(0)}{L_0}\right)_T \\ \left(\frac{x_2(0)}{L_0}\right)_T \\ \vdots \\ \left(\frac{x_n(0)}{L_0}\right)_T \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{r_1} \alpha_{1i} 2^{-i} \\ \sum_{i=1}^{r_2} \alpha_{2i} 2^{-i} \\ \vdots \\ \sum_{i=1}^{r_n} \alpha_{ni} 2^{-i} \end{bmatrix}; \tag{6}$$

where $\alpha_{ij} \in \{0,1\}$. The r_j bits per state component j are sent through the channel and, at the receiver site, the decoder transforms the bits back into decimal numbers, and multiplies them by L_0 in order to obtain $\bar{x}(0)$. With this encoding/decoding process, we guarantee that the error between the actual state component and its encoded version, $\varepsilon_j(0) = x_j(0) - \bar{x}_j(0)$, is limited by $\|\varepsilon_j(0)\| < 2^{-r_j} L_0$, $\forall j \in \{0,1,\ldots,n\}$. Using the triangle inequality, the norm of the total error is bounded by

$$\begin{aligned} \|\varepsilon(0)\| &\leqslant \|\varepsilon_1(0)\| + \ldots + \|\varepsilon_n(0)\| \\ &\leqslant \sqrt{\sum_{j=1}^{n} 2^{-2r_j}} L_0. \end{aligned} \tag{7}$$

Let us then consider the evolution of the system starting at time $k = 0$:

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ x(2) &= Ax(1) + Bu(1) \\ &= A^2 x(0) + ABu(0) + Bu(1) \\ &\vdots \\ x(l) &= A^l x(0) + \sum_{i=1}^{l} A^{l-i} Bu(i-1); \ \ \forall l \geq 3. \end{aligned}$$

Recalling that μ represents the controllability index, after $k+\mu$ steps we have

$$\begin{aligned} x(\mu) &= A^{\mu} x(0) + A^{\mu-1} Bu(0) + A^{\mu-2} Bu(1) \\ &\quad + \ldots + Bu(\mu - 1). \end{aligned} \tag{8}$$

This equation may be re-arranged as $x(\mu) = A^{\mu} x(0) + \zeta_{\mu} \mathbb{U}$, where

$$\begin{aligned} \zeta_{\mu} &= \begin{bmatrix} B| & AB| & \ldots| & A^{\mu-1}B \end{bmatrix} \\ &= \begin{bmatrix} \delta_1| & \delta_2| & \ldots| & \delta_j| & \ldots| & \delta_{n\mu} \end{bmatrix} \end{aligned}$$

and

$$\mathbb{U} = \begin{bmatrix} u(\mu-1) \\ \vdots \\ u(0) \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_{m\mu} \end{bmatrix},$$

noting that δ_j is the jth column in ζ_μ and u_j is the jth element in the vector $\mathbb{U}$. Let us select the first n independent columns of ζ_μ and build a new matrix, called ζ_n. Let us also select the elements of $\mathbb{U}(k)$ corresponding to the columns chosen from ζ_μ and form a new vector, called $\mathbb{U}_n$. Recalling that $x(0) = \bar{x}(0) + \varepsilon(0)$ we have $x(\mu) = A^\mu \bar{x}(k) + A^\mu \varepsilon(k) + \zeta_\mu \mathbb{U}(k)$. If we choose the control law

$$\mathbb{U}_n = -\zeta_\mu^{-1} A^\mu \bar{x}(0); \tag{9}$$

we may reconstruct $\mathbb{U}$ replacing u_j with the corresponding values of $\mathbb{U}_n$ in the proper order and letting $u_j = 0$ for the remaining elements. After μ steps, and by applying the control sequence $\mathbb{U}(k)$ we obtain

$$x(\mu) = A^\mu \varepsilon(0). \tag{10}$$

Then, from equations (20) and (7) and the properties of matrix norms, we obtain

$$\begin{aligned} \|x(\mu)\| &= \|A^\mu \varepsilon(0)\| \\ &\leqslant \|A^\mu\| \|\varepsilon(0)\| \\ &\leqslant \|A^\mu\| \sqrt{\sum_{j=1}^{n} 2^{-2r_j}} L_0. \end{aligned}$$

In order to force the state to decrease in the norm (after μ steps), we shrink the upper bound of the state $x(\mu)$ by forcing it to be less than a fraction of the upper bound of the state $x(0)$, i.e., $\|A^\mu\| \sqrt{\sum_{j=1}^{n} 2^{-2r_j}} L_0 < L_0/\delta$, for some $\delta > 1$. At this point, we have to decide on the value of each r_j. This may be converted into an optimization problem whose objective is to minimize the total rate given by $\sum_{j=1}^{n} r_j$. In other words, let us consider the optimization problem:

$$\min_{r_j} \sum_{j=1}^{n} r_j \tag{11}$$

subject to

$$\sqrt{\sum_{j=1}^{n} 2^{-2r_j}} < \frac{1}{\delta \|A^\mu\|} = C_* \tag{12}$$

This problem may be solved by applying the Karush-Kuhn-Tucker (KKT) conditions [9] on the Lagrangian function $L(r_1, r_2, \ldots, r_n, l)$ with Lagrange multiplier l as is given by

$$L = r_1 + r_2 + \ldots + r_n - l(C_* - \sqrt{2^{-2r_1} + 2^{-2r_2} + \ldots + 2^{-2r_n}}).$$

The KKT conditions are then:

$$\begin{aligned} \frac{\partial L}{\partial r_1} &= 1 - l \frac{2^{-2r_1} \ln(2)}{\sqrt{2^{-2r_1} + 2^{-2r_2} + \ldots + 2^{-2r_n}}} = 0; \\ \frac{\partial L}{\partial r_2} &= 1 - l \frac{2^{-2r_2} \ln(2)}{\sqrt{2^{-2r_1} + 2^{-2r_2} + \ldots + 2^{-2r_n}}} = 0; \\ &\vdots \\ \frac{\partial L}{\partial r_2} &= 1 - l \frac{2^{-2r_n} \ln(2)}{\sqrt{2^{-2r_1} + 2^{-2r_2} + \ldots + 2^{-2r_n}}} = 0; \end{aligned}$$

Solving this system of n-equations system, we obtain:

$$r_1 = r_2 = \ldots = r_j = \ldots = r_n.$$

Therefore, an equal allocation of bits per each state component actually guarantees the minimum total rate. Using the constraint (12) we obtain the optimal rate allocation $r_n \geqslant \lceil \log(\|A^\mu\|) + \frac{1}{2}\log n + \log\delta \rceil$. We notice that δ is a parameter that determines the fraction by which the upper bound of $\|x(0)\|$ is shrinking. Therefore, it is sufficient to consider the *infimum* of this quantity to obtain $r_n \geqslant \lceil \log(\|A^\mu\|) + \frac{1}{2}\log n \rceil$. Note that the $\lceil . \rceil$ function was introduced since r_n must be an integer denoting the number of bits for each state component. We can therefore define the total R bits in a packet (or packets) as $R = nr_n + n$ where the second n term may be used to code the sign of each state component.

For the next μ steps, we repeat the same steps above but using $x(\mu)$ as the initial condition. To stop the growth of L_k, and noting that $x(\mu) \leq n\|A^\mu\|2^{-r_n}L_0$, we reset $L_\mu = n\|A^\mu\|2^{-r_n}L_0$ for the next μ time steps. We repeat this procedure every μ steps. Using the same algorithm to generate the control sequence and the same rate R, the state $x(2\mu)$ will be a shrunken version of $x(\mu)$. Proceeding in the same fashion, $x(t\mu)$ will tend to zero as $t \in \mathbb{N}$ grows and, therefore, the state x will tend to zero and asymptotic stabilizability will be achieved. Note that R is the sufficient number effective bits that we need to transmit of the whole state for stabilization, but since a packet has a maximum length D_{Max}, if $R \leq D_{Max}$, we need a packet rate of $R_p = 1$ packet/sample-time. If on the other hand, $R > D_{Max}$ then, a minimum of $\left\lceil \frac{R}{D_{Max}} \right\rceil$ packets/time-step are needed. Note that the last expression actually covers both cases, since $\frac{R}{D_{Max}} < 1$ gives a 1 packet/sample-time when the ceil function is applied.

■

This analysis may be summarized in the following theorem.

Theorem 4.1. *Assuming an equal allocation of bits per state component, a network rate R_p packets/time-step, and assuming that (A,B) is a controllable pair with controllability index μ, a sufficient condition for system (1) to be asymptotically stabilizable is*

$$R_p \geqslant \left\lceil \frac{R}{D_{Max}} \right\rceil,$$

where $R = n\lceil \log(\|A^\mu\|) + \log n \rceil + n$ and every state allocates $\frac{R}{n}$ bits/time-step.

An immediate consequence of Theorem 4.1 in the specific case of a single input system is given in the following corollary.

Corollary 4.1. *Assuming an equal allocation of bits per state component, a network rate R_p packets/time-step, (A,B) is a controllable pair, and B is $n \times 1$ and the control law, $u(k)$, is 1×1, a sufficient condition for system (1) to be asymptotically stabilizable is*

$$R_p \geq \left\lceil \frac{R}{D_{Max}} \right\rceil,$$

where $R = n\lceil \log(\|A^n\|) + \log n \rceil + n$ and every state allocates $\frac{R}{n}$ bits/sample.

Proof: The proof is the same as that of Theorem 4.1. If B is $n \times 1$ and $u(k)$ is 1×1, then $\mu = n$. Substituting μ in R in the proof of Theorem 4.1, we obtain the rate given by the corollary.

■

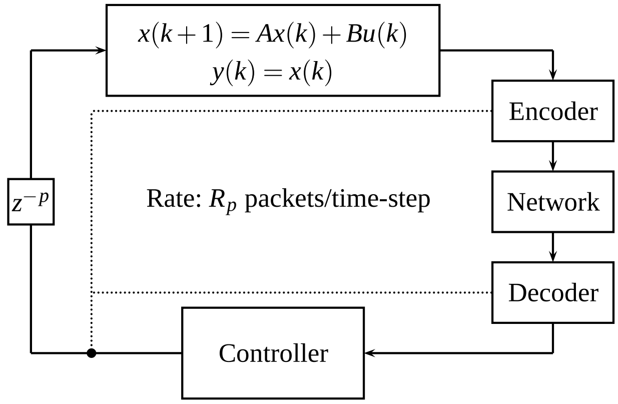

Figure 5. Closed-loop network control system: Type 1

4.2. Network Control System Type I with Time Delay

One of our motivations for extending the results of [12], was to include the effects of time delays that may be present in the network. As mentioned earlier, even for the scalar case, the invertibility requirement of B would not allow the traditional augmentation of the state by its delayed versions. Let us consider the modified network control system type I shown in Figure 5 and the discrete LTI system given by the following equation

$$x(k+1) = Ax(k) + Bu(k-p), \tag{13}$$

where A is $n \times n$, B is $n \times 1$ and $u(k)$ is 1×1. We assume here that the control signal to actuator delay is a constant equal to $p \in \mathbb{N}$ time-steps. Under such conditions, we obtain the following theorem

Theorem 4.2. *Assuming an equal allocation of bits per state component, a network rate of* $R_p = \left\lceil \frac{R}{D_{Max}} \right\rceil$ *packets/time-step, and*

$$\mathbb{A} = \begin{bmatrix} A & B & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ & & & & 1 \\ 0 & 0 & \vdots & \dots & 0 \end{bmatrix}, \quad \mathbb{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

such that $(\mathbb{A}, \mathbb{B})$ *is a controllable pair. A sufficient condition for system (13) to be asymptotically stabilizable is*

$$R_p \geq \left\lceil \frac{R}{D_{Max}} \right\rceil;$$

where $R = (n+p)\lceil \log(\|\mathbb{A}^{n+p}\|) + \log n \rceil + n$, *and each state component of the augmented system allocates* $\frac{R}{n+p}$ *bits/time-step.*

Proof: We start out by augmenting the state vector, considering as new states the last p previous inputs. We then obtain

$$\begin{aligned}\mathbb{X}(k+1) &= \begin{bmatrix}\mathbf{x}(k+1)\\ x_{n+1}(k+1)\\ x_{n+2}(k+1)\\ \vdots\\ x_{n+p}(k+1)\end{bmatrix}\\ &= \begin{bmatrix}A & B & 0 & \dots & 0\\ 0 & 0 & 1 & \dots & 0\\ 0 & 0 & 0 & \dots & 0\\ & & & & 1\\ 0 & 0 & \vdots & \dots & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}(k)\\ x_{n+1}(k)\\ x_{n+2}(k)\\ \vdots\\ x_{n+p}(k)\end{bmatrix} + \begin{bmatrix}0\\0\\0\\ \vdots\\ 1\end{bmatrix}u(k).\end{aligned}$$

This may be written as

$$\mathbb{X}(k+1) = \mathbb{A}\mathbb{X}(k) + \mathbb{B}u(k). \tag{14}$$

We now have a system similar to the one treated in Corollary 4.1 with a state dimension $n+p$ instead of n. Therefore, in order to shrink the upper bound of the state $\mathbb{X}(k+n+p)$ we need a rate R given by

$$\frac{R}{n+p} \geqslant \left\lceil \log(\|\mathbb{A}^{n+p}\|) + \frac{1}{2}\log(n+p)\right\rceil + 1.$$

Similarly to previous proofs, we find a minimum rate of $R_p = \left\lceil \frac{R}{D_{Max}} \right\rceil$ packets/time-step.

■

4.3. Network Control System: Type II

We now consider an NCS Type II and show the following result.

Theorem 4.3. *Assume an equal allocation of bits per state component, network rates of $R_{p1} = \left\lceil \frac{R_1}{D_{Max}} \right\rceil$ packets/time-step and $R_{p2} = \left\lceil \frac{R_2}{D_{Max}} \right\rceil$ packets/time-step for network 1 and 2, respectively. Assuming also that (A,B) is a controllable pair, where B is $n\times 1$, the controllability matrix is given by $\zeta = \begin{bmatrix}B| & AB| & \dots| & A^{n-1}B\end{bmatrix}$ and the control law, $u(k)$, is 1×1, a sufficient condition for system (1) to be asymptotically stabilizable is*

$$n\,\|A^n\|\,2^{-\frac{R_1}{n}} + \|\zeta\|\,\left\|\zeta^{-1}A\right\|\,2^{-R_2} < 1.$$

Proof: Since there is now a rate constraint from the controller to the plant actuators, we can no longer apply the calculated control signal $u(k)$ directly to the plant. Instead, only the bits encoding $u(k)$ according to the available rate, R_2, may be used. This encoded control signal $\widetilde{u}(k)$ is the one that is received by the plant. We then have

$$x(k+1) = Ax(k) + B\widetilde{u}(k). \tag{15}$$

Let us assume that we have exactly the same encoding and decoding schemes used in Theorem 4.1. The evolution of the system in the first n time steps is given by $x(n) = A^n x(0) + \zeta\widetilde{\mathbb{U}}$, where $\widetilde{\mathbb{U}} = \left[\widetilde{u}(n-1) \quad \ldots \quad \widetilde{u}(0)\right]'$. If we choose the control signal $\mathbb{U} = -\zeta^{-1}A^n\overline{x}(0)$, then $\|\mathbb{U}\| \leqslant \left\|\zeta^{-1}A^n L_0\right\| \leq \left\|\zeta^{-1}A^n\right\| L_0 = L2_0$. For other time k, the normalization value that is kept in the memory of the encoder/decoder of network II, i.e. $L2_k$, is given by $L2_k = \|\zeta_n^{-1}A^{\mu}\|L_k$.

Since $\widetilde{u}(k)$ represents the R_2 most significant bits of $u(k)$ we know that

$$\left\|\mathbb{U} - \widetilde{\mathbb{U}}\right\| \leqslant \left\|\zeta^{-1}A^n\right\| L_0 2^{-R_2}. \tag{16}$$

From equation (16) and recalling that $x(0) = \overline{x}(0) + \varepsilon(0)$ and, similarly to previous proofs, $\|\varepsilon(0)\| < \sqrt{n}L_0 2^{-\frac{R_1}{n}}$, we have

$$\begin{aligned}
\|x(n)\| &= \left\|A^n\overline{x}(0) + A^n\varepsilon(0) + \zeta\widetilde{\mathbb{U}}\right\| \\
&= \left\|\zeta\left(\zeta^{-1}A^n\overline{x}(0) + \widetilde{\mathbb{U}}\right) + A^n\varepsilon(0)\right\| \\
&\leqslant \|\zeta\| \left\|\mathbb{U}(k) - \widetilde{\mathbb{U}}\right\| + \|A^n\varepsilon(0)\| \\
&\leqslant \|\zeta\| \left\|\zeta^{-1}A\right\| L_0 2^{-R_2} + \sqrt{n}\|A^n\| L_0 2^{-\frac{R_1}{n}} \\
&< \frac{L_0}{\delta}.
\end{aligned}$$

If we want to guarantee the shrinking of $x(n)$, we enforce that $\|\zeta\| \left\|\zeta^{-1}A\right\| L_0 2^{-R2} + \sqrt{n}L_0 \|A^n\| 2^{-\frac{R_1}{n}} < L_0$, i.e., $\sqrt{n}\|A^n\| 2^{-\frac{R_1}{n}} + \|\zeta\| \left\|\zeta^{-1}A\right\| 2^{-R_2} < 1$. As in previous proofs we now select $x(n)$ as the new initial condition and using the same control law and rates, R_1 and R_2, the state $x(2n)$ will be a shrunken version of $x(n)$. Continuing in the same fashion, $x(tn)$ will tend to zero as $t \in \mathbb{N}$ grows and, therefore $x(k)$ will tend to zero and asymptotic stability is achieved. Here again we will need a minimum of $R_{p1} = \left\lceil \frac{R_1}{D_{Max}} \right\rceil$ packets/time-step for the sensor-controller network and a minimum of $R_{p2} = \left\lceil \frac{R_2}{D_{Max}} \right\rceil$ packets/time-step in the controller-actuator network.

■

5. Simulations

To verify some of the results derived in the paper, we present several numerical examples using Matlab®. We note that although $x(k)$ is discrete and exists only at the time instants $k = \{0, 1, 2, \ldots\}$, the plots below show the components of $x(k)$ at all times for ease of visualization.

5.1. Example 1

First, we tested the results of Theorem 4.1 for the system

$$x(k+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} u(k)$$

We assume $L_0 = 71.68$ and choose the initial condition $x(0) = \begin{bmatrix} -16.333 & 30.768 & 8.44 \end{bmatrix}'$. The rate in bits obtained according to Theorem 4.1 is $R = 18$ *bit/time-step* (equivalent to 6 per state component) and the simulation for such a rate is shown in Figure 6. Note that asymptotic stability is indeed achieved.

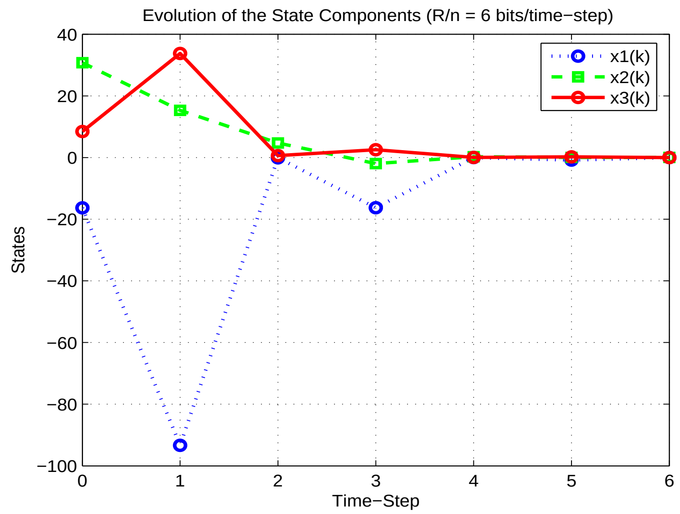

Figure 6. Closed-loop NCS (Type I): MIMO case using $R = 18$ bits/time-step

5.2. *Example 2*

In order to test the conservativeness of our results, we considered a single-input system given by

$$x(k+1) = \begin{bmatrix} 20 & 0 & 10 \\ 0 & 10 & 0 \\ 0 & 10 & 30 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(k)$$

We choose the initial condition to be $x(0) = \begin{bmatrix} 16.333 & 13.768 & -80.44 \end{bmatrix}'$. Using Corollary 4.1, we have $R = 54$ *bit/time-step*. We then verify in Figure 7 the asymptotic stability claims of the corollary. Since our results provide sufficient conditions only, we tried for smaller values of R and found out that for this particular example, $R = 42$ *bit/time-step* leads to instability, see Figure 8.

5.3. *Example 3*

Let us finally consider a system with time-delay $p = 2$ evolving according to the following dynamics

$$x(k+1) = \begin{bmatrix} 2 & 0 \\ 0 & 1.5 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k-2)$$

with the initial condition state vector $x(0) = \begin{bmatrix} -16.333 & 30.768 \end{bmatrix}'$. For this system, Theorem 4.2 gives a rate bounded below by $R = 28$ *bit/time-step*. The corresponding simulation is shown in Figure 9.

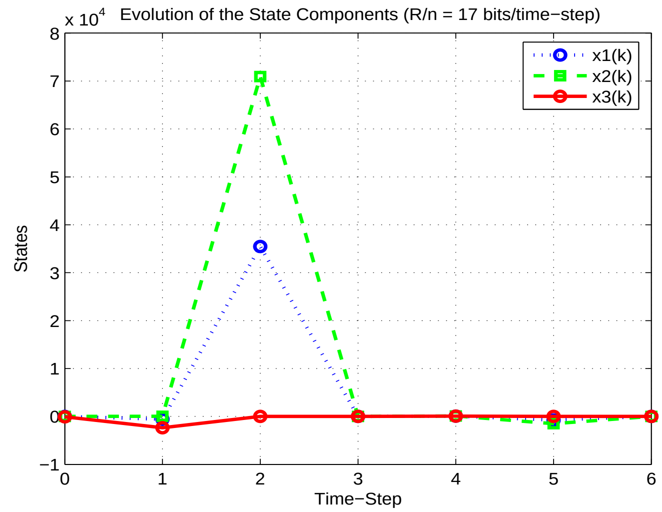

Figure 7. Closed-loop NCS (Type I): SISO case using $R = 54$ bit/time-step.

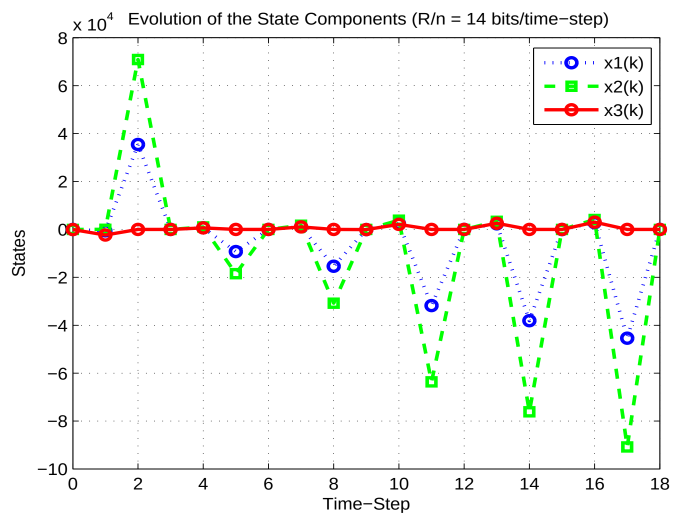

Figure 8. Closed-loop NCS (Type I): SISO case using $R = 42$ bits/time-step.

6. Removing the rate dependency on $\|A\|$.

The result of Theorem 4.1 (as well as Corollary 4.1 and Theorem 4.2) established a sufficient rate in terms of the norm of A. For different matrices A with the same eigenvalues however, this may lead to

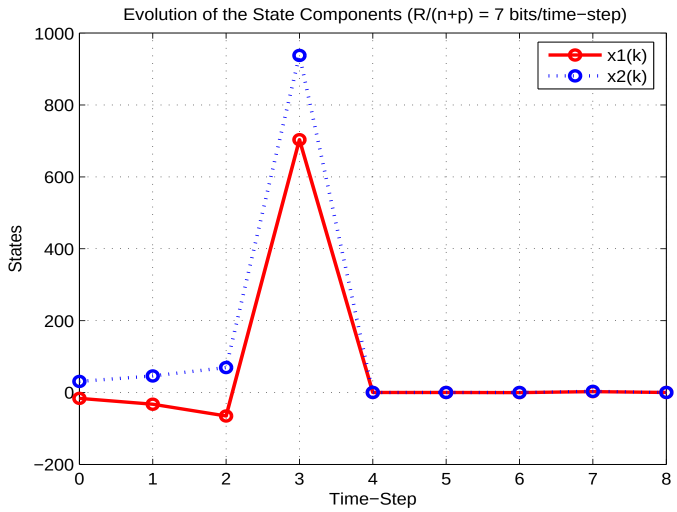

Figure 9. Closed-loop NCS with Time-Delay: Type 1

very different rates, some of which may also be very large compared to the minimum rates of the Data Rate Theorem [**?**]. For example, the following two matrices A have the same eigenvalues (therefore, the same minimum stabilization rate according to the Data Rate Theorem) but different norms (therefore, different sufficient rates according to Theorem 4.1):

$$A_1 = \begin{bmatrix} 2 & 100000 \\ 0 & 2 \end{bmatrix} \tag{17}$$

and

$$A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \tag{18}$$

Then $\|A_1\| = 1 \times 10^5$. and $\|A_2\| = 2$ but A_1 and A_2 have the same eigenvalues $\lambda = \{2, 2\}$. One way to remove this disadvantage is to change the control law used in the proof of Theorem 4.1. Instead of trying to asymptotically stabilize the state x, we attempt to stabilize the state $z = \Phi^{-1}x$, where Φ is a linear transformation such that $\Phi^{-1}A\Phi$ is the diagonal matrix equivalent to A (or more generally the Jordan-block matrix). The error $\varepsilon_z(0)$ in the z space is given by $\Phi^{-1}(x_j(0) - \overline{x}_j(0))$. For stabilization purposes, designing a control law to stabilize the state z is equivalent to stabilizing x since $z \longrightarrow 0$ implies $x \longrightarrow 0$. There will however be a difference in the transient response as we will see later. The change of variable implies that the control law in equation (9) no longer depends on the controllability matrix of the pair (A, B), i.e. ζ_μ. It will depend, however, on the controllability matrix of the pair $(\Phi^{-1}A\Phi, \Phi B)$, denoted by ζ_{Φ_μ}. Therefore, the new control law is given by

$$\mathbb{U}_n = -\zeta_{\Phi_\mu}^{-1}(\Phi^{-1}A\Phi)^\mu \Phi^{-1}\overline{x}(0); \tag{19}$$

and in the z space, after μ time-steps, we will have

$$z(\mu) = (\Phi^{-1}A\Phi)^\mu \varepsilon_z(0). \tag{20}$$

Then, from equations (20) and (7), and using the properties of matrix norms, we obtain

$$\begin{aligned}\|z(\mu)\| &= \|(\Phi^{-1}A\Phi)^{\mu}\varepsilon_z(0)\| \\ &\leqslant \|(\Phi^{-1}A\Phi)^{\mu}\|\|\varepsilon_z(0)\| \\ &\leqslant \sqrt{n}2^{-r_n}\|(\Phi^{-1}A\Phi)^{\mu}\|\|\Phi^{-1}\|L_0.\end{aligned}$$

Similarly, in order to force the state z to decrease in the norm (after μ steps), we shrink the upper bound of the state $z(\mu)$ by forcing it to be less than the lower bound of the state $z(0)$, i.e., $2^{-R_n}\sqrt{n}\|(\Phi^{-1}A\Phi)^{\mu}\|\|\Phi^{-1}\|L_0 < \|\Phi^{-1}\|L_0$. However, if $\Phi^{-1}A\Phi$ is a diagonal matrix then $\|(\Phi^{-1}A\Phi)^{\mu}\| = |\lambda_{max}|^{\mu}$ where λ_{max} is the eigenvalue of A with the largest magnitude. We can then replace in Theorem 4.1 the expression $R = n\left\lceil \log(\|A^{\mu}\|) + \frac{1}{2}\log n \right\rceil + n$ with

$$R = n\left\lceil \log(|\lambda_{max}|^{\mu}) + \frac{1}{2}\log n \right\rceil + n. \tag{21}$$

If on the other hand, matrix $\Phi^{-1}A\Phi$ is a Jordan-block matrix (as is the case for repeated eigenvalues of A), we can use the norm of $\Phi^{-1}A\Phi$ noting that $\|(\Phi^{-1}A\Phi)^{\mu}\| \approx |\lambda_{max}|^{\mu}$ which will in general be less than $\|A^{\mu}\|$. Therefore, the rate is no longer a function of the norm of A but rather a function of the largest eigenvalue of A. In general, this may lead to a lower sufficient rate for stabilizability, but with the possible deterioration in the transient response.

6.1. Example

The following simulation shows the evolution of x when using the control law given in equation (19) with the rate given by $R = n\left\lceil \log(|\lambda_{max}|^{\mu}) + \frac{1}{2}\log n \right\rceil + n$. Let us consider the following system:

$$x(k+1) = \begin{bmatrix} 2 & 100 & 100 \\ 0 & 4 & 100 \\ 0 & 1 & 4 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u(k)$$

Let initial condition to be $x(0) = \begin{bmatrix} 16.333 & 13.768 & -80.44 \end{bmatrix}'$. Using Equation (21), we find that $R = 42$ *bit/time-step* is now sufficient for stabilization. This was not the case using the control law depending on the controllability matrix of the pair (A,B). The simulation using this control law is shown in Figure 10. We also show in Figure 11 the simulation using the results of Theorem 4.1 and the rate was $R = 57$ *bit/time-step*. The tradeoff is evident when comparing the two simulations: although a lower rate is needed in the simulation in Figure 10, the transient response (overshoot, settling time) in Figure 11 is actually better.

7. A New Encoder/Decoder Design: Optimizing the Rate for Stabilization

In the previous sections we obtained sufficient rates with an easily implementable encoder-decoder scheme. Although such rates are larger than the ones given by the Data Rate Theorem, the implementation of our encoder-decoder requires less computational power than other published schemes. Specifically, the evolution of the quantizer in our scheme uses one scalar equation (equation (3)). On the other hand, encoder-decoder schemes such as the ones proposed in [5] or [16] achieve the minimum rate established by the Data Rate Theorem. These rates are achieved however with a higher computational cost since they require state-space predictors, the use of rotational matrices (to undo the

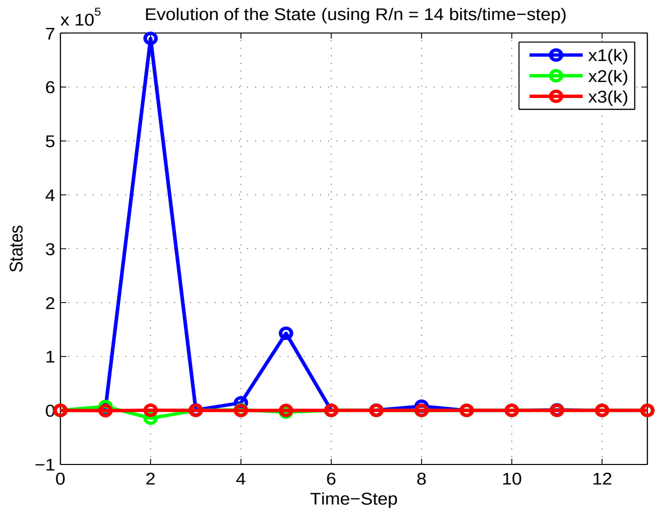

Figure 10. Closed-loop NCS (Type I): SISO case using $R = 14$ bit/time-step.

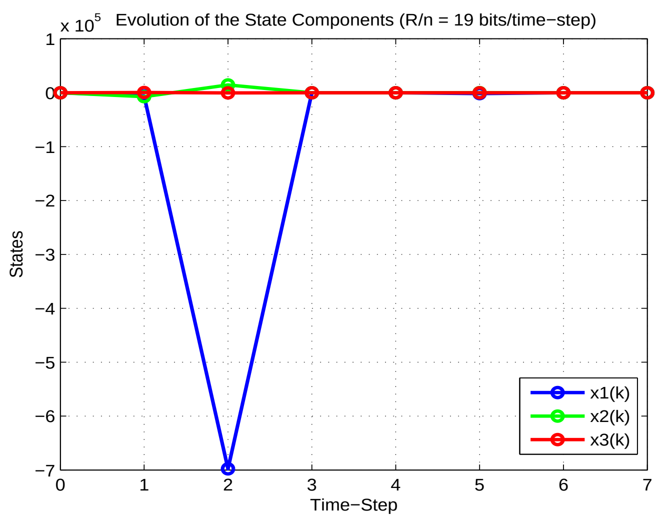

Figure 11. Closed-loop NCS (Type I): SISO case using $R = 19$ bit/time-step.

possible rotations caused by the A matrix), and the calculation of the centroid of the region that traps the state space variable.

In some scenarios, both the computational power and the rate may be constrained. Our purpose in this section is to design an encoder-decoder scheme that achieves a rate close to that provided by the

Data Rate Theorem, while using less computational power. The following builds upon ideas described in [7], [5], [16].

7.1. Encoder-Decoder Design

Let the initial state be bounded by some value L_0, i.e. $\|x\|_2 \leq L_0$. This equally-length side n-cube region will have 2^n vertices. This set of vertices is denoted by V_0, and each vertex is denoted by, v_0. We will allocate r_i bits for the state space component x_i, $\forall i \in \{1, 2, \ldots, n\}$. We will necessarily consider that $r_i > \log_2 |\lambda_i|$, where λ_i is the eigenvalue associated with x_i. This last assumption is imposed by the Data Rate Theorem and cannot be removed if we want to achieve stabilization. Each rate r_i must be an integer number such that $r_i \geq \lceil \log_2 |\lambda_i| \rceil$. We introduce a matrix Q_R

$$Q_R = \begin{bmatrix} 1/2^{r_1} & 0 & \ldots & 0 \\ 0 & 1/2^{r_2} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1/2^{r_n} \end{bmatrix}. \tag{22}$$

Moreover, we will assume that $r_1, r_2, \ldots, r_n$ are such that the matrix $A_Q = AQ_R$ is a stable matrix (we will show later how to accomplish this goal). In the following, we assume that the plant is deterministic and undriven as described by $x(k+1) = Ax(k)$. The controller design problem will be discussed in subsection 7.2. The first step is to generate a n dimensional cube centered at the origin with sides of length $2L_0$. The center of this first quantizer will be labeled $C_Q(0)$. The uncertainty region is divided in 2^{r_1} subregions in the x_1 direction, 2^{r_2} subregions in the x_2 direction, and so on until we obtain 2^{r_n} subregions for the x_n direction. After one time step, the state will land in one of these smaller n dimensional cubes and the total of small cubes will be $2^{r_1+r_2+\ldots+r_n}$. Therefore, the number of bits needed to represent all the cube centroids is $R = r_1 + r_2 + \ldots + r_n$ which is the actual rate in bits/time-step. After determining in which cube the state has landed, we calculate the centroid of this smaller cube. This centroid will be chosen as the encoder estimate of the state, $\overline{x}(0)$. The binary symbol, s, that represents $\overline{x}(0)$ is transmitted to the receiver. Note that the error between the state and the state estimate, $\varepsilon(0)$, lies in the region $\{[-L_0/2^{r_1}, L_0/2^{r_1}], [-L_0/2^{r_2}, L_0/2^{r_2}], \ldots, [-L_0/2^{r_n}, L_0/2^{r_n}]\}$. This is the key property of this quantizer. Figure (12) shows an example of a two dimensional quantizer with $r_1 = 1$ and $r_2 = 3$. The encoder and decoder will evolve the center of the quantizer, C_Q at time $k+1$:

$$C_Q(k+1) = A\hat{x}(k) \tag{23}$$

This new center is used to generate an uncertainty region that may be divided into another $2^{r_1+r_2+\ldots+r_n}$ subregions with the same 2^{r_i} subregions in the direction x_i direction as explained before. At time $k+1$, the length of each n sides is determines by the n quantities $\Delta_i(k+1)$. The sides of the $k+1$ box are determined using the matrix A_Q and the vertices v_0 of the original uncertain n-dimensional cube. The length of the side parallel to the x_i direction at time $k+1$ is given by:

$$\Delta_{x_i} = \max_{v_0} |(A_{Q,i})^{k+1} V_0|, \forall\, v_0 \in V_0. \tag{24}$$

where $A_{Q,i}$ is the "i-th" row of matrix A_Q. Equation (24) evaluates the maximum over absolute values, therefore, we can guarantee that the state $x(k+1)$ at time $k+1$ will land in an n-dimensional box (not necessarily a cube) that is centered on $C_Q(k+1)$ and with sides of length $2\Delta_{x_i}$ in the x_i direction. In other words, the hyper-planes that are perpendicular to the x_i component direction will be located at $-\Delta_{x_i}$ and Δ_{x_i} units from $C_Q(k+1)$ in the x_i direction. The new uncertainty box, will again be divided

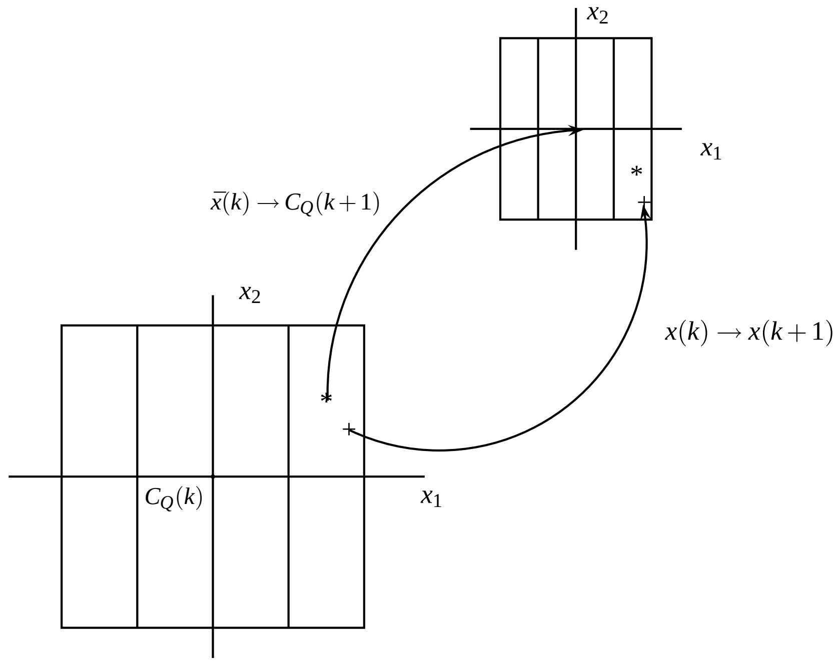

Figure 12. Quantizer evolution sample including centroid, state and state estimator.

into $2^{r_1+r_2+\ldots+r_n}$ boxes with 2^{r_i} in the x_i direction. We label these small boxes with binary symbols (a total of $2^{r_1+r_2+\ldots+r_n}$ binary symbols). We then determine in which of these boxes the actual state, $x(k+1)$, lies and use the centroid of this specific box as the state estimate $\bar{x}(k+1)$ at time $k+1$. We again transmit the binary symbol, s, that corresponds to the box where the state lies. Because of the way we have constructed this quantizer and since A_Q was assumed to be stable, the uncertainty box keeps on shrinking as k tends to infinity, which guarantees that our state estimate reaches the actual state and that $\|\varepsilon\|$ tends to zero. Note that both encoder and decoder must know the original size L_0 of the uncertainty as well as the exact dynamics of the plant. Also, both encoder and decoder must be able to compute the equations (23) and (24). This guarantees the equimemory property. The only remaining issue is to guarantee that A_Q is stable. This can be done by a trial and error procedure as follows:

1. Set $r_i = \lceil \log_2 |\lambda_i| \rceil \ \forall i \in \{1,2,\ldots,n\}$, where λ_i is the eigenvalue associated with the state component x_i and r_i are the bits/time-step allocated to x_i.
2. Using rates r_i, we form the matrix Q_R and obtain the eigenvalues of $A_Q = AQ_R$.
3. Check that all such eigenvalues are inside the unit circle, i.e., $|\lambda_{A_Q}| < 1$.
4. If $|\lambda_{A_Q}| < 1$, stop and use the rates r_i for transmission. If $|\lambda_{A_Q}| \geq 1$, we add 1 to each r_i in Q_R that corresponds to an eigenvalue of A_Q that was outside of the unit circle and return to step 2.

We test this algorithm in the following example. Given the following matrix, find r_1 and r_2 such that A_Q is stable.

$$A = \begin{bmatrix} 2 & 0.5 \\ 3 & 4 \end{bmatrix}.$$

Prepared using rncauth.cls

Since the eigenvalues of A are $\lambda_A = \{1.418, 4.581\}$ we choose $r_1 = 1$ and $r_2 = 3$. Then, Q_R is given by

$$Q_R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.125 \end{bmatrix}. \quad (25)$$

We then obtain the eigenvalues of A_Q: $\lambda_{A_Q} = \{1.14, 0.35\}$. Since one of them is outside the unit circle, and corresponds to the row where r_1 is located, we add 1 to r_1 to obtain $r_1 = 2$. We obtain the new eigenvalues of . The new eigenvalues of A_Q with the updated Q_R are $\lambda_{A_Q} = \{0.761, 0.283\}$. A_Q is now stable, and the values $r_1 = 2$ and $r_2 = 3$ may be used as the rates for transmission.

7.2. *Adding a controller for stabilization*

We have seen in the previous subsection that by expanding more computational power, a lower stabilization rate may be achieved. Designing an asymptotically stabilizing controllers is however not obvious. In order to address this issue, consider the system described by

$$x(k+1) = Ax(k) + Bu(k).$$

We next consider this system in the encoder/decoder computations and modify equations (23) and (24)accordingly. The new equations are

$$C_Q(k+1) = A\hat{x}(k) + Bu(k) \quad (26)$$

and

$$\Delta_{x_i} = \max_{v_0} |(A_{Q,i})^{k+1} V_0|, \; \forall\, v_0 \in V_0. \quad (27)$$

where B_i is the "i-th" row of vector B. To use these new equations we assume that the encoder/decoder have access to the control signal or that it may be computed locally. The derivations of the previous subsection remain valid since the addition of the control law, only represent a *translation* of the centroid of the quantizer. At this point the simplest controller is the estimated state linear feedback controller, $u(k) = K_c\overline{x}(k)$, which is motivated by the following Lemma found in [16].

Lemma 7.1. *[16] Let A_s be a stable matrix. Let Bs_k a set of matrices such that $\|Bs_k\| \leq L$ and the limit $\lim_{k\to\infty} Bs_k \to 0$. Let $S_t = \sum_{i=0}^{k-1} A^{k-1-i} Bs_i$ then $\lim_{k\to\infty} S_k \to 0$.*

If a K_c is found such that $A - BK_c$ is stable, the evolution of the state will be given by

$$x(k) = (A - BK_c)^k \overline{x}(o) + \sum_{i=0}^{k-1} (A - BK_c)^{k-1-i} BK_c \varepsilon(i) \quad (28)$$

Our encoder/decoder scheme guarantees that $\|\varepsilon(i)\| \leq \|(A_{Q,i})^{k+1} V_0\|$ and that $\|\varepsilon(i)\|$ tends to zero when i grows. If we let $A_s = A - BK_c$ and $Bs_k = BK_c\varepsilon(k)$, then we may apply Lemma 7.1. Therefore, any stabilizing K_c will guarantee asymptotic stabilization of the system using the rates obtained before since the first additive term in equation (28) tends to zero since $A - BK_c$ is stable, and the second additive term tends to zero by Lemma 7.1. We present next an example considering the following system:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 2 & 2.5 \\ 3 & 4 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k) \\ u(k) &= -\begin{bmatrix} 2.533 & 2.566 \end{bmatrix} u(k) \end{aligned}$$

This K_c allocates the poles of $A - BK_c$ at 0.5 and 0.4. Since the A matrix of this system is the same used in the example of the previous subsection, we know that the rates that stabilize A_Q are $r_1 = 2$ and $r_2 = 3$. This give us a total rate of $R = 5$ bits/time-step. Using the encoder/decoder scheme proposed we obtain the plots in Figure 13. The exponential decrease of the error in shown in Figure 14.

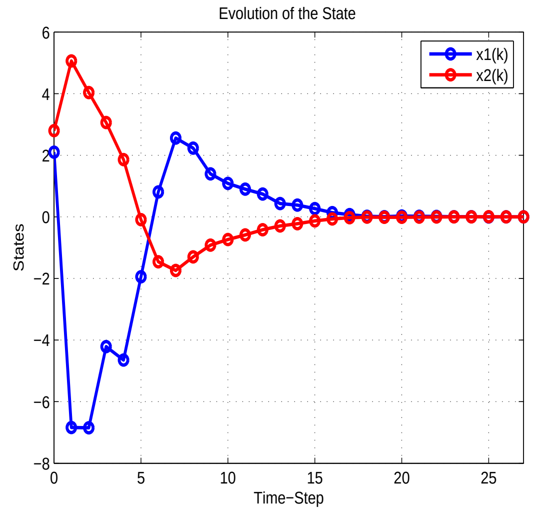

Figure 13. State evolution in NCS Type I using $R = 5$ bits/time-step

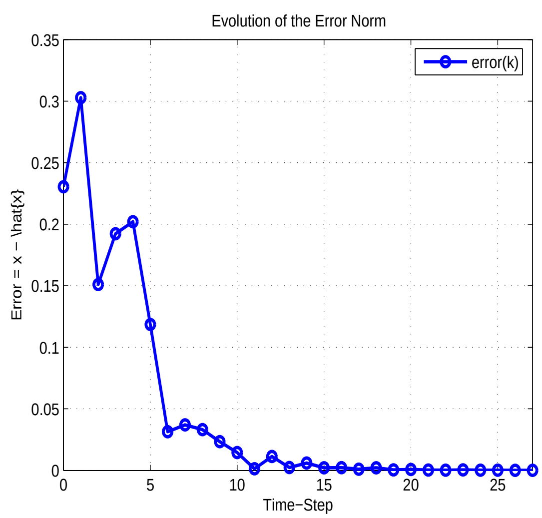

Figure 14. Evolution of the Error Norm in NCS Type I using $R = 5$ bits/time-step

8. Conclusions and Future Work

This paper has extended previous results for determining the sufficient rate for stabilization of a packet-based networked control system. While the rates obtained for Network Type I are higher that the limits given by the Data Rate Theorem, the computational cost of our scheme is simpler than earlier proposed schemes. In this setup we were able to include to treat the case of a constant time delay in the network.

We also obtained sufficient rates for stabilizing a system using a Type II Network. In order to lower the required transmission rates, we proposed a more complex encoder/decoder scheme that achieves rates close to those specified by the Data Rate Theorem.

Future work will include the inclusion of time delays in a Network Control System Type II, and the extension of the general case of m inputs of this type of closed-loop system. Other ideas for future work include dealing with noise in the loop and the generalization to the case of packet drops and saturation in the control signal.

REFERENCES

1. P. Antsaklis and A. N. Michel. *Linear Systems*. Mc Graw Hill, 1997.
2. N. Elia and S. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, September 2001.
3. G. W. Kim, K. Ji, and A. Ambike. Networked real-time control strategies dealing with stochastic time delays and packet losses. *Proceedings of the American Control Conference*, 2005.
4. F. Lian, J. Moyne, and D. Tilbury. Time delay modeling and sample time selection for networked control systems. *Proceedings of the ASME-DSC*, 10:1–8, 2001.
5. Q. Ling and M. Lemmon. Stability of quantized control systems under dynamic bit assignment. *IEEE Transactions on Automatic Control*, 50(5):734–739, May 2005.
6. S. Mitter. Control with limited information: the role of systems theory and information theory. *Proceedings IEEE Int. Sympt. Information Theory*, 7:1–23, 2001.
7. L. A. Montestruque and P. Antsaklis. Stability of model-based networked control, systems with time-varying transmission times. *IEEE Transactions on Automatic Control*, 49(9):2876–2880, September 2004.
8. J. Nilsson. *Real-Time Control Systems with Delays*. Department of Automatic Control, Lund Institute of Technology, 1998.
9. J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.
10. A. Sahai. Evaluating channels for control: Capacity reconsidered. *Proceedings of the American Control Conference*, 4:2358–2362, 2000.
11. S. V. Sarma. *Finite-Rate Control: Stability and Performance, Ph.D. Dissertation*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2006.
12. L. Shi, M. Epstein, and R. Murray. Networked control systems with norm bounded uncertainties: A stability analysis. *Proceedings of the American Control Conference*, pages 2783–2788, 2006.
13. L. Shi and R. Murray. Towards a packet-based control theory - part i: Stabilization over a packet-based network. *Proceedings of the American Control Conference*, 2:1251–1256, 2005.
14. L. Shi and R. Murray. Towards a packet-based control theory - part ii: Rates issues. *Proceedings of the American Control Conference*, pages 3482–3487, 2006.
15. S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Transactions on Automatic Control*, 49(7):1196–1201, 2004.
16. S. Tatikonda and S. Mitter. Control under communication constraints. *IEEE Transactions on Automatic Control*, 49(7):1056–1068, 2004.
17. M. B. W. Zhang and S. Philips. Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1):84–99, February 2001.
18. G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *Proceedings of the American Control Conference*, 4:2876–2880, 1999.
19. 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.