Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

ThIP12.1

Level Crossing Sampling in Feedback Stabilization under Data-Rate Constraints

Ernesto Kofman and Julio H. Braslavsky

Abstract— This paper introduces a novel event-driven sampled-data feedback scheme where the plant output samples are triggered by the crossings—with hysteresis—of the signal through its quantization levels. The plant and controller communicate over binary channels that operate asynchronously and are assumed to be error and delay-free. The paper proposes two systematic output feedback control design strategies. The first strategy consists in the digital emulation of a previously designed analog controller. The second strategy is a simple direct design that drives the plant state to the origin in finite time after a total transmission of $2n + 2$ bits, where n is the order of the plant.

I. INTRODUCTION

A number of recent works consider the problem of feedback stabilization over a communication channel with data-rate constrains (see [1] for an overview). A fundamental result in this area states that there exist a positive lower bound on the data-rate [bits per second] required for closed loop stability of a linear time invariant, continuous-time plant with poles in the open right-half plane [2], [3].

An underlying assumption for this fundamental result is that the output of the plant is sampled in an essentially *time-driven* fashion; be it uniform and periodic, as in [2], or nonuniform, as in [4], [5]. Namely, the samples are obtained largely independently of the plant dynamic evolution.

Event-driven sampling schemes, such as *Lebesgue* sampling, have been considered as alternatives to conventional time-driven sampling schemes [6], [7], [8]. In Lebesgue sampling, samples are triggered by the crossing of the signal through quantization levels, which presents a number of potential advantages for networked control, such as clock-free operation, low data-rate requirements [6], and higher energy efficiency in digital-to-analog conversion [9]. On the other hand, a main disadvantage is that control analysis and design become more difficult, thus little theory for event-driven sampled-data systems seems available, in contrast with the well-developed theory for time-driven periodic sampled-data systems [10].

The present paper explores the use of a level-crossing sampling (LCS) scheme based on *hysteretic quantization* for feedback stabilization. LCS may be viewed as a Lebesgue sampling scheme in which the quantizer includes hysteresis. Hysteresis allows us to implement 1-bit coding feedback communication in the proposed control strategies, which has the potential to achieve the most efficient data-rates

[5]. In addition, under noisy scenarios, hysteresis would also minimize spurious sampling, further contributing to low data-rate feedback communication.

Two asynchronous control design strategies based on LCS are introduced: 1) an event-driven "emulation" of a previously designed analog controller, and 2) a "direct" design using a combined deadbeat estimation and control scheme. The "emulated" controller is implemented as a *quantized state system* [11] that approximates the original analog controller. If the latter achieves closed-loop asymptotic stability, we show that its "emulation" guarantees ultimate boundedness of the trajectories around the origin. We also estimate an upper bound for the difference between the closed-loop trajectories obtained by both controllers.

The "direct" design control strategy consists in two sequential open-loop procedures: (i) deadbeat estimation of the unknown initial condition, (ii) deadbeat bang-bang control to the origin. Assuming no disturbances in the plant outputs, this strategy drives the plant state to the origin in *finite time* under the constraint of 1-bit (delay and error-free) transmissions, both between sensor and controller, and between controller and actuator. Consequently, under the idealized conditions assumed, the scheme achieves stability with effectively *zero* average data-rate, appearing to circumvent the data-rate limitations known for feedback over communication channels based on time-driven sampling. However, noisy measurements, plant uncertainties and time delays will certainly impose a non-zero average data-rate, as we discuss in Section IV-D.

In Section II we present our assumptions and the scheme under study. Section III presents the proposed asynchronous emulation design, gives its main properties and introduces a numerical simulation example. Section IV presents the proposed direct design strategy and poses a discussion on the issue of stabilization under data-rate constraints with conventional sampling, and with the proposed LCS design strategy. Section V summarizes the paper conclusions.

II. LEVEL CROSSING SAMPLING SCHEME

The general feedback scheme considered is shown in Figure 1. We model two communication links: a sensor link, between the measured plant output and the controller input, and an actuator link, between the controller output and plant input. These communication links are assumed to be error-free and have no noise disturbances nor transmission delays.
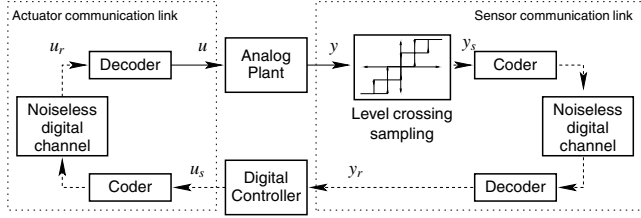
Fig. 1. Event-driven sampled-data scheme for feedback stabilization

Plant.: The plant is a continuous-time, finite dimensional LTI system given by the minimal realization

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t). \qquad (1)$$

The matrices A, B, C are assumed known, but the system initial condition $x(0)$ is unknown.

Actuator.: The actuator can only produce values in the set $\mathscr{U} \triangleq \{-pU, \cdots, -U, 0, U, \cdots, pU\}$ where U is a positive real constant and p is a positive integer. We assume that the switching of the actuator is instantaneous and the values are held constant (in a zero-order-hold fashion) until a different value of $u(t)$ is generated.

Level crossing sampling with hysteresis: Given $h > 0$ and a continuous function $y(t) : \mathbb{R} \to \mathbb{R}$, we define the *level-crossing sampled sequence* $\{y_s(t_k)\}_{k=0}^{\infty}$ *with quantization interval h* by the piecewise constant function $y_s(t) : \mathbb{R} \to \mathbb{R}$,

$$y_s(t) = \begin{cases} \lfloor y(t_0)/h \rfloor h, & \text{if } t_0 \le t < t_1, \\ y(t_k), & \text{if } t_k \le t < t_{k+1}, \end{cases} \qquad (2)$$

where $\lfloor \cdot \rfloor$ denotes integer part, and the sampling instants $\{t_k\}_{k=1}^{\infty}$ defined by

$$t_k = \inf\{\tau \in (t_{k-1}, \infty) : |y(\tau) - y_s(t_{k-1})| > h\}. \qquad (3)$$

¿From (2) and (3) we see that the samples $y_s(t_k)$ are triggered by the crossings (with hysteresis) of the signal $y(t)$ through quantization levels regularly spaced by h. We denote by $Q_h\{\cdot\}$ the *hysteretic quantization operator* with quantization interval h defined by $y_s(t) = Q_h\{y\}(t)$. Note that $Q_h\{\cdot\}$ is a *dynamic* nonlinearity, because of the inbuilt hysteresis in the definition of the sequence $\{t_k\}$ in (3).

The LCS device in Figure 1 generates a new sample of the output $y(t)$ whenever $y_s(t) = Q_h\{y\}(t) \ne y_s(t^-)$. Because successive samples always differ in $\pm h$, each sample can be coded using only one bit, thus reducing the amount of information that needs to be transmitted. In this sense, the LCS operation may be likened to an "event-driven" differential pulse-code modulation.

Coder/Digital Channel. Sensor link: We assume that the digital channel is memory-less and error-free, and can only transmit one bit, 0 or 1, per asynchronous sample produced by the LCS device. The coding strategy is the following: When a sample is produced, at say $t = t_k$, the channel transmits

$$\begin{cases} 1, & \text{if } y_s(t_k) > y_s(t_{k-1}), \\ 0, & \text{if } y_s(t_k) < y_s(t_{k-1}). \end{cases} \qquad (4)$$

Thus, a 1 is transmitted if the current sample of the output has increased with respect to its previous sample, or a 0 if it has decreased. If no samples are produced, no transmission takes place. However, notice that in the proposed scheme there is also information when no samples are produced; namely, the output $y(t)$ remains within its quantization band.

Decoder. Sensor link.: The decoder receives the sequence of bits indicating the changes in y_s and calculates

$$y_r(t_k) = \begin{cases} h, & \text{when 1 is received}, \\ -h, & \text{when 0 is received}. \end{cases} \qquad (5)$$

Notice that

$$y_r(t_k) = y_s(t_k) - y_s(t_{k-1}). \qquad (6)$$

Digital controller: The controller receives the asynchronous sequence of values $y_r(t_k)$ and produces a sequence of control actions $u_s(\tau_j)$, where the time sequences $\{t_k\}$ and $\{\tau_j\}$ are different in general. We consider $u_s(\tau_j) \in \mathscr{U}$, and $u_s(\tau_0) = 0$. We also restrict $|u_s(\tau_j) - u_s(\tau_{j-1})| \in \{U, 2pU\}$, that is, successive control actions differ in one quantization level or they can jump between the saturation limits. This restriction permits coding the successive control values using only one bit. In the next sections we shall propose two different strategies to calculate the sequence $u_s(\tau_j)$.

Coder/digital channel. Actuator link: The coding strategy is similar to that of the sensor link, transmitting only when the controller produces a control signal that differs from its value at the previous sampling instant. Then, when a sample is produced at $t = \tau_j$, the channel transmits

$$\begin{cases} 1, & \text{if } u_s(\tau_j) - u_s(\tau_{j-1}) \in \{-2pU, U\} \\ 0, & \text{if } u_s(\tau_j) - u_s(\tau_{j-1}) \in \{-U, 2pU\} \end{cases} \qquad (7)$$

Decoder. Actuator link: The actuator decoder receives the sequence of bits informing the changes in u_s, and it builds the signal $u_r(\tau_j)$ according to following logic

$$\begin{cases} u_r(\tau_{j-1}) + U, & \text{if } u_r(\tau_{j-1}) < pU \text{ and 1 is received} \\ u_r(\tau_{j-1}) - U, & \text{if } u_r(\tau_{j-1}) > -pU \text{ and 0 is received} \\ -pU, & \text{if } u_r(\tau_{j-1}) = pU \text{ and 1 is received} \\ pU, & \text{if } u_r(\tau_{j-1}) = -pU \text{ and 0 is received} \end{cases}$$

Provided that $u_r(\tau_0) = u_s(\tau_0) = 0$, we have $u_r(\tau_j) = u_s(\tau_j)$.

III. CONTROLLER DESIGN BY EMULATION

We next introduce a method for designing a digital control law as an approximation of a previously designed continuous-time controller. The approximation is carried out by implementing the analog controller as a *quantized state controller*. Such emulation approach was introduced in [11], and is based on a numerical integration algorithm called the Quantized State System (QSS) method [12]. The QSS method can be seen as the state discretization counterpart of Euler's time discretization.

4424

A. An Introductory Example

Consider the unstable plant

$$\dot{x}(t) = x(t) + u(t); \quad y(t) = x(t), \tag{8}$$

which is asymptotically stabilized by the PI controller

$$\dot{z}(t) = y(t); \quad u(t) = -2y(t) - z(t). \tag{9}$$

We replace the control law (9) by the QSS approximation

$$\dot{z}(t) = y_s(t); \; y_c(t) = -2y_s(t) - \zeta(t), \tag{10}$$

where

$$y_s(t) = Q_h\{y\}(t) \quad \text{and} \quad \zeta(t) = Q_g\{z\}(t) \tag{11}$$

are variables quantized with intervals h and g.

Let us assume that $\mathscr{U} = \{\cdots, -4, -2, 0, 2, 4, \cdots\}$; that is, $U = 2$ in $\mathscr{U}$ as defined in Section II. Then, $u_s(t)$ is calculated as the nearest element of $\mathscr{U}$ from $y_c(t)$.

We select the quantization intervals in (11) $h = 0.5$ and $g = 0.1$. Consider the initial conditions $x(0) = 10$ and $z(0) = \zeta(0) = 0$; the system will then evolve as follows:

- At $t = 0$ the controller receives the value $y_s(t) = 10$. According to (10) we have $y_c(t) = -20$ and $\dot{z}(t) = 10$. Thus, $u_s(t) = -20$. These values remain constant until a new sample y_s is received or until $\zeta(t)$ changes. The next change in $\zeta(t)$ is produced when it differs from $z(t)$ in 0.1. Since $z(t)$ grows with a slope of 10, that will happen when $t = 0.1/10$. Thus, the controller schedules the next change in $\zeta(t)$ for $t = 0.01$.
- At $t = 0.01$ we have $\zeta(t) = z(t) = 0.1$, $y_c(t) = -20.1$ and $u(t)$ remains equal to -20. The slope $\dot{z}(t)$ also remains constant and next change in $\zeta(t)$ is then scheduled for $t = 0.02$.
- At $t = 0.02$, $\zeta(t) = z(t) = 0.2$, $y_c(t) = -20.2$ and the remaining variables do not change. After two identical steps, we have $t = 0.04$ and $\zeta(t) = z(t) = 0.4$, $y_c = -20.4$ and the next change is scheduled for $t = 0.05$.
- However, when $t = 0.0488$, it turns out that $y(t) = 9.5$ and a new sample $y_s = y = 9.5$ is then sent to the controller. Thus, $y_c(t) = -19.4$ and then $u_s(t)$ remains equal to -20. The controller state can be calculated as $z(t) = 0.4 + 0.0088 \times 10 = 0.488$, while its new slope becomes 9.5. Thus, the next change in $\zeta(t)$ will occur after $(0.05 - 0.0488)/9.5 = 0.000126$ units of time and the calculations continue in the same way.

The quantized controller (10) works as a discrete event system: it receives a sequence of values $y_s(t)$ and it schedules *internal* changes (when $\zeta(t)$ changes) as well as generating *output* events (when $u_s(t)$ changes).

As we can see from the initial sequence of events in the evolution of the closed-loop system (8), (10), the calculations inside the controller are very simple and can be implemented in a digital system. As we shall show, the emulated controller (10) behaves similarly to the continuous controller (9), in the sense that the closed-loop trajectories

generated will be close to those that would be generated with the original analog controller.

We next generalize and more formally define the idea behind the controller approximation in this example.

B. Quantized State Controller (QSC)

Let the continuous-time controller previously designed for the plant (1) be given by the state realization

$$\begin{aligned} \dot{z}(t) &= A_c z(t) + B_c y(t) \\ u(t) &= C_c z(t) + D_c y(t), \end{aligned} \tag{12}$$

and assume that the continuous-time closed loop system (1), (12) is asymptotically stable. We also assume that the state realization (12) is in the observer canonical form [13].

The QSC approximation of the controller (12) is a continuous-time system defined by the state equations

$$\begin{aligned} \dot{z}(t) &= A_c \zeta(t) + B_c u_c(t) \\ y_c(t) &= C_c \zeta(t) + D_c u_c(t), \end{aligned} \tag{13}$$

where $z \in \mathbb{R}^{n_c}$, and $\zeta \in \mathbb{R}^{n_c}$ is the component-wise quantized version of the state vector z, that is,

$$\zeta = \begin{bmatrix} Q_{g_1}\{z_1\} & Q_{g_2}\{z_2\} & \cdots & Q_{g_{n_c}}\{z_{n_c}\} \end{bmatrix}^T, \tag{14}$$

where each $Q_{g_i}\{\cdot\}$ is a quantization function with hysteresis, with quantization intervals g_i, $i = 1, \ldots, n_c$.

Since the plant output is only available from the received samples $y_r(t_k) = y_s(t_k) - y_s(t_{k-1})$, we can calculate

$$u_c(t) = u_c(t_{k-1}) + y_r(t_k), \quad \text{for } t_k \le t < t_{k+1}. \tag{15}$$

Then, provided that the sequence $y_r(t_k)$ includes information about the initial output $y(t_0)$ (e.g., by allowing the transmission of m bits during initialization to inform that the initial output is $\pm mh$) we have $u_c(t_k) = y(t_k)$.

Since $u_c(t)$ is piecewise constant, the components of $\zeta(t)$ and $y_c(t)$ are also piecewise constant and the components of $z(t)$ are piecewise linear and continuous.[1]

Note that: (i) the calculation of the resulting output sequence $y_c(\tau_j)$ does not require numerical integration as (13) is already a discrete system, and (ii) each component $\zeta_i(t)$ of $\zeta(t)$ only changes when $|\zeta_i(t_j) - z_i(t_j)| = g_i$. Since $x_i(t)$ is continuous, this implies that $|\zeta_i(t_j) - \zeta_i(t_{j-1})| = g_i$. By taking into account that System (12) is in the observer canonical form, it follows that $y_c(t) = \zeta_1(t) + Du_c(t)$ and then $y_c(t)$ changes with jumps of either $\pm g_1$ (when ζ_1 changes) or $\pm D_c h$ (when u_c changes).

We have assumed that the controller output $u_s(t) \in \mathscr{U}$. To accomplish this condition, define $u_s(t)$ as the nearest element of $\mathscr{U}$ from $y_c(t)$. Then, if $g_1 \le U$ and $D_c h \le U$, we have that $u_s(\tau_j) - u_s(\tau_{j-1}) = \pm U$, and we can apply the coding and decoding strategies proposed in Section II.

---

[1] System (13) is equivalent to a *discrete event system* in terms of the DEVS formalism [14]; see [12] for a DEVS model equivalent to (13).

C. Properties of the QSC Scheme

By defining $\Delta z(t) \triangleq \zeta(t) - z(t)$, $\Delta u(t) \triangleq u(t) - y_c(t)$ and $\Delta y(t) \triangleq u_c(t) - y(t)$, we can rewrite Equation (13) as

$$\dot{z}(t) = A_c(z + \Delta z) + B_c(y + \Delta y) \qquad (16)$$
$$u(t) = C_c(z + \Delta z) + D_c(y + \Delta y) + \Delta u(t).$$

Notice that $|\Delta y(t)| \le h$ and $|\Delta z_i| \le g_i$ for $i = 1, \cdots, n_c$.

If we assume also that the controller output $y_c(t)$ cannot reach the saturation bounds $\pm pU$ (because p is sufficiently large), then it is also true that $|\Delta u(t)| \le U$.

The closed loop system (1), (16) can be written as

$$\dot{w}(t) = \tilde{A}(w(t) + \Delta w(t)) + F\Delta v(t) \qquad (17)$$

where

$$w(t) \triangleq \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}; \Delta w(t) \triangleq \begin{bmatrix} 0_{n \times 1} \\ \Delta z(t) \end{bmatrix}; \Delta v(t) \triangleq \begin{bmatrix} y(t) \\ u(t) \end{bmatrix},$$

and

$$\tilde{A} = \begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{bmatrix}; \quad F = \begin{bmatrix} BD_c & B \\ B_c & 0 \end{bmatrix}.$$

Since the ideal closed loop system (1), (12) is

$$\dot{w}(t) = \tilde{A}w(t), \qquad (18)$$

we have that (17) is just a perturbed version of (18). Thus, the absolute value of each component of the perturbation terms $\Delta w(t)$ and $\Delta v(t)$ in (17) is bounded by either h, g_i or U. To formalize, write

$$|\Delta w(t)| \le g \triangleq \begin{bmatrix} 0_{n \times 1} \\ g_1 \\ \vdots \\ g_{n_c} \end{bmatrix}; \quad |\Delta v(t)| \le \gamma \triangleq \begin{bmatrix} h \\ U \end{bmatrix} \qquad (19)$$

where $|\cdot|$ denotes the component-wise module and the inequality sign $\le$ denotes a component-wise inequality.

Theorem 1: Suppose that the closed loop system (18) is asymptotically stable and let $w(t)$ be its solution from the initial condition $w(0) = w_0$. Let $\tilde{w}(t)$ be the solution of (17) from the same initial condition w_0 and let $\Lambda = V^{-1}\tilde{A}V$ be a modal decomposition of $\tilde{A}$. Then, for all $t \ge 0$,

$$|\tilde{w}(t) - w(t)| \le |V| \times$$
$$\left(|\mathbb{Re}(\Lambda)^{-1}\Lambda||V^{-1}|g + |\mathbb{Re}(\Lambda)^{-1}V^{-1}F|\gamma \right). \qquad (20)$$

Proof: Define $\varepsilon(t) \triangleq \tilde{w}(t) - w(t)$. Then, subtracting (18) from (17) we obtain

$$\dot{\varepsilon}(t) = \tilde{A}(\varepsilon(t) + \Delta w(t)) + F\Delta v(t)$$

with $\varepsilon(0) = 0$. The perturbation terms verify (19).

If $\tilde{A}$ is diagonalizable, Theorem 3 of [15] concludes

$$|\varepsilon(t)| \le |V| \times (|\mathbb{Re}(\Lambda)^{-1}\Lambda||V^{-1}|g + |\mathbb{Re}(\Lambda)^{-1}V^{-1}F|\gamma)$$

for all $t \ge 0$. Otherwise, the same result is derived following a procedure developed in Theorem 3.3 of [16]. ∎

This theorem says that the state trajectory of the plant with the quantized state controller remains close to the state trajectory of the plant with the original continuous controller. Since the state trajectories go to zero, the QSC trajectories go to a bounded region around the origin whose size is calculated by the right hand side of Eq.(20). The existence of the ultimate region is ensured irrespective of the discretization parameters h, U, and g_i, $i = 1, \cdots, n_c$ (they only affect the size of that region).

D. Example Revisited

Figure 2 shows simulation results of the System (8),(10). After $t = 7.0823$ the plant reaches the origin. In that moment the controller produces the input $u = 0$ and then the system remains in that state without producing more samples or calculations at the controller. The total number of samples y_s received by the controller were 41, while the number of samples u sent by the controller were 28. Since each change can be coded with only one bit, the number of transmitted bits was 61 in the sensor link (we added here 20 bits to transmit the initial value of y_s) and 28 in the actuator link.
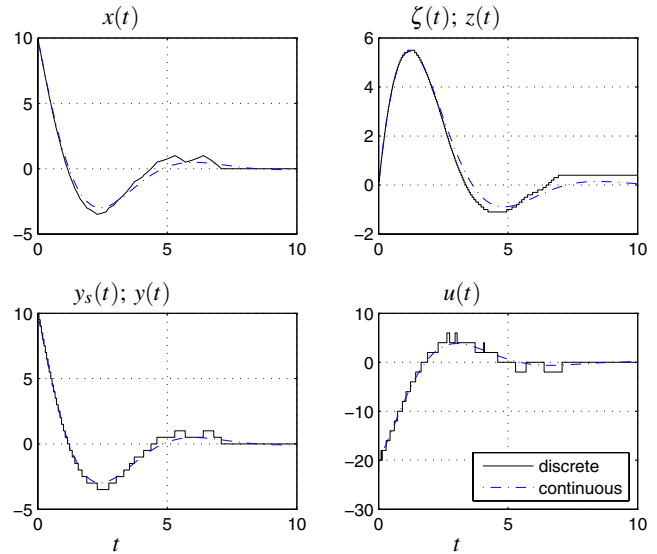

Fig. 2. Clockwise from top left corner: plant state $x(t)$; controller state $\zeta(t)$; plant input $u(t)$; and controller input $y_s(t)$ and plant output $y(t)$. Solid lines indicate trajectories with the emulated controller; dash-dot lines indicate those with the target continuous-time controller.

Using Theorem 1 we can prove that the trajectories of the plant with the continuous controller and those of the system with the discrete controller never differ in more than 1.8856 (in both variables, x and z). In this case, due to the particular choice of the controller and quantization intervals h, g, U, the plant effectively reaches the origin. More generally, however, the QSC implementation will yield final oscillations (bounded according to Theorem 1) requiring a nonzero average data-rate. The quantification of bounds of such data-rate is subject of current work.

IV. DIRECT DEADBEAT CONTROL DESIGN

We now present a different stabilization strategy based on LCS by hysteretic quantization that drives the state of

4426

the system (1) to the origin in finite time from an unknown initial condition $x(0) = x_0$. In other words, we will show that with an asynchronous LCS scheme, it is possible to achieve stability after transmitting a finite number of bits.

Consider the feedback scheme of Figure 1, as described in Section II, and we assume that the plant (1) is unstable.

The proposed stabilization strategy consists in two open-loop procedures performed sequentially: 1) Finite-time initial state estimation 2) Finite-time control to the origin.

A. Finite-Time Initial State Estimation

Set the control input $u(t) = 0$. The controller input receives one bit per sample indicating whether the system output has increased (a 1 received) or decreased (a 0 received) in a value h with respect to the previous sample. The sampling times $\{t_0, t_1, \dots\}$ are also recorded. The sampling times coincide with the samples *arrival* times, since there is no transmission delay.

Let $x(0) = x_0$ be the plant initial state and let $\{t_0, t_1, \dots, t_n\}$, be the sequence of sampling times corresponding to the first $n+1$ samples, where n, recall, is the order of the plant (1). Since $u(t) = 0$, we have

$$y(t_0) = Ce^{At_0}x_0, \quad y(t_1) = Ce^{At_1}x_0, \quad \cdots \quad y(t_n) = Ce^{At_n}x_0.$$

Since $y(0)$ is unknown, from (6) we write

$$Y_r \triangleq \begin{bmatrix} y_r(t_1) \\ y_r(t_2) \\ \vdots \\ y_r(t_n) \end{bmatrix} = \begin{bmatrix} C(e^{At_1} - e^{At_0}) \\ C(e^{At_2} - e^{At_1}) \\ \vdots \\ C(e^{At_n} - e^{At_{n-1}}) \end{bmatrix} x_0 \triangleq E x_0. \quad (21)$$

Then, since the sampling instants $\{t_0, t_1, \dots, t_n\}$ are available, and the vector Y_r is known independently of the value $y(0)$ (due to the property (6)), we can obtain $x_0 = E^{-1}Y_r$ after receiving $n+1$ bits, provided E is nonsingular.

B. Non Pathological Sampling

The nonsingularity of the matrix E can be seen as analogous to the well-known non pathological sampling condition in periodic sampled-data systems [17, p. 40–42].

Theorem 2: Assume that the matrix A is unstable and it has only nonzero real eigenvalues. Then, provided that the pair (A,C) is observable, the matrix E defined in Equation (21) is non singular.

Proof: Suppose that the matrix E is singular. Then, the equation $E x_0 = 0$ has non trivial solutions. Let $\tilde{x}_0 \neq 0$ be one of these solutions and define $\tilde{y}(t) \triangleq Ce^{At}\tilde{x}_0$. It follows that $\tilde{y}(t_k) - \tilde{y}(t_{k-1}) = E_k\tilde{x}_0 = 0$ for $k = 1, 2, \dots, n$, where $E_k \triangleq C(e^{At_k} - e^{At_{k-1}})$. Then, $\tilde{y}(t_0) = \tilde{y}(t_1) = \cdots = \tilde{y}(t_{n+1})$, which implies that the system output crosses $n+1$ times the value $\tilde{y}(t_0)$. This, however, is impossible for a system of order n in absence of null or complex eigenvalues. ∎

If the plant had complex or null eigenvalues, the matrix E might be singular. However, the pathological sampling condition can only occur if the sampling instants $t_1, \dots, t_{n+1}$ (the instants of crossing of the output $y(t)$ by levels separated by $\pm h$) coincide with the instants in which $\tilde{y}$ (the output from

a different initial condition) crosses a unique level. Such situation is indeed possible if the plant has null eigenvalues, because the output might then remain constant from some initial condition. For complex eigenvalues, we conjecture that pathological sampling can be made impossible by choosing h sufficiently small with respect to $y(t_0)$. In that case, the sampling instants $t_0, t_1, \cdots$ will turn out to be such that the elapsed time $t_{n+1} - t_0$ results shorter than the shortest period of any natural oscillation mode in the plant.

C. Finite-time Control to the Origin

Now, at any time $\tau_0 > t_n$ we know $x(0)$ exactly, and thus can compute $x(\tau_0) = e^{A\tau_0}x_0$, and from $x(\tau_0)$ we can drive the system to the origin in finite time by an open loop bang-bang strategy in a maximum of $n+1$ switches of the input $u(t)$, which for simplicity we can assume to take values in the set $\{-U, 0, U\}$, for some sufficiently large $U > 0$.

The switching times $\{\tau_1, \tau_2, \dots, \tau_{n-1}\}$ and the time of arrival to the origin τ_n can be computed by different methods by solving a minimum time control problem [18].

Thus, stability with the proposed LCS scheme is achieved in *finite-time*; namely, with the transmission of $2n+2$ bits.

D. Remarks on Stabilization under Data-Rate Constraints

The above two-cycle LCS stabilization strategy appears to, at least in some simple cases, circumvent limitations in the problem of feedback stabilization under data-rate constraints. If the plant is stabilized by digital feedback control by periodically sampling of $y(t)$ at a rate $1/T$ Hz and a synchronous zero-order-hold is used to generate the control $u(t)$, then it is known the plant (1) can be stabilized if and only if the average data-rate satisfies [19], [3]

$$\frac{R}{T} > (\log_2 e) \sum_{i=1}^{m} \text{Re}\, p_i > 0 \quad \text{bits per second.}$$

On the other hand, we have seen above that the LCS strategy achieves feedback stability after the transmission of $2n+2$ bits; namely, with *zero* average data-rate.

This may appear as an intriguing property of the proposed asynchronous LCS digital control strategy. However, it should be noted that we have relied on: 1) instantaneous availability of information about the sampling instants $\{t_0, t_1, \dots, t_n\}$ 2) knowledge of the sampling instants $\{t_0, t_1, \dots, t_n\}$ to infinite precision. In practice, there will be delays in transmission, which, if unknown, will prevent stability with the proposed ideal strategy. Also, note that although the information about the measured output is quantized, it is exact. Yet, approximation will be necessary in recording the sampling times $\{t_0, t_1, \dots\}$ digitally, which would then need to be quantized to finite precision.

Effectively, this stabilization strategy makes use of an inherent coding of the output amplitudes into the sequence of sampling instants. If these sampling instants were not exactly known (due to unknown delays or quantization), the plant will not in general reach the origin in finite time at the end of the control cycle. Since the plant is

assumed to be unstable, the output will then start crossing new quantization levels, and a new set of estimation/control cycles will have to be repeated. Following this idea, we would then necessarily arrive at a nonzero average data-rate to achieve closed-loop stability.

Let us now consider an alternative time-driven sampling stabilization approach. In the ideal situation we assumed that the channel can transmit one bit at any instant of time. Thus, it could be alternatively argued to periodically sample the plant output and send two bits, together with an interval of time proportional to that output value, in a pulse-width modulated fashion. Namely, we could then code (and also decode) the plant output with infinite accuracy. Thus, after receiving n samples (corresponding in this case to $2n$ bits) it would be possible to estimate the plant initial state and apply a control strategy to drive the state to the origin, again achieving stability with a finite number of transmitted bits.

In terms of transmitted information, the above reasoning shows that by allowing the channel to transmit at any instant of time is equivalent to the transmission of an infinite amount of information per sample. Under these conditions, a time-driven sampling strategy could also achieve stability in a finite number of transmissions.

It would thus seem that the only advantage of the proposed event-driven sampling scheme is that it uses simpler devices than those required by the above pulse-width modulation coding strategy.

However, in a non-ideal case (that is, in the presence of unknown delays, noise, etc), such time-driven sampling scheme must continue to sample using a *worst-case* sampling period, so that the output does not escape too far away from the origin.

On the other hand, the proposed event-driven sampling scheme will generate samples (and act on the actuator communication link) only when the plant output reaches the value $\pm h$. Thus, if the estimation of the initial state was sufficiently good—for example, because during such period we did not have large perturbations—the plant will reach a state very close to the origin and it will take a relatively long time until the output reaches $\pm h$.

V. CONCLUSIONS

We have presented a digital control formulation based on an asynchronous level-crossing sampling scheme by hysteretic quantization. Such formulation appears attractive in control scenarios in which feedback is implemented over data-rate limited channels, which we have illustrated by presenting two approaches for control design: an asynchronous digital emulation design, and a direct digital design strategy.

In asynchronous emulation, a previously designed analog controller is implemented as a quantized state controller, which is shown to guarantee practical stability. In some cases—as in the example introduced—it also achieves stability after a finite number of transmissions.

The direct digital design strategy consists in an open loop deadbeat estimation cycle, followed by a bang-bang control cycle to drive the plant state to the origin. Stability is achieved with a finite number of transmissions.

Measurement noise and model uncertainty will in general prevent finite-time stability with these schemes. Nevertheless, it would be of interest to quantify and compare their overall performance with that of approaches based on conventional periodic time-driven sampling and quantized measurements. This will be the topic of future work.

REFERENCES

[1] G.N. Nair, F. Fagnani, S. Zampieri, and R.J. Evans, "Feedback control under data rate contraints: an overview", *IEEE Proceedings. Special Issue on Technology of Networked Control Systems*, December 2006, To appear.

[2] J. Baillieul, "Feedback designs in information-based control", in *Stochastic Theory and Control: Proceedings of a Workshop held in Lawrence, Kansas*, 2002, vol. 280 of *Lecture Notes in Control and Information Sciences*, pp. 35–58.

[3] J. H. Braslavsky, R. H. Middleton, and J. S. Freudenberg, "Feedback stabilisation over signal-to-noise ratio constrained channels", in *Proc. 2004 American Control Conference*, Boston, USA, 2004, vol. 6, pp. 4903–4908.

[4] W.S. Wong and R.W. Brockett, "Systems with finite communication bandwidth constraints. ii. stabilization with limited information feedback", *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1049–1053, May 1999.

[5] K. Li and J. Baillieul, "Robust quantization for digital finite communication bandwidth (dfcb) control", *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1573–1584, 2004.

[6] K.J. Åström and B.M. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems", in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002, pp. 2011–2016.

[7] M. Miśkowicz, "The event-triggered sampling optimization criterion for distributed networked monitoring and control systems", in *Proc. of the IEEE International Conference on Industrial Technology*, 2003, vol. 2, pp. 1083–1088.

[8] R. McCann, A.K. Gunda, and S.R. Damugatla, "Improved operation of networked control systems using Lebesgue sampling", in *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting*, 2004, vol. 2, pp. 1211–1216.

[9] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "Asynchronous level crossing analog to digital converters", *Measurement*, vol. 37, pp. 296–309, 2005.

[10] K.J. Åström, "Event based sampling", Presented at the Pravin Varaiya Symposium on Systems, Control and Networks, June 2005.

[11] E. Kofman, "Quantized state control: a method for discrete event control of continuous systems", *Latin American Applied Research*, vol. 33, no. 4, pp. 399–406, 2003.

[12] E. Kofman, "Discrete Event Simulation of Hybrid Systems", *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1771–1797, 2004.

[13] T. Kailath, *Linear Systems*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[14] B. Zeigler, T.G. Kim, and H. Praehofer, *Theory of Modeling and Simulation. Second edition*, Academic Press, New York, 2000.

[15] E. Kofman, "Non conservative ultimate bound estimation in LTI perturbed systems", *Automatica*, vol. 41, no. 10, pp. 1835–1838, 2005.

[16] Ernesto Kofman, Hernan Haimovich, and Maria Seron, "A systematic method to obtain ultimate bounds for perturbed systems", *International Journal of Control*, 2006, in press.

[17] T. Chen and B.A. Francis, *Optimal sampled-data control systems*, Springer-Verlag, 1995.

[18] F. Grognard and R. Sepulchre, "Computation of time-optimal switchings for linear systems with complex poles", in *Proceedings of the European Control Conference*, Cambridge, United Kingdom, 2003, CDROM.

[19] G.N. Nair and R.J. Evans, "Exponential stabilisability of finite-dimensional linear systems with limited data rates", *Automatica*, vol. 39, no. 4, pp. 585–593, April 2003.