This may be the author’s version of a work that was submitted/accepted for publication in the following source:

Quevedo, D.E., Gupta, Vijay, Ma, W.-J., & Yuksel, S. (2014)
Stochastic stability of event-triggered anytime control.
*IEEE Transactions on Automatic Control, 59*(12), pp. 3373-3379.

This file was downloaded from: https://eprints.qut.edu.au/200470/

© IEEE

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

**Notice:** Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.

https://doi.org/10.1109/TAC.2014.2351952
Stochastic Stability of Event-triggered Anytime Control
Daniel E. Quevedo, Member, IEEE, Vijay Gupta, Member, IEEE, Wann-Juin Ma, Student Member, IEEE, Serdar Yüksel, Member, IEEE

Abstract—We investigate control of a non-linear process when communication and processing capabilities are limited. The sensor communicates with the controller node through an erasure channel which introduces i.i.d. packet dropouts. Processor availability for control is random and, at times, insufficient to calculate plant inputs. To make efficient use of communication and processing resources, the sensor only transmits when the plant state lies outside a bounded target set. Control calculations are triggered by the received data. If a plant state measurement is successfully received and while the processor is available for control, the algorithm recursively calculates a sequence of tentative plant inputs, which are stored in a buffer for potential future use. This safeguards for time-steps when the processor is unavailable for control. We derive sufficient conditions on system parameters for stochastic stability of the closed loop and illustrate performance gains through numerical studies.

I. INTRODUCTION

Due to the tight coupling among the cyber and the physical cores, in many cyber-physical systems it is imperative to develop systematic design principles for controllers with limited communication and processing resources. Both the areas of control with communication constraints and control with limited and time-varying processor availability have accordingly received much attention.

Control design in the presence of practical communication channels and protocols has been studied in the area of networked control systems. Of particular interest to the present work is the literature on control across analog erasure channels; see, e.g., [1]–[4]. Apart from arising from data transmission across a wireless channel, data loss may also arise due to congestion in a communication network, possibly transmitted by a control loop. To minimize the latter source of data loss, one can conceive event triggered communication schemes in which sensors transmit information only if the system state exceeds a certain bound; see, e.g., [5]–[9]. Recently, work has also been done on designing event triggering rules to ensure stability in the face of data dropouts. However, most works are restricted to single integrator dynamics [10], [11] or are numerical studies [12].

On the other hand, various works have also considered the impact of limited or time-varying processing power on closed-loop control [13]–[15]. Interestingly, event-triggered and self-triggered updates of the control inputs have also been proposed to ensure less demand on the processor on average, by calculating the control input on demand [6], [16]. The direction of anytime control has also shown promise [17]–[20]. Such algorithms calculate a coarse control input even with limited processing resources and refine the input as more processing resources become available. The quality of control inputs is thus time-varying, but no control input is obtained only rarely.

Notwithstanding the advances discussed above, relatively few works have considered control design under both limited communication and processing resources. Optimal control design for arbitrary non-linear processes under communication and processing constraints is likely a challenging problem, since certainty equivalence would not hold in general [21]. Accordingly, in the present note we consider a pre-designed control law, and focus on the implementation of this controller in the presence of both communication and processing limitations. As depicted in Fig. 1, we consider a discrete-time nonlinear plant being controlled across a communication network that stochastically erases data transmitted across it. To reduce congestion in the network, the sensor employs an event triggered communication strategy. However, due to time-varying availability of the processing resources, it is not guaranteed that the processor can calculate a control input at all time steps when the sensor transmits (even if the network does not erase the data). To maximally utilize the processing resources, the controller employs an anytime control algorithm. Under such a setting, we analyze stochastic stability of the closed loop. Our main stability results are stated in terms of an inequality that relates open-loop growth of the plant state, packet erasure probability, and parameters of the processor availability model. For the particular case where processing resources are available at every time step, our result reduces to a sufficient condition for stochastic stability of non-linear control in where the sensor communicates according to an event-triggering condition across an analog i.i.d. erasure link. This may be of independent interest.

Notation: We write \( \mathbb{N} \) for \{1, 2, \ldots\} and \( \mathbb{N}_0 \) for \( \mathbb{N} \cup \{0\} \). \( \mathbb{R} \) represents the real numbers and \( \mathbb{R}_{\geq 0} \triangleq [0, \infty) \). The \( p \times p \) identity matrix is denoted via \( I_p \), \( 0_{p \times q} \) is the \( p \times q \) all-zeros matrix, \( 0_p \triangleq 0_{p \times p} \), and \( 0_p \triangleq 0_{p \times 1} \). The notation \( \{x\}_K \) stands for \( \{x(k) \mid k \in K\} \), where \( K \subseteq \mathbb{N}_0 \). We adopt the conventions \( \sum_{k=1}^{\ell_2} a_k = 0 \) and \( \prod_{k=1}^{\ell_1} a_k = 1 \), if \( \ell_1 > \ell_2 \) and irrespective of \( a_k \in \mathbb{R} \). The superscript \( T \) refers to transpose. The Euclidean norm of a vector \( x \) is denoted via \( \|x\| = \sqrt{x^T x} \). A function \( \varphi: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \) is of class \( \mathcal{K}_\infty \) ( \( \varphi \in \mathcal{K}_\infty \)), if it is continuous, zero at zero, strictly increasing, and unbounded. The probability of an event \( \Omega \) is denoted by \( \Pr\{\Omega\} \) and the conditional probability of \( \Omega \) given \( \Gamma \) by \( \Pr\{\Omega \mid \Gamma\} \). The expected value of a random variable \( x \) given \( \Gamma \) is denoted by \( \mathbb{E}\{x \mid \Gamma\} \), while \( \mathbb{E}\{x\} \) refers to the unconditional expectation. The expression \( x \sim \nu \) denotes that the random variable \( x \) has probability distribution \( \nu \) and \( \mathbb{E}_\nu \{x\} \) denotes the expectation under probability distribution \( \nu \).
II. EVENT-DRIVEN CONTROL OVER AN ERASURE CHANNEL

We consider non-linear (and possibly open-loop unstable) plants, sampled periodically with sampling interval \( T_s > 0 \) (see Fig. 1),

\[
x(k + 1) = f(x(k), u(k)), \quad k \in \mathbb{N}_0,
\]

where \( x \in \mathbb{R}^n \) is the plant state, and \( u \in \mathbb{U} \subseteq \mathbb{R}^p \) with \( \mathbf{0}_p \in \mathbb{U} \) is the (possibly constrained) plant input. The initial state \( x(0) \) is arbitrarily distributed. The plant is equipped with a sensor, which has direct access to the plant state at the sampling instants \( k \in \mathbb{N}_0 \).

To save on communication expenditure, the sensor adopts an event-triggered transmission strategy, in which the sensor transmits only at instances \( k \in \mathbb{N}_0 \), where

\[
x(k) \not\in \mathcal{B}_d \triangleq \{ x \in \mathbb{R}^n : |x| < d \}.
\]

This transmission is across an erasure channel which introduces random packet dropouts. To keep communication costs low, the controller does not send acknowledgments back to the sensor and no re-transmissions are allowed. We introduce two discrete random processes, namely \( \{ \gamma \}_{k=0}^{\infty} \) and \( \{ \beta \}_{k=0}^{\infty} \). The binary transmission success process \( \{ \gamma \}_{k=0}^{\infty} \) describes packet loss: a successful transmission at time \( k \) is denoted by \( \gamma(k) = 1 \) and a packet erasure by \( \gamma(k) = 0 \). The ternary process \( \{ \beta \}_{k=0}^{\infty} \) incorporates the event-based transmission rule,

\[
\beta(k) = \begin{cases} 
\gamma(k) & \text{if the sensor transmitted at time } k, \\
2 & \text{if the sensor did not transmit at time } k.
\end{cases}
\]

Thus, \( \beta(k) = 2 \Leftrightarrow |x(k)| < d \). We assume that \( \beta(k) \) is known to the controller at time \( k \) through error-detection coding and monitoring of received energy in the sensor transmission band. Transmission outcomes trigger the functions carried out by the controller. The scalar \( d \in \mathbb{R}_{\geq 0} \) is a design parameter, which determines communication channel utilization and control performance. Elucidating the trade-off between these quantities is one of the motivations of the present work.

When implementing discrete-time control systems, it is generally assumed that the processing resources available to the controller are such that the desired control law can be evaluated within a fixed time-delay, say \( \delta \in (0, T_s) \). However, in practical networked and embedded systems, the processing resources available for control calculations may vary and, at times, be insufficient to generate a control input within the prescribed time-delay \( \delta \) [15]. In the sequel we will further develop our anytime control algorithm of [20, 22] to seek favorable trade-offs between processor and communication availability, and control performance. We will assume that the plant model (1) is globally stabilizable via state feedback.

**Assumption 1 (Stabilizability):** There exist \( V : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}, \varphi_1, \varphi_2 \in \mathcal{K}_{\infty}, \kappa : \mathbb{R}^n \rightarrow \mathbb{U} \), and a constant \( \rho \in [0, 1) \), such that

\[
\varphi_1(|x|) \leq V(x) \leq \varphi_2(|x|), \quad \forall x \in \mathbb{R}^n,
\]

\[
V(f(x, \kappa(x))) \leq \rho V(x), \quad \forall x \not\in \mathcal{B}_d.
\]

(3)

To encompass processing constraints, we will assume that the controller needs the processor to carry out mathematical computations, such as evaluating \( \kappa \). However, input-output operations and simple operations at a bit level, e.g., writing data into buffers, shifting buffer contents and setting values to zero, do not require processor time.

Before proceeding we note that a direct implementation of \( \kappa \) used in Assumption 1, when processing resources are time varying, sensor transmissions are event-triggered, and the sensor transmissions are affected by dropouts, results in the baseline event-based algorithm

\[
u(k) = \begin{cases} 
\kappa(x(k)) & \text{if } \beta(k) = 1 \text{ and processor is available,} \\
\mathbf{0}_p & \text{otherwise,}
\end{cases}
\]

where \( \kappa(x(k)) \) is the current control value, provided the processor is available for control. This sequence will be stored in a buffer. If the processor is not available \( \kappa(x(k)) \) is not sent to the controller, and a new state measurement is received. Fig. 2 outlines the proposed algorithm. In this figure,

\[
\begin{pmatrix}
0_p & I_p & 0_p & \ldots & 0_p \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_p & \ldots & \ldots & 0_p & I_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
\end{pmatrix} \in \mathbb{R}^{Ap \times \Lambda_p}, \quad b(k) =
\begin{pmatrix}
b_1(k) \\
b_2(k) \\
\vdots \\
b_{\Lambda}(k)
\end{pmatrix},
\]

where \( \{ b \}_{k=0}^{\infty} \) denotes the buffer states for a given buffer size \( \Lambda \in \mathbb{N} \) and each \( b_j(k) \in \mathbb{R}^p, j \in \{1, \ldots, \Lambda\} \).

For future use, we will denote by \( N(k) \in \{0, 1, \ldots, \Lambda\} \) the total number of iterations of the while-loop (see Fig. 2), which are carried out during the interval \( t \in [kT_s, (k+1)T_s) \). Thus, as described above, if \( N(k) \geq 1 \), then the entire sequence of tentative controls is \( \{ b_1(k), b_2(k), \ldots, b_{N(k)}(k) \} \) and the plant input is set to \( b_1(k) \). If \( N(k) = 0 \), then the plant input depends on the variable \( \beta(k) \).

If \( \beta(k) \in \{0, 1\} \) (i.e., \( x(k) \) does not lie inside the desired region), then \( u(k) \) is taken as the first \( p \) elements of the shifted state \( b(k) = \begin{pmatrix}
0_p & I_p & 0_p & \ldots & 0_p \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_p & \ldots & \ldots & 0_p & I_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
\end{pmatrix} \in \mathbb{R}^{Ap \times \Lambda_p}, \quad b(k) =
\begin{pmatrix}
b_1(k) \\
b_2(k) \\
\vdots \\
b_{\Lambda}(k)
\end{pmatrix},
\]

where \( \{ b \}_{k=0}^{\infty} \) denotes the buffer states for a given buffer size \( \Lambda \in \mathbb{N} \) and each \( b_j(k) \in \mathbb{R}^p, j \in \{1, \ldots, \Lambda\} \).

For future use, we will denote by \( N(k) \in \{0, 1, \ldots, \Lambda\} \) the total number of iterations of the while-loop (see Fig. 2), which are carried out during the interval \( t \in [kT_s, (k+1)T_s) \). Thus, as described above, if \( N(k) \geq 1 \), then the entire sequence of tentative controls is \( \{ b_1(k), b_2(k), \ldots, b_{N(k)}(k) \} \) and the plant input is set to \( b_1(k) \). If \( N(k) = 0 \), then the plant input depends on the variable \( \beta(k) \).

If \( \beta(k) \in \{0, 1\} \) (i.e., \( x(k) \) does not lie inside the desired region), then \( u(k) \) is taken as the first \( p \) elements of the shifted state \( b(k) = \begin{pmatrix}
0_p & I_p & 0_p & \ldots & 0_p \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0_p & \ldots & \ldots & 0_p & I_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
0_p & \ldots & \ldots & 0_p & 0_p \\
\end{pmatrix} \in \mathbb{R}^{Ap \times \Lambda_p}, \quad b(k) =
\begin{pmatrix}
b_1(k) \\
b_2(k) \\
\vdots \\
b_{\Lambda}(k)
\end{pmatrix},
\]
Algorithm $A_k$ amounts to a dynamic state feedback policy with internal state variable $b(k)$ which provides the plant input $u(k)$ and suggested plant inputs at future time steps. If new state information is received and more processor time is available, a longer trajectory of control inputs is calculated and stored in the buffer. If the buffer runs out of tentative plant inputs, then actuator values are set to zero.

The algorithm does not require prior knowledge of future processor availability and hence can be employed in shared systems where the controller task can be preempted by other computational tasks.

IV. STOCHASTIC STABILITY: PRELIMINARIES

For our subsequent analysis, it is convenient to investigate how many values in the state $b(k)$ stem from having evaluated $\kappa$. As in [20], [22], we will refer to this value as the effective buffer length (at time $k$), and denote it as $\lambda(k) \in \{0, 1, \ldots, \Lambda\}, \Lambda \in \mathbb{N}$ with $\lambda(-1) = 0$. It is easy to see that for all $k \in \mathbb{N}$ we have

$$\lambda(k) = \begin{cases} N(k) & \text{if } N(k) \geq 1, \\ \max\{0, \lambda(k-1) - 1\}, & \text{if } N(k) = 0 \land \beta(k) \in \{0, 1\}, \\ 0 & \beta(k) = 2. \end{cases}$$

To investigate stability, we make the following assumptions:

Assumption 2 (Processor availability): The sampling interval $T_s$ is such that processor availability for control at different intervals is independent and identically distributed (i.i.d.). Thus, the process $\{N\}_{n=0}^\infty$ has conditional probability distribution

$$P_j := P\{N(j) = j | \beta(k) = 1\}, \quad j \in \{0, 1, 2, \ldots, \Lambda\}$$

where $p_j \in [0, 1]$ are given. For other realizations of $\beta(k)$, no plant inputs are calculated, thus, $P\{N(k) = 0 | \beta(k) \in \{0, 2\}\} = 1$.

Assumption 3 (Erasure channel): The binary transmission success process $\{\gamma\}_{n=0}^\infty$ has conditional probabilities

$$P_{\gamma(j) = 1} = \{\gamma(\chi) \geq d\} = q, \quad q \in (0, 1]$$

whereas $P_{\gamma(\chi) = 0} = \{\gamma(\chi) < d\} = 1$.

Assumption 4 (Open-loop bound): There exists $\alpha \geq \rho$ such that

$$V(f(x, 0)) \leq \alpha V(\chi), \quad \forall \chi \in \mathbb{R}^n, \quad (5)$$

where $\rho$ and $V$ are as in (3). Furthermore, $E\{\varphi_2(|x(0)|)\} < \infty$. It is worth noting that, by allowing for $\alpha > 1$, Assumption 4 does not require that the open-loop system $x(k+1) = f(x(k), 0)$ be asymptotically stable. Further, note that Assumptions 1 and 4 are stated in terms of the same function $V$, see also [20, Section IV-A].

To go beyond stability and investigate stationarity, it is convenient to impose the following assumptions on the control policy $\kappa$:

Assumption 5 (Continuity of $\kappa$): The control law $\kappa$ in (3) is such that $\kappa(x) = \nu_0$ for all $x \in \mathbb{B}_\delta$ and $\kappa$ is continuous on $\mathbb{R}^n$. \hfill $\square$

V. STABILITY WITH THE BASELINE ALGORITHM

If the baseline algorithm (4) is used and Assumption 2 holds, then

$$x(k+1) = \begin{cases} f(x(k), \kappa(x(k))), & \text{if } N(k) \geq 1, \\ f(x(k), 0), & \text{if } N(k) = 0. \end{cases} \quad (6)$$

The following result establishes conditions on system parameters which ensure that the closed loop (6) is stable in a stochastic sense.

Theorem 1 (Stability with baseline algorithm): Consider (6) and define $D \triangleq \varphi_2(d)$. Suppose that Assumptions 1 to 4 hold and that

$$\Gamma \triangleq (1 - \rho)\alpha + q(\rho \alpha + (1 - \rho)\rho) < 1, \quad (7)$$

where $\rho \in [0, 1]$ is the closed-loop bound in (3), $\alpha$ is the bound in (5), $q$ is the transmission success probability, and $p_0$ is the probability of the processor not being available for control. Then, for all $k \in \mathbb{N}_0$,

$$E\{\varphi_1(|x(k)|)\} \leq \Gamma^k E\{\varphi_2(x(0))\} + \frac{q(1 - p_0)(\alpha - \rho)D}{1 - \Gamma} < \infty.$$ 

Proof: Note that for i.i.d. processor and channel availabilities, $\{x\}_{n=0}^\infty$ in (6) is Markovian. This can be verified by noting that conditioning on $x(k)$ makes the event outcome $\beta(k)$ depend on $\gamma(k)$ only. We next analyze stochastic stability using Lyapunov functions (see, e.g., [24]) and use the law of total expectation to write

$$E\{V(x(1)) \mid x(0) = \chi, \beta(0) = j\} \times P_{\beta(0) = j \mid x(0) = \chi}, \quad \forall \chi \in \mathbb{R}^n. \quad (8)$$

If we now use (2), (3), (5) and the definition of $\mathbb{B}_\delta$, then:

$$E\{V(x(1)) \mid x(0) = \chi, \beta(0) = 0\} \leq \alpha V(\chi)$$

$$E\{V(x(1)) \mid x(0) = \chi, \beta(0) = 2\} \leq \alpha V(\chi) < \alpha D. \quad (9)$$

For $\beta(0) = 1$, $x(0)$ is received. Using (5) and (6), we have

$$E\{V(x(1)) \mid x(0) = \chi, \beta(0) = 1\} = \sum_{j \in \mathbb{N}_0} E\{V(x(1)) \mid x(0) = \chi, \beta(0) = 1, N(0) = j\}$$

$$\times P_{\beta(0) = 1 \mid x(0) = \chi} = \beta(0) = 1 \leq \left[p_0 + (1 - p_0)\rho\right] V(\chi). \quad (10)$$

Now, if $x(0) \not\in \mathbb{B}_\delta$, then $\beta(0) = 2$, thus (8) and (9) provide

$$E\{V(x(1)) \mid x(0) = \chi \in \mathbb{B}_\delta\} \leq \alpha V(\chi). \quad (11)$$

Further, since $\alpha - \Gamma = q(1 - p_0)(\alpha - \rho) > 0$ (see (7)) and $V(\chi) < \alpha D$ for all $\chi \in \mathbb{B}_\delta$, we have (see, e.g., [24])

$$(\alpha - \Gamma) V(\chi) < (\alpha - \Gamma) D \Rightarrow \alpha V(\chi) < \Gamma V(\chi) + (\alpha - \Gamma) D. \quad (12)$$

On the other hand, if $x(0) \in \mathbb{B}_\delta$, then (in view of Assumption 3),

$$P_{\beta(0) = 0 \mid x(0) \in \mathbb{B}_\delta} = 1 - q,$$ and $P_{\beta(0) = 1 \mid x(0) \not\in \mathbb{B}_\delta} = q$. Thereby, substitution of (9) and (10) into (8) provides

$$E\{V(x(1)) \mid x(0) = \chi \not\in \mathbb{B}_\delta\} \leq \Gamma V(\chi). \quad (13)$$

Expressions (11)–(13) lead to:

$$E\{V(x(1)) \mid x(0) = \chi \} < \Gamma V(\chi) + (\alpha - \Gamma) D = \Gamma V(\chi) + q(1 - p_0)(\alpha - \rho) D.$$ Consequently, Proposition 3.2 of [24], and (3) give

$$E\{\varphi_1(|x(k)|) \mid x(0) = \chi\} \leq \Gamma^k V(\chi) + \frac{q(1 - p_0)(\alpha - \rho) D}{1 - \Gamma},$$

for all $k \in \mathbb{N}_0$. Using the law of total expectation and (3) yields the first inequality; the second follows from Assumption 4.

It is worth noting that whilst the condition (7) is independent of the size of $\mathbb{B}_\delta$, the ultimate bound is increasing in $d$. We can also consider two special cases. If $d = 0$ and $q = 1$, so that the sensor transmits at every instant $k \in \mathbb{N}_0$ and the communication channel does not introduce any dropouts, (7) reduces to $p_0 \alpha + (1 - p_0)\rho < 1$, thereby recovering our earlier result [20, Thm.1]. If the processor is available at every time-step (i.e., $p_0 = 0$), then the situation amounts to event-based control for non-linear systems using an erasure channel. In this case, the sufficient condition (7) becomes $(1 - q)\alpha + pq < 1$.

Theorem 2 (Stationarity with baseline algorithm): Consider (6), suppose that Assumptions 1 to 5 hold and that (7) is satisfied. Then there exists an invariant probability measure for $\{x\}_{n=0}^\infty$. Furthermore, under every such invariant probability measure $\pi$,

$$E_{\pi}\{\varphi_1(|x|)\} \leq q(1 - p_0)(\alpha - \rho)\varphi_2(d)/(1 - \Gamma).$$
Proof: Let $P(R^n)$ denote the set of probability measures on $R^n$ and define for every Borel $B$, $v_T(B) = (1/T)E\{\sum_{k=0}^{T-1} 1_{x(k)\in B}\}$, such that $v_T \in P(R^n)$ forms an expected empirical occupation measure sequence. We then have,

$$\langle v_T, \varphi \rangle \triangleq \int v_T(dx)\varphi_1(|x|) = \frac{1}{T}E\left\{\sum_{k=0}^{T-1} \varphi_1(x(k))\right\}.$$ 

Let $t_0 \in N$. By Theorem 1, we have that $E\{\varphi_1(|x|)\}$ and the subsequence $\langle v_{t_0}, v_{t_0+1}, \ldots, v_{t_0+k}, \ldots \rangle$ are uniformly bounded by some $M_{t_0} < \infty$. Define $N_r := \{x: \varphi_1(|x|) \leq r\}$. Since $\varphi_1$ is monotone and unbounded, by an application of Markov’s inequality, we have $M_{t_0} \geq \int v_{t_0}(dx)\varphi_1(|x|) \geq \int_{x:N_r} v_{t_0}(dx)\varphi_1(|x|) \geq v_T(R^n)\setminus N_r$.

Thus, $v_{t_0}(N_r) \geq 1 - M_{t_0}/r$, and hence for every $\epsilon = M_{t_0}/r > 0$, there exists a compact set $N_{M_{t_0}/\epsilon} := \{x: \varphi_1(|x|) \leq M_{t_0}/\epsilon\}$ such that $v_{t_0}(N_{M_{t_0}/\epsilon}) \geq 1 - \epsilon$. The sequence $\{v_{t_0}, t \geq t_0\}$ is therefore a tight sequence with a converging subsequence $v_{t_k}$, converging to some $v^* \in P(R^n)$. By (6), if $x(t) \in B_d$ the control action is zero and outside $B_d$, either zero control is applied or $\kappa(x(t))$ is applied. Since $\kappa$ is continuous and is zero inside $B_d$ (see Assumption 5), the Markov chain is weak Feller. Consequently, it can be shown that every limit of such a subsequence is invariant (see, e.g., [25, Ch. 12]) and satisfies $\langle v_T, \varphi \rangle \leq M_{t_0}$. By Theorem 1, by increasing $t_0$, the bound $M_{t_0}$ can be taken to be arbitrarily close to $g(1 - p_0)(\alpha - \rho)\varphi_2(d)/(1 - \Gamma)$. This proves the result.

VI. STABILITY WITH THE ANYTIME ALGORITHM

The analysis of the event-based anytime algorithm is more involved than that of the baseline algorithm (6). First, due to buffering, $\{x\}_{t_0}$ will in general not be a Markov process. Further, the distribution of $\{\beta\}_{t_0}$ is difficult to derive for general plant models (1). This makes the approaches of [20, [22] insufficient to treat the present case.

For ease of exposition, we assume that the initial effective buffer length, $\lambda(0) = 0$, and denote the step times where $\lambda(k) = 0$ via $K = \{k_i\}_{i \leq N_0}$, where $k_0 = 0$ and $k_{i+1} = \inf\{k \in N: k > k_i, \lambda(k) = 0\}$, $i \in N_0$. We also describe the amount of time steps between consecutive elements of $K$ via the process $\{\Delta_i\}_{i \leq N_0}$, where $\Delta_i \triangleq k_{i+1} - k_i$. It is easy to see that $\beta(k_i + \ell) \in (0, 1], \forall \ell \in \{1, 2, \ldots, \Delta_i - 1\}, \forall i \leq N_0$ (14) whereas $\beta(k) \in (0, 1, 2], \forall i \leq N_0$ and $x(k^*) \in B_d \Rightarrow k^* \in K$. In contrast to the cases examined in [20, [22], due to the event-triggering mechanism, $\{\Delta_i\}_{i \leq N_0}$ is, in general, not i.i.d. In fact, the distribution of $\Delta_i$ depends on $x$ and is difficult to characterize. To study stability of the event-based anytime algorithm, we will next develop a state-dependent random-time drift condition. Our first result, states that whilst $\{x\}_{t_0}$ is in general not Markovian, the state sequence at the step times $k_i \in K$, is a Markov process.

Lemma 1 (Markov property of the sampled process): Consider (1) controlled via Algorithm A1 and suppose that Assumptions 2 and 3 hold. Then $\{x\}_{t_0}$ is Markovian.

Proof: The definition of $K$ gives that $\forall k_i \in K$, we have $u(k_i) = 0_p, b(k_i) = 0_p, \lambda(k_i) = N(k_i) = 0$. Thus, the plant state at time $k_{i+1}$ depends only on $x(k_i)$ and the sample paths $\{N(k_i+1), N(k_i+2), \ldots, N(k_{i+1} - 1)\}$ and $\{\gamma(k_i+1), \gamma(k_i+2), \ldots, \gamma(k_{i+1} - 1)\}$. The result follows by Assumptions 3 and 4.

The following theorem provides a sufficient condition for stochastic stability of the closed loop when the event-based anytime control algorithm of Section III is used over an erasure channel.

$^1$A Markov chain $\{x(k)\}_{k \in N}$ is (weak) Feller if $E\{h(|x(k+1)|)|x(k) = \chi\}$ is continuous in $\chi$, for every continuous and bounded function $h$.

Theorem 3 (Stability with Algorithm A1): Suppose that Assumptions 1 to 4 hold and define

$$\Omega \triangleq \frac{\alpha}{\frac{1}{\rho} + \rho} \Pr\{\Delta_i = j | \beta(k_{i+1}) \neq 2\}.$$ 

If Algorithm A1 is used and $\Omega < 1$, then

$$\label{eq:16} \Pr\{\Delta_i = j | \beta(k_{i+1}) \neq 2\} \leq \frac{1 + \alpha - \rho}{1 - \rho} \Omega \frac{\varphi_2(d)}{1 - \Omega} < \infty, \forall i \in N.$$ 

Proof: We first note that for all $k_i \in K$ and $i \in \{1, \ldots, \Delta_i - 1\}$, $u(k_i) = 0_p$, and $u(k_i + \ell) = \kappa(x(k_i + \ell))$. Therefore, the function $V(x(k_i + 1))$ can be bounded by using (3) and (5), leading to

$$\Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \Delta_i = j \leq \alpha \rho^{j-1} V(\chi), \forall \chi \in R^n. (17)$$

To account for event-based transmissions, we consider instances where the buffer is emptied triggered by $\beta(k) = 2$. At these instances, (17) holds; further, $V(k_i + 1) < D \triangleq \varphi_2(d)$. We thus have

$$\Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \Delta_i = j, \beta(k_{i+1}) = 2 < D \leq x \in N.$$ 

By using the law of total expectation twice, we consequently obtain,

$$\Pr\{V(x(k_i + 1)) | x(k_i) = \chi\} = \Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \beta(k_{i+1}) = 2\} \times \Pr\{\beta(k_{i+1}) = 2 | x(k_i) = \chi\} + \Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \beta(k_{i+1}) \neq 2\} \times \Pr\{\beta(k_{i+1}) \neq 2 | x(k_i) = \chi\} \leq D + \Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \beta(k_{i+1}) = 2\} \leq D + \Pr\{V(x(k_i + 1)) | x(k_i) = \chi, \beta(k_{i+1}) \neq 2\} \leq D + \Omega V(\chi), \forall x \in R^n$$

with $\Omega$ as in (15) and where, to derive the last equality, we have used Assumption 2. Since $\{x\}_{K}$ is Markovian, [24, Prop. 3.2] yields that $\Omega < 1$ is a sufficient condition for exponential boundedness at the instants $k_i \in K$:

$$\Pr\{V(x(k_i)) | x(k_0) = \chi \leq \Omega V(\chi) + \frac{D}{1 - \Omega}, \forall i \in N_0.$$ 

Now, since (17) holds, by a method similar to the one used in the proof of [22, Thm.1], we can establish the (admittedly loose) bound,

$$\Pr\{x(k_{i+1}) | \sum_{k=k_i}^{k_{i+1}-1} V(x(k)) \geq \chi\} \leq \frac{1 + \alpha - \rho}{1 - \rho} \Omega \frac{\Delta_i}{1 - \Omega}, \forall i \leq N_0.$$ 

Using the law total expectation, (3) and Assumption 4 gives (16).

The above result establishes a sufficient condition for the system to be stochastically stable. The quantity (15) is stated in terms of a conditional distribution of $\Delta_i$, which can be characterized as follows:

Lemma 2 (Conditional distribution of $\Delta_i$): Suppose that Assumptions 2 and 3 hold and that Algorithm A1 is used. We then have

$$\Pr\{\Delta_i = j | \beta(k_{i+1}) \neq 2\} = \frac{1}{\theta^j \Gamma^{j-2} e_{1}} \text{ if } j = 1,$$

$$\Pr\{\Delta_i = j | \beta(k_{i+1}) \neq 2\} = \frac{1}{\theta^j \Gamma^{j-2} e_{1}} \text{ if } j \geq 2,$$

for all $(i, j) \in N_0 \times N$, where $\theta^T = q[p_1, \ldots, p_N]$ and $e_1^T = [1, 0, \ldots, 0]$. In (20), the entries of the matrix $G = [g_{ij}]$, 

Preprint submitted to IEEE Transactions on Automatic Control. Received: April 1, 2014 04:38:48 PST
parameter

The channel utilization result illustrates that the channel utilization based on the anytime control

\[ g_{tj} = p_{tj} + p_{t}q, \forall (t, j) \in \{3, 4, \ldots, \Lambda \} \times \{1, 2, \ldots, \ell - 2 \} \cup \{2, 3, \ldots, \Lambda \} \times \{1, \ell + 1, \ldots, \Lambda \}; \]  

and \( g_{t(\ell - 1)} = 1 - q + (p_{0} + p_{\ell - 1})q, \forall \ell \in \{2, 3, \Lambda \} \).

**Proof:** We first note that our focus is on the time sequences of the form \( T_{k} = \{k_{1}, 1, \ldots, k_{n-1}\} \) where \( k_{i} \in \mathbb{K}, i \in \mathbb{N}_{0} \) in which case \( k_{i} \in T_{i} \), we have \( \beta(k) \neq 2 \). Given Assumptions 2 and 3 and the buffering mechanism described in Section III, it follows that \( (\lambda(k)) \) during every interval \( k \in T_{i}, i \in \mathbb{N}_{0} \), is a homogeneous Markov Chain. The process \( \Delta_{i} \) then amounts to the first return times to 0 of this finite Markov Chain. To characterize the latter, we evaluate the transition probabilities \( g_{il} \triangleq \mathbf{Pr}(\lambda(k + 1) = j | \lambda(k) = \ell, k \in T_{i}, k + 1 \in T_{j}) \). Without loss of generality, we will set \( k = 0 \). We begin by considering transitions from \( \ell \in \{0, 1\} \) to 0:

\[ g_{00} = \mathbf{Pr}(N(1) = 0 | \beta(1) = 0) \mathbf{Pr}(\beta(1) = 0 | \beta(1) \neq 2) \]
\[ + \mathbf{Pr}(N(1) = 0 | \beta(1) = 1) \mathbf{Pr}(\beta(1) = 1 | \beta(1) \neq 2) \]
\[ = (1 - q) + p_{0}q. \forall \ell \in \{0, 1\}. \]

For \( \ell \in \{2, 3, \ldots, \Lambda\} \), we have \( g_{t0} = 0 \). The buffer length diminishes by one for the scenarios considered below:

\[ g_{t(t-1)} = \mathbf{Pr}(N(1) = 0 | \beta(1) = 0) \mathbf{Pr}(\beta(1) = 0 | \beta(1) \neq 2) \]
\[ + \mathbf{Pr}(N(1) = 0 | \beta(1) = 1) \mathbf{Pr}(\beta(1) = 1 | \beta(1) \neq 2) \]
\[ = (1 - q) + p_{0}q + p_{\ell - 1}q. \forall \ell \in \{2, 3, \ldots, \Lambda\}. \]

The other transitions are related to when \( \lambda(k + 1) = N(k + 1) \), for \( (\ell, j) \in \{3, 4, \ldots, \Lambda \} \times \{1, 2, \ldots, \ell - 2\} \cup \{1, 2, \ldots, \lambda \} \times \{\ell, \ell + 1, \ldots, \Lambda\} \cup \{0 \times 1, 2, \ldots, \Lambda\} \). Here we have:

\[ g_{ij} = \mathbf{Pr}(\lambda(1) = j | \lambda(0) = \ell, \beta(1) = 0) \mathbf{Pr}(\beta(1) = 0 | \beta(1) \neq 2) \]
\[ + \mathbf{Pr}(\lambda(1) = j | \lambda(0) = \ell, \beta(1) = 1) \mathbf{Pr}(\beta(1) = 1 | \beta(1) \neq 2) \]
\[ = \mathbf{Pr}(N(1) = 0 | \ell, j - 1) \mathbf{Pr}(\beta(1) = 1 | \beta(1) \neq 2) \]
\[ = 0(1 - q) + p_{j}q. \forall \ell \in \{2, 3, \ldots, \Lambda\}. \]

The derivation of (20) now follows as in, e.g., [22, Lemma 2] by setting up a recursion on the first passage time of state \( \ell \in \{1, \ldots, \Lambda\} \) to 0 and then considering the transitions away from state \( \lambda = 0 \).

As a consequence of Lemma 2, \( \Omega \) in (15) can be written as:

\[ \Omega = \alpha(1 - q + p_{q}) (1 + \rho \sum_{r=1}^{\infty} (I - \rho G)^{-1} e_{1} \),

and the stability condition in Theorem 3, \( \Omega < 1 \), becomes

\[ p_{1} \ldots p_{\Lambda} (I - \rho G)^{-1} e_{1} < \frac{1 - \alpha + \alpha q(1 - p_{0})}{\alpha q(1 - q(1 - p_{0}))} \]

which, as in (7), is independent of the size of \( \mathbb{B}_{G} \).

Sufficient conditions for stationarity can be stated as follows:

**Theorem 4 (Stationarity with Algorithm A1):** Suppose that Assumptions 1 to 5 hold. If Algorithm A1 is used and \( \Omega < 1 \), then there exists an invariant probability measure for \( \{x\}_{\mathbb{K}} \) as well as for the aggregated Markov process, \( \{x_{[k,k-(\Lambda-1)]}\} \in \mathbb{K} \), where

\[ x_{[k,k-(\Lambda-1)]} \triangleq \{x(k), x(k - 1), \ldots, x(k - \Lambda + 1)\}. \]

Furthermore, under every invariant probability measure \( \pi \), \( E_{\pi}(V(x(k))) \leq \varphi_{\Omega}(d)/(1 - \Omega) \).

**Proof:** First note that if \( N(k) \geq 1 \), then \( u(k) \) is determined by the current state. If the processor is not available, then either \( u(k) \) has been determined by the states which are at most \( \Lambda \) time stages old, or \( u(k) = \emptyset \). Since the processor availability is independent of the state, the stochastic process \( \{x_{[k,k-(\Lambda-1)]}\} \) is Markovian. Let \( z(k) \triangleq x_{[k,k-(\Lambda-1)]} \). From Assumption 5, \( \{z\}_{\mathbb{H}_{0}} \) is also weak Feller.

...
VII. NUMERICAL EXAMPLES

We first compare the stability conditions derived for a specific case. Suppose that the buffer length is given by \( \Lambda = 4 \), that \( p_j = 0.2 \), \( j \in \{0, \ldots, 4\} \), and that \( q = 0.75 \). The stability region boundaries, see (7) and (15), in terms of \( \alpha \) and \( \rho \) are depicted in Fig. 3. It can be seen that the guaranteed stable region (under the curve) provided by our results is larger when using Algorithm A1 than when using (4).

Next we consider an open-loop unstable constrained plant model of the form (1), but with additive noise:

\[
\begin{bmatrix}
    x_1(k+1) \\
    x_2(k+1)
\end{bmatrix} = \begin{bmatrix}
    x_2(k) + u_1(k) \\
    x_2(k) + u_2(k)
\end{bmatrix} + \begin{bmatrix}
    w_1(k) \\
    w_2(k)
\end{bmatrix}
\]

where

\[
sat(\mu) = \begin{cases} 
-10, & \text{if } \mu < -10, \\
\mu, & \text{if } \mu \in [-10, 10], \\
10, & \text{if } \mu > 10,
\end{cases}
\]

see [20, Example 2]. The initial condition \( x(0) \) and the disturbance \( w(k) \) are zero-mean i.i.d. Gaussian with unit covariance. The control policy \( \kappa \) is taken as \( \kappa(x) = [-x_2, 0.505sat(x_1 + x_2)]^T \), \( x \in \mathbb{R}^2 \).

If we choose \( V(x) = 2|x|^2 \), then direct calculations give that

\[
V(f(x, \kappa(x))) = 0.99|sat(x_1 + x_2)| \leq 0.99|x_1 + x_2| \\
\leq 1.98 \max\{|x_1|, |x_2| - \max\{|x_1|, |x_2|\} + |x| \leq 1.98|x|.
\]

Thus, Assumption 1 holds with \( \rho = 0.99 \), and \( \varphi_1(s) = \varphi_2(s) = 2s \).

Processor availability and \( \Lambda \) are taken as above, but we now set \( q = 0.4 \). Performance is evaluated through the empirical cost \( J = \frac{1}{N} \left( \sum_{k=0}^{N-1} |x(k)|^2 \right) \) and the Channel Utilization (%), calculated as

\[
\text{Total number of time steps at which } \beta(k) \neq 2 \quad (\%)
\]

By averaging over \( 10^4 \) realizations, Fig. 4 is obtained. As can be seen in that figure, the proposed event-based anytime control algorithm gives better trade-offs between empirical cost and channel utilization.

VIII. CONCLUSIONS

This work considered the control of a non-linear process with both communication and processing constraints. A sensor node transmits data to the controller across a channel that stochastically erases data. The control algorithm is executed using a processor that can provide only limited, time-varying and a priori unknown processing resources. To reduce the communication frequency, the sensor utilizes an event-triggered scheme. Similarly, to better utilize the processor availability, the control input is calculated by using an anytime control algorithm. For the resulting system, we present stochastic stability and stationarity results. Numerical studies illustrate that significant performance gains can be obtained by using the proposed algorithm. Future work includes the extension of the analysis to noisy systems, and establishing further stability properties such as ergodicity and rates of convergence to equilibrium.

REFERENCES

[1] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi, “Data transmission over networks for estimation and control,” IEEE Trans. Automat. Contr., vol. 54, pp. 1807–1819, Aug. 2009.
[2] O. C. Imer, S. Yüksel, and T. Başar, “Optimal control of LTI systems over unreliable communication links,” Automatica, vol. 42, pp. 1429–1439, Sept. 2006.
[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, “Foundations of control and estimation over lossy networks,” Proc. IEEE, vol. 95, pp. 163–187, Jan. 2007.
[4] D. E. Quevedo and D. Nesić, “Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses,” Automatica, vol. 48, pp. 1803–1811, Aug. 2012.
[5] L. Li, M. Lemmon, and X. Wang, “Event-triggered state estimation in vector linear processes,” in Proc. Amer. Contr. Conf., pp. 2138–2143, 2010.
[6] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” IEEE Trans. Automat. Contr., vol. 52, pp. 1680–1685, Sept. 2007.
[7] Y. Xu and J. Hespanha, “Optimal communication logics in networked control systems,” in Proc. IEEE Conf. Decis. Contr., pp. 3527–3532, 2004.
[8] C. Ramesh, H. Sandberg, and K. H. Johansson, “Steady state performance analysis of multiple state-based schedulers with CSMA,” in Proc. IEEE Conf. Decis. Contr., 2011.
[9] M. Xia, V. Gupta, and P. J. Antsaklis, “Networked state estimation over a shared communication medium,” in Proc. Amer. Contr. Conf., 2013.
[10] M. Rabi and K. H. Johansson, “Scheduling packets for event-triggered control,” in Proc. European Control Conf., pp. 3779–3784, 2009.
[11] R. Blind and F. Allgöwer, “Analysis of networked event-based control with a shared communication medium: Part 1 - pure aloha,” in Proc. IFAC World Cong., 2011.
[12] A. Cervin and T. Henningssson, “Scheduling of event-triggered controllers on a shared network,” in Proc. IEEE Conf. Decis. Contr., pp. 3601–3606, 2008.
[13] L. K. McGovern and E. Feron, “Closed-loop stability of systems driven by real-time dynamic optimization algorithms,” in Proc. IEEE Conf. Decis. Contr., vol. 4, (Phoenix, AZ), pp. 3690–3696, Dec. 1999.
[14] D. Henriksson and J. Åkesson, “Flexible implementation of model predictive control using sub-optimal solutions,” Tech. Rep. Internal Report No. TRF7-6710-SE, Dep. of Automatic Control, Lund University, 2004.
[15] P. Andriainaina, A. Seuret, and D. Simon, “Robust system control method with short execution deadlines.” European Patent Application EP 2 568 346 A1, Airbus Operations Toulouse, March 2013.
[16] A. Cervin, M. Velasco, P. Marti, and A. Camacho, “Optimal online sample-period assignment: Theory and experiments,” IEEE Trans. Contr. Syst. Technol., vol. 18, June 2010.
[17] R. Bhattacharya and G. J. Balas, “Anytime control algorithms: Model reduction approach,” AIAA Journal of Guidance, Control and Dynamics, vol. 27, pp. 767–776, Sept.–Oct. 2004.
[18] L. Greco, D. Fontanelli, and A. Bicchi, “Almost sure stability of anytime controllers via stochastic scheduling,” in Proc. IEEE Conf. Decis. Contr. (New Orleans, LA), pp. 5640–5645, Dec. 2007.
[19] V. Gupta and F. Luo, “On a control algorithm for time-varying processor availability,” IEEE Trans. Automat. Contr., vol. 58, Mar. 2013.
[20] D. E. Quevedo and V. Gupta, “Sequence-based anytime control,” IEEE Trans. Automat. Contr., vol. 58, pp. 377–390, Feb. 2013.
[21] C. Ramesh, H. Sandberg, and K. H. Johansson, “On the dual effect in state-based scheduling of networked control systems,” in Proc. Amer. Contr. Conf., pp. 2216–2221, 2011.
[22] D. E. Quevedo and V. Gupta, “Stability of sequence-based anytime control with Markovian processor availability,” in Proc. Austr. Contr. Conf., 2011.
[23] D. E. Quevedo, V. Gupta, W.-J. Ma, and S. Yüksel, “Stochastic stability of event-triggered anytime control.” http://arxiv.org/abs/1312.2390, 2013.
[24] S. P. Meyn, “Ergodic theorems for discrete time stochastic systems using a stochastic Lyapunov function,” SIAM Journal on Control and Optimization, vol. 27, pp. 1409–1439, Nov. 1989.
[25] S. Meyn and R. L. Tweedie, Markov Chains and Stochastic Stability. Cambridge University Press, 2009.
[26] S. Yüksel and S. P. Meyn, “Random-time, state-dependent stochastic drift for Markov chains and application to stochastic stabilization over erasure channels,” IEEE Trans. Automat. Contr., vol. 58, no. 1, pp. 47–59, 2013.