Event-Triggered Second Moment Stabilization under Markov Packet Drops

Shourya Bose Pavankumar Tallapragada

***Abstract*— In this paper, we consider the problem of second moment stabilization of scalar linear systems under Markov packet drops. We assume that the channel state evolution is given by a Markov chain. Corresponding to each possible value of the channel state the packet drop probability distribution may be different. We design an event-triggered transmission policy that ensures that the second moment of the plant state converges exponentially, at a desired rate, to an ultimate bound. Our approach relies on an online assessment of the necessity of transmission at any given time step and a transmission occurs only if necessary. We illustrate the results through simulations.**

I. INTRODUCTION

The area of Networked Control Systems (NCS) is concerned with controlling systems under communication constraints. In such systems, it becomes necessary to design control and communication in an integrated manner. To address this need, the framework of event-triggered control has gained a lot of popularity in the last decade. In this paper, we study the problem of designing a policy for event-triggered transmission over a Markov channel for second moment stabilization of a plant.

Literature Review: Research work on NCS spanning nearly the last two decades has been extensive and includes results that give fundamental limits, control design methods under various network effects and using various methodologies. Some important survey papers on this area include [1]–[4], while [5] is a book on stochastic NCS. A particularly useful theme that has emerged in the area of NCS in the last decade is that of event-triggered control (along with the related ideas such as self-triggered control). A good introduction to the area is [6] and [7]–[10] provide a fairly comprehensive survey of the current literature.

However, state-triggered control in a stochastic setting is still relatively limited. [11]–[17] explore finite or infinite horizon optimal control problems, typically with fixed threshold-based triggering. [13]–[16] also consider packet drops. Stochastic stability, in the sense of moment stability, with event-triggered control has received even less attention. The work [18] designs self-triggered sampling for second-moment stability of state-feedback controlled stochastic differential equations and [19] proposes a fixed threshold-based event-triggered anytime control policy under packet drops. It assumes that the controller has knowledge of the transmission times, including when a packet is dropped, and the policy guarantees second-moment stability with exponential convergence to a finite bound asymptotically. The current paper builds on [20], [21], which consider the problem of second moment stabilization of the plant state under independent identically distributed (i.i.d.) Bernoulli packet drops. The main difference here is that we consider a channel with Markov packet drops. In the context of NCS, a channel with i.i.d. Bernoulli packet drops is a very popular choice for modeling uncertain channels. However, channels that evolve according to a Markov chain are a generalization and better model real communication phenomena [22]–[24]. Thus, in recent years there have been some papers that explore estimation [25], [26] or NCS under Markov packet drops [27]–[29] or under Markov bit rates [30]. However, these papers are all in the context of time-triggered control.

Contributions: We consider the problem of second moment stabilization of a scalar system under process noise and Markov packet drops. In particular, we assume that the channel state at any time can be in one of a finite number of states. Corresponding to each state, there is an associated Bernoulli packet drop probability distribution. We assume that the channel state evolves according to a Markov chain on the finite states. Then, we design an event-triggered transmission policy (along the lines of the two-step design principle proposed in [21] for control over a channel with i.i.d. Bernoulli packet drops) that achieves second moment stability of the plant state with a desired rate of convergence. Thus, this paper is an important generalization of [20], [21].

Notation: We let $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{N}_0$ denote the set of real numbers, positive integers and non-negative integers respectively. We use boldface to represent vectors and matrices. In particular $\mathbf{1}$ is the vector of all ones, $\boldsymbol{\delta}_i$ represents the n-dimensioned vector which assumes value 1 at index i and 0 everywhere else, and $\mathbf{I}_n$ is the identity matrix of order n. $\rho(\mathbf{P})$ represents the spectral radius of the matrix $\mathbf{P}$. Finally, $\mathbb{E}_{\mathcal{T}}[.]$ is the expectation under the transmission policy $\mathcal{T}$.

II. PROBLEM STATEMENT

In this section, we first set up the various elements of the system including the plant, the sensor, the controller as well as the Finite-State Markov Channel (FSMC). Building on this, we present the main problem statement.

A. System Overview

Consider a scalar, linear time-invariant system, given by

$$x_{k+1} = ax_k + u_k + v_k, \quad x_k, u_k, v_k, a \in \mathbb{R} \tag{1}$$

for all $k \in \mathbb{N}_0$. Here, x_k, u_k and v_k are the plant state, external control effort applied by the actuator, and the process noise at timestep k, respectively. We assume that $\{v_k\}$ is independent identically distributed process with zero mean and a finite variance M for all time $k \in \mathbb{N}_0$. The parameter a

S. Bose is with Department of EEE&I, BITS Pilani, KK Birla Goa Campus (f20140692@goa.bits-pilani.ac.in) and P. Tallapragada is with the Department of Electrical Engineering, Indian Institute of Science (pavant@iisc.ac.in)

is the inherent plant gain, and further we assume that $|a| > 1$ so that control is necessary.

The sensor is not collocated with the controller, but rather communicates with it over an unreliable communication channel. At each timestep k, the sensor measures the plant state x_k. The sensor decides whether to transmit or not on each time step k based on the control objective, the nature of the unreliable communication channel, and past history. Thus, we define the *transmission process* $\{t_k\}_{k\in\mathbb{N}_0}$ as

$$t_k := \begin{cases} 1, & \text{if sensor transmits on time } k \\ 0, & \text{if sensor does not transmit on time } k. \end{cases} \tag{2}$$

The algorithm which computes this decision is called a *transmission policy*, denoted by $\mathcal{T}$. The channel is unreliable and hence we define the *reception process* $\{r_k\}_{k\in\mathbb{N}_0}$ as

$$r_k := \begin{cases} 1, \text{ if } t_k = 1 \text{ and successful transmission} \\ 0, \text{ if } t_k = 1 \text{ and unsuccessful transmission} \\ 0, \text{ if } t_k = 0 \end{cases} \tag{3}$$

After a successful reception, the controller acknowledges the reception (we assume a perfect feedback channel for the acknowledgment). Thus the sensor knows r_{k-1} while computing t_k for timestep k. We further define the *latest reception time before* k and *latest reception time up to* k as

$$R_k := \max\{i < k : r_i = 1\}, \quad R_k^+ := \max\{i \le k : r_i = 1\}.$$

Note that $R_k = R_k^+$ for any given k if $r_k = 0$. We also denote the sequence of all successful reception times as follows

$$S_0 = 0, \quad S_{j+1} := \min\{k > S_j : r_k = 1\}.$$

We let $\hat{x}_k^+$ be the *controller state* at timestep k. Based on this, the controller produces a control action $u_k = L\hat{x}_k^+$ where $L \in \mathbb{R}$. If $r_k = 1$, then the controller knows x_k exactly, and hence on such k we let $\hat{x}_k^+ = x_k$. When $r_k = 0$, the controller uses an estimate of the plant state, $\hat{x}_k$, to generate the control effort. Thus, $\hat{x}_k^+ = \hat{x}_k$ when $r_k = 0$. This estimate of the plant state is calculated as $\hat{x}_k = \bar{a}\hat{x}_{k-1}^+$, where $\bar{a} = a + L$. Note that at time k, since sensor knows the reception process until time $k-1$, it can compute $\hat{x}_{k-1}^+$ and $\hat{x}_k$ independently. Thus, we also call $\hat{x}_k$ as the *sensor's estimate* at time k. Corresponding to these, we define the *sensor's estimation error* and the *controller state error* respectively as $z_k := x_k - \hat{x}_k$, and $z_k^+ := x_k - \hat{x}_k^+$. Thus, the overall *system evolution* is

$$x_{k+1} = ax_k + L\hat{x}_k^+ + v_k = \bar{a}x_k - Lz_k^+ + v_k \tag{4a}$$
$$\hat{x}_{k+1} = \bar{a}\hat{x}_k^+ \tag{4b}$$

where $\bar{a} = a + L$, $z_k^+ = x_k - \hat{x}_k^+$, and

$$\hat{x}_k^+ = \begin{cases} \hat{x}_k, & \text{if } r_k = 0 \\ x_k, & \text{if } r_k = 1. \end{cases} \tag{4c}$$

With the above setup, we can define two sets of *information* available to the sensor. At time k, *before* the sensor has decided t_k, the information available to it can be denoted as

$$I_k := \{k, x_k, z_k, R_k, x_{R_k}\},$$

while after the transmit-acknowledge process is complete, the sensor has access to the following information

$$I_k^+ := \{k, x_k, z_k^+, R_k^+, x_{R_k^+}\}.$$

Recall that the sensor receives a perfect acknowledgement when $r_k = 1$. Thus, we can assume that the sensor also knows I_k^+, though after the transmission on time k or after the decision by the sensor not to transmit. Intuitively, I_k is the information used by the sensor to decide t_k, whereas I_k^+ allows the sensor to monitor the system performance. Furthermore, note that $I_k = I_k^+$ whenever $r_k = 0$.

B. Communication Channel

We model the unreliable communication channel as a Finite-State Markov Channel (FSMC). We assume that at time k, the state of the communication channel, γ_k can take values out of a set of n states given by $\mathcal{L} := \{l_1, l_2, \cdots, l_n\}$. We assume that the evolution of γ_k is Markov. Thus,

$$Pr[\gamma_{k+1} = l_i | \gamma_k = l_j] = p_{ij}$$

Thus, if the probability distribution of the channel state at time k is $\mathbf{p}_k$, then we have

$$\mathbf{p}_{k+1} = \mathbf{P}\mathbf{p}_k, \tag{5}$$

where $\mathbf{P} = \{p_{ij}\}$ is the transition kernel for the channel state change between timesteps k and $k+1$. We assume that the transition kernel is invariant across time, which is a good first-order approximation for flat-fading wireless channels [31]. With each state of the FSMC, we associate a Bernoulli packet drop model for the channel. In particular, if the channel is in state j, then the probability of successful communication of a packet over the channel is d_j, and the probability dropping the packet is $e_j := 1 - d_j$. Thus,

$$\mathbf{d} := [d_1, d_2, \cdots, d_n]^T \tag{6a}$$
$$\mathbf{e} := \mathbf{1} - \mathbf{d} = [e_1, e_2, \cdots, e_n]^T. \tag{6b}$$

Further, we assume that before the sensor decides t_k, it knows γ_{k-1} the state of the channel at timestep $k-1$. Practically, identification of FSMC state at the receiver (controller) can be achieved using techniques such as *symbol training* (also referred to in literature as *pilot training*) to keep an updated estimate of the state of the channel at every timestep. A practical algorithm for symbol training in flat-fading wireless channels is provided in [32]. Further characterization of state-recognition principles in FSMC's is provided in [31]. Thus, we assume that the controller communicates the identified FSMC state back to the sensor through a low bandwidth reliable feedback channel.

C. Control Objective

We formulate the policy $\mathcal{T}$, for deciding the sequence of transmission times $\{t_k\}_{k\in\mathbb{N}_0}$, based on the *control objective* that we want to achieve for the system. Since the system is stochastic in nature, we let the control objective to be the *exponential convergence of the second-moment of the plant state to an ultimate bound*. That is, we require that

$$\mathbb{E}_{\mathcal{T}}[x_k^2 | I_0^+] \le \max\{c^{2k}x_0^2, B\}, \quad \forall\, k \in \mathbb{N}_0. \tag{7}$$

Here, $\mathcal{T}$ is the transmission policy that determines $\{t_k\}$, $c^2 < 1$ is the desired exponential convergence rate and B is the ultimate bound of the second moment. Thus, the right hand side of (7) defines an envelope within which the second moment of the plant state must evolve. However, this control objective is a specification on the entire trajectory of the system from time $k = 0$. Instead, we borrow Lemma 3.1 from [20], which defines a stronger control objective on the system in an *online* fashion, rather than for the entire trajectory in an "open-loop" manner.

Lemma 1: (Alternate Control Objective [20]). For a transmission policy $\mathcal{T}$, consider the control objective

$$\mathbb{E}_{\mathcal{T}}\left[h_k | I^+_{R_k}\right] \leq 0, \ \forall k \in \mathbb{N}, \tag{8}$$

where h_k is the *performance function* and we define it as

$$h_k := x_k^2 - \max\{c^{2(k-R_k)} x^2_{R_k}, B\}.$$

If a transmission policy satisfies the *online objective* (8) then it also satisfies the objective (7). ■

The main idea behind Lemma 1 is that if a transmission policy meets (8) between each successive reception times $S_j \leq k \leq S_{j+1}$, then it also satisfies (7). Thus, throughout the rest of the paper, we focus on the online objective (8). Next, we highlight the main idea behind our design strategy.

III. The Two-Step Transmission Policy Design

Designing an event-triggered transmission policy even for the online objective (8) is challenging. This is because the random packet drops make the evaluation of the necessity of a transmission on a timestep coupled with the future state evolution as well as the future actions. Therefore, we adopt a two-stage approach. In the first stage, we consider a *nominal policy* that does not transmit for D timesteps from the current time k, and transmits at every timestep thereafter. If the nominal policy can maintain the control objective at time k, then we know that there exists *some* policy which will maintain the control objective even if $t_k = 0$. This forms the main idea for the design of our event-triggered transmission policy. In the event-triggered transmission policy, the sensor evaluates the necessity of transmitting on the current time k based on the nominal transmission policy as described above. If under the nominal policy, the control objective can be met then there is no actual transmission (because a transmission is not necessary). If on the other hand, the control objective is not met under the nominal policy starting at the current time, then the sensor actually transmits. The sensor executes this two stage process on each timestep. Thus, we use a time-based nominal policy as a building block to obtain the composite *event-triggered policy*. [20] contains a more elaborate discussion of this design idea and the motivation for it. Next, we give the specifics of the nominal policy, and subsequently the event-triggered transmission policy.

A. The Nominal Policy, $\mathcal{T}_k^D$

We define the nominal policy starting from time k with parameter D, denoted by $\mathcal{T}_k^D$, as

$$\mathcal{T}_k^D : t_i = \begin{cases} 0, & i \in \{k, k+1, \cdots, k+D-1\} \\ 1, & i \geq k+D. \end{cases} \tag{9}$$

We associate with the nominal policy $\mathcal{T}_k^D$ the *look-ahead criterion*, which is the conditional expectation (under the policy $\mathcal{T}_k^D$) of the performance function at the next successful reception timestep. Thus, the look-ahead criterion is

$$\mathcal{G}_k^D := \mathbb{E}_{\mathcal{T}_k^D}\left[h_{S_{j+1}} | I_k, S_j = R_k\right]. \tag{10}$$

From the structure of the nominal policy, we get

$$\mathcal{G}_k^D = \sum_{w=D}^{\infty} \mathbb{E}_{\mathcal{T}_k^D}\left[h_{S_{j+1}} | I_k, S_j = R_k, S_{j+1} = k+w\right] \Phi_D(w, \mathbf{p}_k), \tag{11}$$

where recall that $\mathbf{p}_k$ is the state probability distribution of the channel state at timestep k and $\Phi_D(w, \mathbf{p})$ is defined as

$$\Phi_D(w, \mathbf{p}) := \Pr[S_{j+1} = k+w | \mathcal{T} = \mathcal{T}_k^D, \mathbf{p}_k = \mathbf{p}, S_j = R_k],$$

which is the probability that the first successful reception after time k occurs on timestep $k+w$, given the channel state distribution at timestep k is $\mathbf{p}$ and the transmission policy is the nominal policy $\mathcal{T}_k^D$. Also, note that according to the channel evolution described in (5), $\mathbf{p}_k$ is given by

$$\mathbf{p}_k = \mathbf{P}\boldsymbol{\delta}_{\gamma_{k-1}} \tag{12}$$

where γ_{k-1} is the channel state at time $k-1$, which the sensor knows at time k before deciding t_k. Note that $\boldsymbol{\delta}_i$ is a probability distribution. The closed form of $\Phi_D(w, \mathbf{p})$ is

$$\Phi_D(w, \mathbf{p}) = \mathbf{1}^T \mathcal{D} (\mathbf{P}\mathcal{E})^{(w-D)} \mathbf{P}^D \mathbf{p} \tag{13}$$

where $\mathcal{D} := \text{diag}(\mathbf{d})$ and $\mathcal{E} := \text{diag}(\mathbf{e})$. The explanation for (13) is as follows: left multiplying the channel state distribution $\mathbf{p}$ with $\mathbf{P}^D$ gives us the channel state distribution at timestep $k+D$ at the end of the idle time in the policy $\mathcal{T}_k^D$, during which the sensor does not transmit. Multiplying this distribution by $(\mathbf{P}\mathcal{E})^{w-D}$ gives us a vector whose j^{th} element is the probability that at timestep $k+w$, the channel state is l_j given that the sensor transmits at each timestep from $k+D$ to $k+w$ but reception is unsuccessful every time up to $k+w-1$. Pre-multiplying this vector with $\mathcal{D}$ gives a vector where the j^{th} element denotes the probability that the transmission over the channel was finally successful at the $(k+w)^{\text{th}}$ timestep. Left multiplying this vector by $\mathbf{1}^T$ sums the probabilities over all the channel states.

B. The Event-Triggered Policy, $\mathcal{T}_{\mathcal{E}}$

From the previous section, we see that the sign of the *look-ahead criterion* indicates whether or not the online objective (8) would be violated if, starting from the current time, the sensor does not transmit for the next D timesteps. This gives us a simple criteria to decide whether or not to transmit at timestep k. If $\mathcal{G}_k^D$ is negative, then the sensor does not transmit - because there is ("on average") a leeway of D timesteps for course correction. If $\mathcal{G}_k^D$ is positive, then the sensor does not have the aforementioned leeway, and therefore, it transmits at k to avoid violation of (8). Thus, the event-triggered policy is represented as

$$\mathcal{T}_{\mathcal{E}} := t_k = \begin{cases} 0, & \text{if } k \in \{S_j + 1, \cdots, T_j - 1\}, \\ 1, & \text{if } k \in \{T_j, \cdots, S_{j+1}\}, \end{cases} \tag{14}$$

where

$$T_j := \min\{k > S_j : \mathcal{G}_k^D \geq 0\}. \tag{15}$$

IV. Implementation and Convergence Analysis of the Event-Triggered Policy

A. Evaluation of the Lookahead Function, $\mathcal{G}_k^D$

Even though (11) gives us the definition of $\mathcal{G}_k^D$, we still need a closed form expression of $\mathcal{G}_k^D$ in order to compute it efficiently onboard the sensor at every time. The first step to finding a closed form for $\mathcal{G}_k^D$ would be to tackle the expectation term in the summation of (11). To this end, we borrow the following result from Lemma 4.1 of [20].

Lemma 2: The closed form of the term $\mathbb{E}_{\mathcal{T}_k^D}[h_{k+w}|I_k]$ for a nominal policy with parameter D is given as

$$\begin{aligned}&\mathbb{E}_{\mathcal{T}_k^D}\left[h_{S_{j+1}}|I_k, S_j = R_k, S_{j+1} = k+w\right] = \\ &\bar{a}^{2w}x_k^2 + 2\bar{a}^w(a^w - \bar{a}^w)x_k z_k + (a^{2w} - 2a^w\bar{a}^w + \bar{a}^{2w})z_k^2 + \\ &\bar{M}(a^{2w} - 1) - \max\{c^{2w}c^{2(k-R_k)}x_{R_k}^2, B\},\end{aligned} \tag{16}$$

where $\bar{M} := \frac{M}{a^2-1}$. ■

As is evident from (11) and (16), we need to prove the existence of the summation of the form

$$\chi_D(b,\mathbf{p}) := \sum_{w=D}^{\infty} b^w \Phi_D(w,\mathbf{p}), \tag{17}$$

which we obtain using the following lemma.

Lemma 3: Assume $\mathbf{P}$ is a left-stochastic matrix, $\mathcal{E} = \text{diag}(\mathbf{e})$ where $\mathbf{e} \in [0,1]^n$ and $b > 0$. Then $\rho(b\mathbf{P}\mathcal{E}) \leq b\bar{e}$, where $\bar{e} := \{\max(e_i), i \in \{1,\dots,n\}|\mathbf{e} = \{e_i\}_{i=1}^n\}$.

The proof of the Lemma follows directly from Corollary 1.2, [33]. As a direct application of Lemma 3 and a result on convergence of geometric series of matrices [34] (pp 193, Theorem 2), we can obtain the closed form of $\chi_D(b,\mathbf{p})$.

Lemma 4: Suppose that $e_i \in [0,1)$ for each $i \in \{1,\dots,n\}$ and that $b \in [0,1]$. Then, the series defining $\chi_D(b,\mathbf{p})$ converges and its closed form is given as

$$\chi_D(b,\mathbf{p}) = b^D \mathbf{1}^T \mathcal{D}\left(\mathbf{I}_n - b\mathbf{P}\mathcal{E}\right)^{-1}\mathbf{P}^D\mathbf{p}. \quad ■ \tag{18}$$

1) Properties of $\chi_D(b,\mathbf{p})$: In this subsection, we note a few properties of the function $\chi_D(b,\mathbf{p})$.

Lemma 5: $\chi_D(b,\mathbf{p})$ is strictly monotonically increasing in b for all values of $b > 0$.

Proof: In (17), $b^w\Phi_D(w,\mathbf{p})$, for each w is nonnegative and strictly increasing in b. ■

We now present a condition for the lookahead function $\mathcal{G}_k^D$ to exist.

Lemma 6: $\mathcal{G}_k^D$ exists if $a^2\bar{e} < 1$.

Proof: From (11), (16) and (17), we see that to find the closed form of $\mathcal{G}_k^D$, $\chi_D(b,\mathbf{p})$ would have to be evaluated for $b = a^2, a\bar{a}, \bar{a}^2$ and c^2. Since a^2 is biggest of these values based on our standing assumptions, the range of possible values of b is $(0,a^2]$. Since $\chi_D(b,\mathbf{p})$ is monotonically increasing in b for all $b > 0$ (Lemma 5), it is sufficient to ensure the existence of $\chi_D(a^2,\mathbf{p})$. However, from Lemma 4, we see that $\chi_D(a^2,\mathbf{p})$ would converge if $a^2\bar{e} < 1$, thus completing the proof. ■

2) Closed form of $\mathcal{G}_k^D$: The closed form of $\mathcal{G}_k^D$, which can be used onboard the sensor to evaluate the value of $\mathcal{G}_k^D$ on every timestep $k \in \mathbb{N}$, is given as follows.

Theorem 1: Suppose $a^2\bar{e} \in [0,1)$. Then $\mathcal{G}_k^D$ is well-defined and its closed form is given as

$$\begin{aligned}\mathcal{G}_k^D = &\chi_D(\bar{a}^2,\mathbf{p}_k)x_k^2 + 2\left[\chi_D(a\bar{a},\mathbf{p}_k) - \chi_D(\bar{a}^2,\mathbf{p}_k)\right]x_k z_k \\ &\left[\chi_D(a^2,\mathbf{p}_k) - 2\chi_D(a\bar{a},\mathbf{p}_k) + \chi_D(\bar{a}^2,\mathbf{p}_k)\right]z_k^2 + \\ &\bar{M}\left[\chi_D(a^2,\mathbf{p}_k) - \chi_D(1,\mathbf{p}_k)\right] - \left[\mu_k(c^2)\chi_D(c^2,\mathbf{p}_k)\right. \\ &\left.+B\tilde{\chi}_D(1,\mathbf{p}_k) - \mu_k(c^2)\tilde{\chi}_D(c^2,\mathbf{p}_k)\right]\end{aligned}$$

where $\bar{M} := \frac{M}{a^2-1}$, and $\tilde{\chi}_D(b,\mathbf{p}_k)$ is defined as

$$\tilde{\chi}_D(b,\mathbf{p}_k) := b^{(D+q_k^D)}\mathbf{1}^T\mathcal{D}(\mathbf{P}\mathcal{E})^{q_k^D}\left(\mathbf{I}_n - b\mathbf{P}\mathcal{E}\right)^{-1}\mathbf{P}^D\mathbf{p}_k$$

$\mu_k(b)$ is given by

$$\mu_k(b) := b^{(k-R_k)}x_{R_k}^2$$

and q_k^D is defined as

$$q_k^D := \max\left\{0, \left\lceil\frac{\log(x_{R_k}^2/B)}{\log(1/c^2)}\right\rceil - (k-R_k) - D\right\}$$

Proof: The evaluation of terms in (11) of the form $\sum_{w=D}^{\infty} b^w\phi_D(w,\mathbf{p}_k)$ equals $\chi_D(b,\mathbf{p}_k)$, which exists and is given in (18). What remains is the evaluation of the term $\sum_{w=D}^{\infty}\max\left\{c^{2w}c^{2(k-R_k)}x_{R_k}^2, B\right\}\Phi_D(w,\mathbf{p}_k)$. Let $\mu_k(b) := b^{k-R_k}x_{R_k}^2$. Then, we have

$$\begin{aligned}&\sum_{w=D}^{\infty}\max\{b^w z_k(b), B\}\Phi_D(w,\mathbf{p}_k) = \\ &\mu_k(b)\sum_{w=D}^{\infty}b^w\Phi_D(w,\mathbf{p}_k) + \sum_{w=D+q_k^D}^{\infty}(B - b^w\mu_k(b))\Phi_D(w,\mathbf{p}_k),\end{aligned}$$

where q_k^D is the number of timesteps from k when we first have $B > b^w\mu_k(b)$. We can verify that the value of q_k^D for $b = c^2$ is the same as provided in the theorem. The first summation in the RHS is equal to $\mu_k(b)\chi_D(b,\mathbf{p}_k)$. Using the definition of $\tilde{\chi}_D(b,\mathbf{p})$ in the theorem statement, we see that the second summation on the right is equal to $B\tilde{\chi}_D(1,\mathbf{p}_k) - \mu_k(b)\tilde{\chi}_D(b,\mathbf{p}_k)$. This verifies the closed form of $\mathcal{G}_k^D$ as given in the theorem statement. ■

B. The Performance Evaluation Function, $\mathcal{J}_k^D$

Recall from Lemma 1 that we seek to ensure the online objective (8) in order to ensure (7). Thus, in what follows, we demonstrate that the event-triggered policy (14) meets the online objective (8). To this end, we introduce *performance evaluation function*, denoted by $\mathcal{J}_k^D$, as

$$\mathcal{J}_k^D := \mathbb{E}_{\mathcal{T}_k^D}\left[h_{S_{j+1}}|I_k^+, S_j = R_k^+\right] \tag{19}$$

$$= \sum_{w=D}^{\infty} H(w,x_k^2)\Phi_D(w,\mathbf{p}_k) \tag{20}$$

where

$$H(w,x_k^2) := \mathbb{E}_{\mathcal{T}_k^D}\left[h_{S_{j+1}}|I_k^+, S_j = R_k^+, S_{j+1} = k+w\right]$$

is the *open-loop performance evolution* function. Applying Lemma 2 to (20) for $k = S_j$, we see that

$$H(w, x_{S_j}^2) = \bar{a}^{2w} x_{S_j}^2 + \bar{M}(a^{2w} - 1) - \max\{c^{2w} x_{S_j}^2, B\}.$$

It can be observed that the performance evaluation function evaluated at a successful reception time, $\mathcal{J}_{S_j}^D$ gives us information about whether or not the system would breach the control objective on the next reception time under the nominal policy (9). Note that $\mathcal{J}_k^D$ differs from $\mathcal{G}_k^D$ only if $k = R_k^+ = S_j$ for some j. Analogous to the closed form expression of $\mathcal{G}_k^D$ in Theorem 1, we can also obtain a closed form expression for $\mathcal{J}_k^D$, which we skip here due to space constraints. Crucial to the analysis of the event-triggered transmission policy is the behaviour of the function $H(w, y)$ described in the following result.

Theorem 2: (Sign Monotonicity of $H(w, y)$ [21], Prop. 4.6). There exists a $B^* > 0$ such that, if $B > B^*$ and $B \log\left(\frac{c^2}{\bar{a}^2}\right) > \bar{M} \log a^2$, then for all $y \geq 0$, the function $H(., y)$ has the property

$$H(w_1, y) > 0 \implies H(w_2, y) > 0, \;\; \forall w_2 \geq w_1. \qquad \blacksquare$$

Note that B^* can be computed numerically as described in [21]. It should be noted here that the use of the term monotonicity is not used for the function $H(., y)$ *per se* but rather for the behaviour of the system when evolving over a string of timesteps with lack of communication. Insofar as the system is considered, Theorem 2 demonstrates that faced with the lack of successful receptions, the system will satisfy the control goal for a finite number of steps, and then violate it at every step thereafter. Now the only thing left for us to prove to demonstrate that our scheme satisfies the control goal is to show that the value of the performance evaluation function is negative at every successful reception timestep. The following theorem demonstrates the same.

Theorem 3: We have the following

(a) $\mathbb{E}_{\mathcal{T}}\left[\mathcal{G}_{k+1}^D | I_k, t_k = 0, r_k = 0\right] = \mathcal{G}_k^{D+1}$

$\mathbb{E}_{\mathcal{T}}\left[\mathcal{G}_{k+1}^D | I_k, t_k = 1, r_k = 1\right] = \mathcal{J}_k^{D+1}$

(b) Suppose

$$\left(\chi_D(\bar{a}^2, \boldsymbol{\delta}_i) - \chi_D(c^2, \boldsymbol{\delta}_i)\right) \frac{B}{c^{2D}} + \bar{M}\left(\chi_D(a^2, \boldsymbol{\delta}_i) - \chi_D(1, \boldsymbol{\delta}_i)\right) < 0, \; \forall i \in \{1, \ldots, n\} \tag{21}$$

Then $\mathcal{J}_{S_j}^D < 0$.

(c) If the previous proposition is true, then $\mathcal{J}_{S_j}^\theta \leq \mathcal{J}_{S_j}^{\theta+1}$ for all $\theta \in \{1, \ldots, D-1\}$ and for all $j \in \mathbb{N}_0$.

Proof: The proofs of parts (a) and (b) follow a very similar structure as the proofs of [20], Proposition 4.4 parts (a) and (b). Thus, we skip their proofs due to space limitations. Now we prove (c). Observe from (13) that

$$\Phi_\theta(w, \mathbf{p}) = \mathbf{1}^T(\mathcal{D}\mathbf{P})(\mathcal{E}\mathbf{P})^{(w-\theta-1)}\mathcal{E}\mathbf{P}^\theta\mathbf{p}$$
$$\Phi_{\theta+1}(w, \mathbf{p}) = \mathbf{1}^T(\mathcal{D}\mathbf{P})(\mathcal{E}\mathbf{P})^{(w-\theta-1)}\mathbf{P}^\theta\mathbf{p}.$$

Thus, we find that $\Phi_\theta(\theta+w, \mathbf{p}) \leq \bar{e}\Phi_{\theta+1}(\theta+w, \mathbf{p}), \; \forall w \geq 1$. Now consider (20) with parameter θ and letting $y = x_{S_j}^2$

$$\begin{aligned}
\mathcal{J}_{S_j}^\theta &= H(\theta, y)\Phi_D(\theta, \mathbf{p}_{S_j}) + \sum_{w=\theta+1}^{\infty} H(w, y)\Phi_\theta(w, \mathbf{p}_{S_j}) \\
&\leq H(\theta, y)\Phi_D(\theta, \mathbf{p}_{S_j}) + \bar{e}\sum_{w=\theta+1}^{\infty} H(w, y)\Phi_{\theta+1}(w, \mathbf{p}_{S_j}) \\
&= H(\theta, y)\Phi_D(\theta, \mathbf{p}_{S_j}) + \bar{e}\mathcal{J}_{S_j}^{\theta+1}
\end{aligned}$$

If $\theta \leq D$ and if D satisfies (21), then by the claim of part (b) and by Theorem 2, we find out that $H(\theta, x_{S_j}^2)$ is negative. Since $\bar{e} \in [0, 1]$, the result follows. $\blacksquare$

All of the aforementioned results culminate in the following result, whose proof we skip due to space limitations as it follows a similar structure as that of Theorem V.1 in [21].

Theorem 4: For a suitable value of B and D calculated according to theorem 2 and theorem 3, part (2), the event triggered policy satisfies the control objective in equation 7.

V. Simulation Results

For the purpose of simulation, we consider the four state wireless Markov channel proposed in [35] which includes both Ricean (LOS) and Rayleigh (non-LOS) fading effects in an indoor IEEE 801.11b WiFi network. We utilise the value of the four state transition kernel mentioned in the aforementioned paper. Correspondingly, we choose our state error probability vector $\mathbf{e} \in [0, 1]^4$ as

$$\mathbf{e} = \begin{bmatrix} 0.2244 & 0.3128 & 0.1031 & 0.5400 \end{bmatrix}^T \tag{22}$$

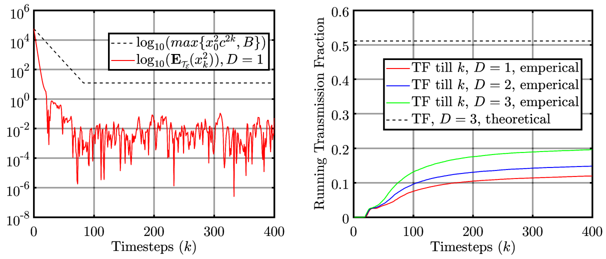
For our system, we choose the following parameters

$$a = 1.15, M = 1, c = 0.95, \bar{a} = 0.78c, B = 12, x(0) = 20B.$$

We observe that $B^* = 2.0975$ as defined in Theorem 2. The method to numerically calculate B^* is presented in [21]. Furthermore, according to Theorem 3, (b), we see that $D = 1, 2, 3$ satisfy the criterion under which the event triggered policy guarantees the achievement of the control goal in expectation. We ran simulations of the system with the aforementioned parameters on MATLAB, and averaged the outcomes over 500 independent runs to create empirical results on the system evolution, which is now presented here. Figure 1(a) shows that the the event-triggered policy meets the control objective by ensuring the second moment converges at the desired exponential rate to the desired ultimate bound. We define the *running transmission fraction* (denoted as $\mathcal{F}_0^k$) as the ratio of expected number timesteps when the sensor attempts to transmit under the event-triggered policy to the total number of elapsed timesteps, that is,

$$\mathcal{F}_0^k := \frac{1}{k}\mathbb{E}_{\mathcal{T}_\mathcal{E}}\left[\sum_{i=0}^{k} t_i\right].$$

We provide the empirical values of the running transmission fraction in Figure 1(b) for the current system. Further, with a variation of the methods developed in [21], it is possible to calculate an upper bound on the *infinite transmission fraction* ($\mathcal{F}_0^\infty$), which we also indicate in Figure 1(b).



(a) (b)

Fig. 1. (a) Plot of emperical system evolution for $D = 1$. (b) Plot of Running Transmission Fraction for $D = 1, 2, 3$ and Transmission Fraction upper bound for $D = 3$.

VI. Conclusions

In this paper, we considered a scalar linear plant with process noise, with the sensor and the controller non-collocated. We assumed that the sensor communicates over an unreliable channel with Markov packet drops. Under this setting, we designed an event-triggered transmission policy to ensure second-moment stability with a desired convergence rate and ultimate bound. We provided a rigorous analysis of the convergence properties and illustrated the results through simulations. Future work in this direction will include rigorous characterization of performance of the current algorithm, analysis of the transmission fraction, design of policies that minimize the transmission fraction, consideration of Markov Decision Process channels, and exploration of challenges in extending the current algorithm to the vector case.

References

[1] J. Hespanha, P. Naghshtabrizi, and Y. Xu, “A survey of recent results in networked control systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[2] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, “Feedback control under data rate constraints: an overview,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[3] Y. Ke-You and X. Li-Hua, “Survey of recent progress in networked control systems,” *Acta Automatica Sinica*, vol. 39, no. 2, pp. 101–117, 2013.

[4] M. Franceschetti and P. Minero, “Elements of information theory for networked control systems,” in *Information and Control in Networks*, G. Como, B. Bernhardsson, and A. Rantzer, Eds. New York: Springer, 2014, vol. 450, pp. 3–37.

[5] S. Yüksel and T. Başar, *Stochastic Networked Control Systems: Stabilization and Optimization under Information Constraints*, ser. Systems & Control: Foundations & Applications. Boston, MA: Birkhäuser, 2013.

[6] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *IEEE Conf. on Decision and Control*, Maui, HI, 2012, pp. 3270–3285.

[7] M. Lemmon, “Event-triggered feedback in control, estimation, and optimization,” in *Networked control systems*. Springer, 2010, pp. 293–358.

[8] L. Zou, Z. D. Wang, and D. H. Zhou, “Event-based control and filtering of networked systems: A survey,” *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 239–253, 2017.

[9] C. Peng and F. Li, “A survey on recent advances in event-triggered communication and control,” *Information Sciences*, vol. 457-458, pp. 113 – 125, 2018.

[10] D. Tolić and S. Hirche, *Networked Control Systems with Intermittent Feedback*. Crc Press, 2017.

[11] T. Henningsson, E. Johannesson, and A. Cervin, “Sporadic event-based control of first-order linear stochastic systems,” *Automatica*, vol. 44, no. 11, pp. 2890–2895, 2008.

[12] X. Meng and T. Chen, “Optimal sampling and performance comparison of periodic and event based impulse control,” *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3252–3259, 2012.

[13] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, “On the trade-off between control performance and communication cost in event-triggered control,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2973–2980, 2017.

[14] M. Rabi and K. H. Johansson, “Scheduling packets for event-triggered control,” in *European Control Conference*, Budapest, Hungary, Aug. 2009, pp. 3779–3784.

[15] R. Blind and F. Allgöwer, “The performance of event-based control for scalar systems with packet losses,” in *IEEE Conf. on Decision and Control*, Maui, HI, Dec. 2012, pp. 6572–6576.

[16] M. H. Mamduhi, D. Tolić, A. Molin, and S. Hirche, “Event-triggered scheduling for stochastic multi-loop networked control systems with packet dropouts,” in *IEEE Conf. on Decision and Control*, Los Angeles, CA, Dec. 2014, pp. 2776–2782.

[17] A. Molin and S. Hirche, “On the optimality of certainty equivalence for event-triggered control systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 470–474, 2013.

[18] R. P. Anderson, D. Milutinović, and D. V. Dimarogonas, “Self-triggered sampling for second-moment stability of state-feedback controlled SDE systems,” *Automatica*, vol. 54, pp. 8–15, 2015.

[19] D. E. Quevedo, V. Gupta, W. Ma, and S. Yüksel, “Stochastic stability of event-triggered anytime control,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3373–3379, 2014.

[20] P. Tallapragada, M. Franceschetti, and J. Cortés, “Event-triggered stabilization of scalar linear systems under packet drops,” in *Allerton Conf. on Communications, Control and Computing*, 2016, pp. 1173–1180.

[21] ——, “Event-triggered second-moment stabilization of linear systems under packet drops,” *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2374–2388, 2018.

[22] E. Gilbert, “Capacity of a burst-noise channel,” *Bell Labs Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[23] E. O. Elliott, “Estimates of error rates for codes on burst-noise channels,” *The Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.

[24] C. C. Tan and N. C. Beaulieu, “On first-order markov modeling for the rayleigh fading channel,” *IEEE Transactions on Communications*, vol. 48, no. 12, pp. 2032–2040, Dec 2000.

[25] K. You, M. Fu, and L. Xie, “Necessary and sufficient conditions for stability of kalman filtering with markovian packet losses,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12 465–12 470, 2011.

[26] J. Wu, G. Shi, B. Anderson, and K. Johansson, “Kalman filtering over gilbert–elliott channels: Stability conditions and critical curve,” *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1003–1017, 2018.

[27] K. You and L. Xie, “Minimum data rate for mean square stabilizability of linear systems with markovian packet losses,” *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 772–785, 2011.

[28] Y. Long, S. Liu, L. Xie, and J. Chen, “Stochastic channel allocation for nonlinear systems with markovian packet dropout,” *Journal of Systems Science and Complexity*, vol. 31, no. 1, pp. 22–37, 2018.

[29] K. Okano and H. Ishii, “Stabilization of uncertain systems with finite data rates and markovian packet losses,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 298–307, 2014.

[30] P. Minero, L. Coviello, and M. Franceschetti, “Stabilization over markov feedback channels: the general case,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 349–362, 2013.

[31] P. Sadeghi, R. Kennedy, P. Rapajic, and R. Shams, “Finite-state Markov modeling of fading channels - a survey of principles and applications,” *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 57–80, 2008.

[32] M. Dong, L. Tong, and B. M. Sadler, “Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels,” *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1403–1418, May 2004.

[33] R. S. Varga, *Geršgorin and his circles*. Springer Science & Business Media, 2010, vol. 36, ISBN: 978-3-642-17798-9.

[34] P. D. Lax, *Functional analysis*. Wiley-Intescience, New York, 2002.

[35] M. Gonzâlez-Mendoza, D. A. Sanchez-Salas, and J. L. C. Ruîz, “Wireless Channel Model with Markov Chains Using MATLAB,” in *Modelling the wireless propagation channel: a simulation approach with MATLAB*, V. Katsikis, Ed. InTech, 2012.