Stability analysis of event-triggered anytime control with multiple control laws

Thuy V. Dang, K. V. Ling, *Member, IEEE,*

and Daniel E. Quevedo, *Senior Member, IEEE*

Abstract

To deal with time-varying processor availability and lossy communication channels in embedded and networked control systems, one can employ an event-triggered sequence-based anytime control (E-SAC) algorithm. The main idea of E-SAC is, when computing resources and measurements are available, to compute a sequence of tentative control inputs and store them in a buffer for potential future use. State-dependent Random-time Drift (SRD) approach is often used to analyse and establish stability properties of such E-SAC algorithms. However, using SRD, the analysis quickly becomes combinatoric and hence difficult to extend to more sophisticated E-SAC. In this technical note, we develop a general model and a new stability analysis for E-SAC based on Markov jump systems. Using the new stability analysis, stochastic stability conditions of existing E-SAC are also recovered. In addition, the proposed technique systematically extends to a more sophisticated E-SAC scheme for which, until now, no analytical expression had been obtained.

Thuy V. Dang is with the Interdisciplinary Graduate School (IGS) and with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore. DANG0028@e.ntu.edu.sg

K.V. Ling is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore. EKVLING@ntu.edu.sg

D.E. Quevedo is with the Faculty of Electrical Engineering and Information Technology (EIM-E), Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany. dquevedo@ieee.org

Index Terms

Anytime control, control with time-varying processor availability, networked control systems, event-triggered control algorithms, stochastic stability, Markov jump systems.

I. INTRODUCTION

It is common in the embedded or networked control system that processor availability varies due to varying computational loads and multi-tasking operations. Anytime control algorithm was first proposed in [1] to deal with time-varying processor availability. It uses an idea from the AI community called *anytime algorithm* [2], which is a computational procedure that could provide a valid answer even when it is terminated prematurely.

There are various forms of anytime control algorithms. In [1], different controllers with different floating point operations were designed. Notable later works include [3]–[6]. In [3], a stochastic switching law within a set of pre-designed controllers is proposed. In [4], the main idea is to sequentially calculate the components of the plant input vector. In [5], named as sequence-based anytime control (SAC), a buffer is used to store the tentative future inputs. In [6], a method for co-design of estimator and controller is proposed where the controller requests a time-varying criterion for the estimator. When sensor measurements are transmitted through a communication network, the measurements may be unavailable due to packet dropouts, or network congestion. Among anytime control algorithms, SAC can handle this situation since it has a buffer which serves to provide a control input even when no measurement is received.

Motivated by the idea of using event-triggered control (see e.g. [7]–[11]) as a method to reduce demands on the network and computing processor while guaranteeing satisfactory levels of performance [12], SAC with an event-triggering mechanism (E-SAC) was proposed in [13] and the State-dependent Random-time Drift (SRD) technique of [14] was employed to analyse the stability of E-SAC.

In our conference contribution [15], the E-SAC was extended to a more sophisticated scheme featuring two control laws, a coarse and a fine law. The fine control law could be viewed as an improved version of the coarse control law that requires more processing resources than the coarse control law. Such ideas are wide-spread, e.g., in Model Predictive Control (MPC) [16]–[19], to compute sub-optimal and optimal solutions are two strategies that one can

choose depending on available computation time. Alternatively, fixed-point and floating-point implementations can be used for trading off computation time and quality (accuracy) [20].

It was demonstrated in [15] that with the multi-control law E-SAC schemes, the communication and processing resources could be used more efficiently. Performance in terms of empirical closed-loop cost, channel utilisation and regions for stochastic stability guarantees could be improved, compared with the basic E-SAC.

In [15], the SRD technique was used to analyse the stability of the proposed multi-control law E-SAC schemes. Unfortunately, this requires one to list all possibilities and the corresponding probabilities. For example, in the two-control law schemes, there are two random variables: (1) the number of times each control law is active during (2) the time interval that the buffer becomes empty again. Therefore, it is a combinatoric problem and quickly becomes intractable. As a result, a closed-form expression for stability condition cannot be readily obtained by the SRD approach. It was concluded that extending SRD technique to more sophisticated E-SAC schemes will be difficult.

In the present work, we propose a new approach to investigate the stochastic stability of E-SAC schemes. By modelling E-SAC as a Markov jump system (MJS) [21] with event-triggering, assuming that processor availability and packet dropouts are identical independent distributed (i.i.d) random processes, we systematically establish stochastic stability guarantee of both one- and multi-control law E-SAC schemes. Our proposed approach recovers stability conditions of [13] which is a one-control law E-SAC scheme.

The remainder of this paper is organised as follows: In Section II we provide a review of the basic E-SAC scheme, including the one control law as proposed in [13], and the multi-control law E-SAC schemes as proposed in [15]. In Section III, we propose the Markov jump system with event-triggering (E-MJS) model for stability analysis of the multi-control law schemes. Section IV investigates stochastic stability issues of this E-MJS model. Section V presents the stochastic stability results of E-SAC, derived by the new approach. Section VI documents a simulation study. Section VII draws conclusions.

Notation: $\mathbb{N} = \{1, 2, ...\}$ represents natural numbers, $\mathbb{N}_0 \triangleq \mathbb{N} \cup \{0\}$; $\mathbb{R}$ represents real numbers, $\mathbb{R}_{\geq 0} \triangleq [0, +\infty)$, $\mathbb{R}_{>0} \triangleq (0, +\infty)$; $\{x\}_{\mathcal{K}}$ stands for $\{x_k : k \in \mathcal{K}\}$, $\mathcal{K} \subseteq \mathbb{N}_0$. $\sigma(M)$ denotes the spectral radius of matrix M. $|x| = \sqrt{x^T x}$ denotes the Euclidean norm of vector x. A function $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of $class - \mathcal{K}_\infty$ ($\phi \in \mathcal{K}_\infty$) if it is continuous, zero at the origin, strictly

increasing and unbounded. $\mathbf{Pr}\{\Omega\}$, $\mathbf{Pr}\{\Omega|\Gamma\}$ denote the probability of an event Ω, and the conditional probability of Ω given Γ respectively. The expected value of a random variable ν given Γ is denoted by $\mathbf{E}\{\nu|\Gamma\}$, and $\mathbf{E}\{\nu\}$ represents the unconditional expectation. For a vector y, $y \succ 0$ means that all of its elements are positive. For a matrix $A \in \mathbb{R}^{n \times n}$, $A_{[i:j;l:m]}$ denotes a block matrix contained in A whose elements are taken from row i to j, and column l to m of A. For $z \in \mathbb{R}$, $\lfloor z \rfloor$ denotes the largest integer that is not bigger than z. For $n, k \in \mathbb{N}$, $n \ (\text{mod}) \ k$ means the remainder of n divided by k; $\mathbf{0}_n$ is a zero vector with dimension n. For a vector x, $x(i) \ (i \in \mathbb{N})$ denotes the i-th element of x. For a matrix A, $||A||_\infty$ denotes the infinity norm of A.

II. Review: Event-triggered sequence-based anytime control (E-SAC) schemes

We consider an input-constrained discrete-time non-linear plant model with dynamics given by:

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{U} \subseteq \mathbb{R}$, $k \in \mathbb{N}_0$, see Fig. 1.

Sensor measurements are transmitted to the controller via a delay-free communication link which introduces packet dropouts. The transmission is a threshold-based event-triggering, i.e., the sensor transmits the measurement only when $|x_k| > d$ where the threshold d is a design parameter. The threshold d is fixed once the system runs, and the triggering event is checked periodically at every sampling instant.

The outcome of the transmission is indicated by the random process $\{\gamma\}_{\mathbb{N}_0}$:

$$\gamma_k = \begin{cases} 0 & \text{if the sensor transmits but a dropout occurs}^1 \\ 1 & \text{if } x_k \text{ is received succesfully} \\ 2 & \text{if the sensor did not transmit (i.e. } |x_k| \leq d) \end{cases}$$

which is assumed to be (conditionally) independent and identical (i.i.d) with a successful transmission probability

$$q \triangleq \mathbf{Pr}\{\gamma_k = 1 | |x_k| > d\} = \mathbf{Pr}\{\gamma_k = 1 | \gamma_k \neq 2\} \tag{2}$$

[1]We assume that packet dropout ($\gamma_k = 0$) is distinguishable from no transmission ($\gamma_k = 2$), e.g., through error-detection coding and monitoring of received energy/waveforms in the sensor transmission band (see [22]).
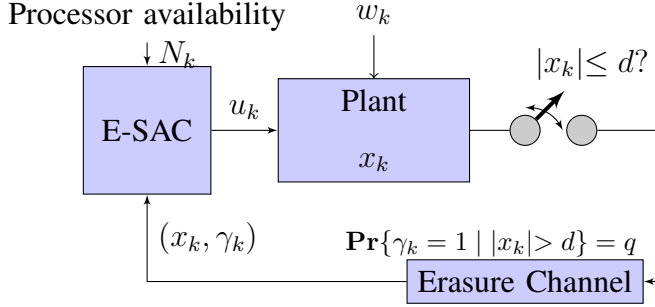
Fig. 1. System Model

Assumption *1 (Processor availability):* The processor is triggered by arrival of valid data. The processor availability for control at different time-instants is (conditionally) i.i.d. Thus, we denote by $N_k \in \{0, 1, 2, \cdots, N_{max}\}$, how many processing units are available at time instants k. The process $\{N\}_{\mathbb{N}_0}$ has conditional probability distribution:

$$\mathbf{Pr}\{N_k = j | \gamma_k = 1\} = p_j, \quad j \in \{0, 1, 2, \cdots, N_{\max}\}, \tag{3}$$

where $p_j \in [0, 1)$ are given.

For other values of γ_k, no plant input is calculated. Thus the processing resources are considered not available regardless, i.e.:

$$\mathbf{Pr}\{N_k = 0 | \gamma_k \in \{0, 2\}\} = 1$$

The next assumption is a combination of some Assumptions from [5] and [15].

Assumption *2 (Coarse and fine control policy):*

The coarse control law $\kappa_1 : \mathbb{R}^n \to \mathbb{U}$ requires 1 processing unit to compute, whereas the fine control law $\kappa_2 : \mathbb{R}^n \to \mathbb{U}$ requires $\eta \in \mathbb{N}$, $\eta \geq 1$, processing units to compute. We also assume that there exist a common Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$; $\varphi_1, \varphi_2 \in \mathcal{K}_\infty$, and $\rho_1 \in \mathbb{R}_{\geq 0}$, $\alpha > 0$, such that $\forall x \in \mathbb{R}^n$

$$\varphi_1(|x|) \leq V(x) \leq \varphi_2(|x|) \tag{4}$$

$$V(f(x, 0)) \leq \alpha V(x) \quad \text{(open-loop bound)} \tag{5}$$

$$V(f(x, \kappa_1(x))) \leq \rho_1 V(x) \quad \text{(closed-loop contraction 1)} \tag{6}$$

$$V(f(x, \kappa_2(x))) \leq \rho_2 V(x) \quad \text{(closed-loop contraction 2)} \tag{7}$$

and the fine control policy κ_2 is better than the coarse policy κ_1 in the sense that $\rho_2 < \rho_1$. $\blacksquare$

A. Event-triggered sequence-based anytime algorithms with one control policy

The baseline algorithm, here denoted by $B1$, amounts to a direct implementation of κ_1 as per

$$u_k = \begin{cases} \kappa_1(x_k) & \text{if } \gamma_k = 1 \text{ and processor is available} \\ 0 & \text{otherwise} \end{cases}$$

Fig. 2 shows the operation of the (one-control law) E-SAC in [13]. We denote this algorithm as $A1$. In $A1$, tentative future inputs using κ_1 are calculated and stored in a local buffer $b_k \in \mathbb{R}^\Lambda$ (Λ: buffer size, the maximum number of control inputs it can store), whenever the computing resources are available ($\gamma_k = 1$ & $N_k > 0$). When $|x_k| > d$ and processing resources are unavailable (due to dropouts or unavailable processor, i.e. $\gamma_k = 0$ or $N_k = 0$), the buffer is shifted, i.e., the first element is thrown away and the rest is kept. If $|x_k| \leq d$ ($\Leftrightarrow \gamma_k = 2$), the buffer is cleared. The matrix representing the shift operation is defined as

$$\Xi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{\Lambda \times \Lambda}$$

The first element in the buffer is used for the current input. We also refer $A1$ to as *one-control law* E-SAC scheme. For more details, see [13].

B. E-SAC with multiple control laws

In control system design, at times one may encounter situations where one would like to switch between different control laws in respond to changing operating conditions. For example, depending on available computational resources, one may switch between a suboptimal or optimal controller, a short or long prediction horizon MPC, or a fixed- or floating-point controller implementation. In our conference contribution [15], E-SAC was extended to schemes featuring two control laws, κ_1 and κ_2, to capture such situations. We refer to κ_1 as the coarse (baseline) control law and to κ_2 as the fine control law. The fine control law κ_2 requires more computational resource to execute than κ_1 as shown in Assumption 2.
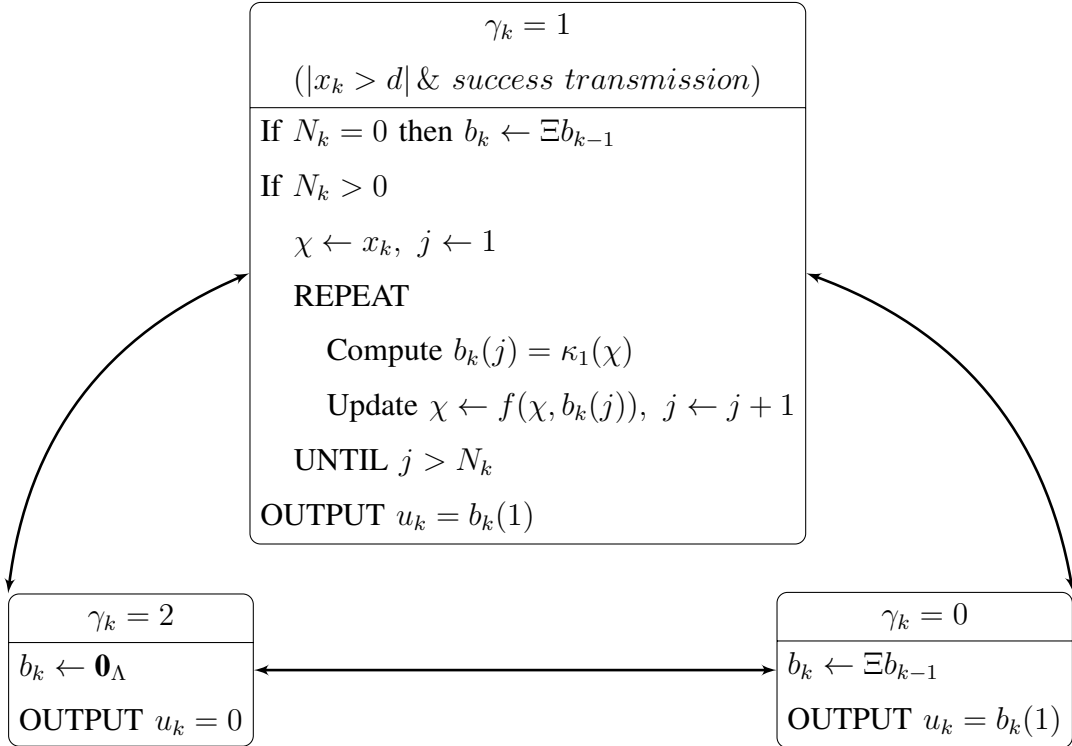
Fig. 2. Operation of $A1$, b_k: buffer content at time instant k.

1) Algorithm $B2$: two-control law E-SAC without buffer: Algorithm $B2$ amounts to a direct implementation of κ_1 and κ_2 without any buffering. The plant input is calculated as

$$u_k = \begin{cases} \kappa_2\left(x_k\right), & \text{if } N_k \geq \eta \\ \kappa_1\left(x_k\right), & \text{if } 0 < N_k < \eta \\ 0, & \text{if } N_k = 0. \end{cases}$$

2) Algorithm $A2$ two-control law E-SAC with buffer: Fig. 3 shows the operation of $A2$. Similar to $A1$, a local buffer with contents $b_k^{\dagger}$ of size $\Lambda \in \mathbb{N}$ is used to store the sequence of tentative future plant inputs calculated by either κ_1 or κ_2 at time k using excess processing resources.

To be more specific, the control policies κ_1 or κ_2 and their tentative future sequence will be executed depending on the values of N_k, as illustrated next.

Given the available processing unit $N_k > 0$ (assumed known in advance), we can write

$$N_k = \tau_k \eta + M_k$$

where $\tau_k \triangleq \lfloor \frac{N_k}{\eta} \rfloor$, and $M_k = N_k \bmod \eta$ (η: see Assumption 2). Firstly, a tentative control
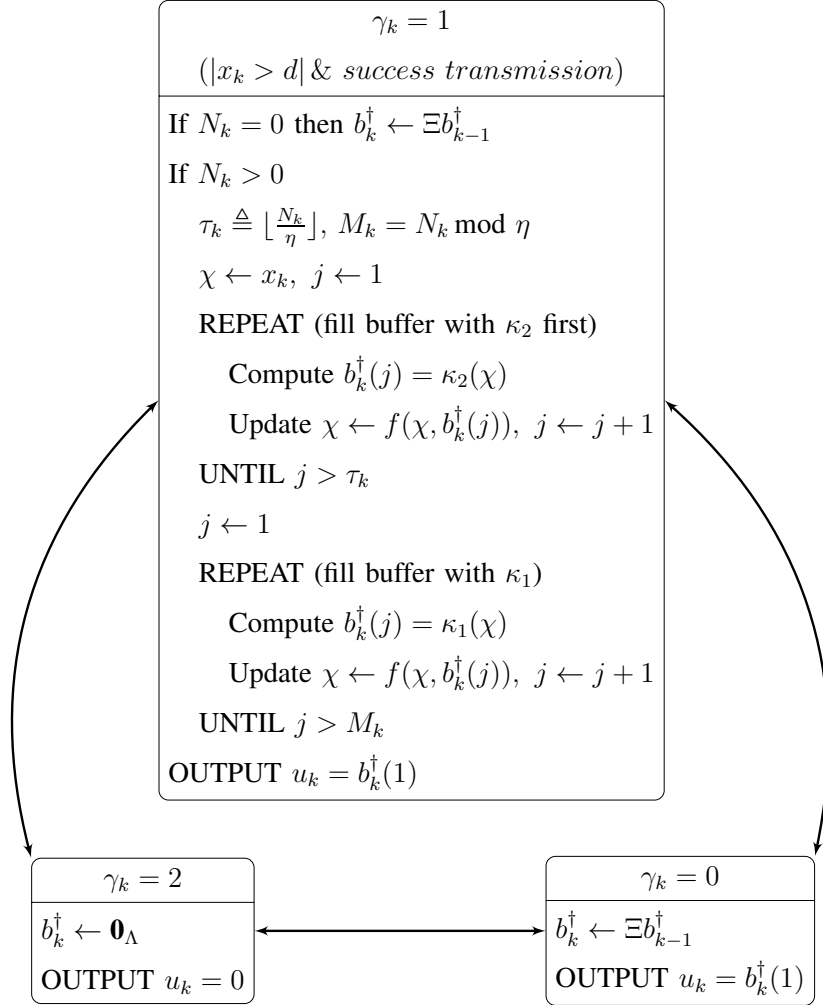
$$\gamma_k = 1$$

$(|x_k > d| \,\&\, success\ transmission)$

If $N_k = 0$ then $b_k^\dagger \leftarrow \Xi b_{k-1}^\dagger$

If $N_k > 0$

$\quad \tau_k \triangleq \lfloor \frac{N_k}{\eta} \rfloor,\ M_k = N_k \bmod \eta$

$\quad \chi \leftarrow x_k,\ j \leftarrow 1$

$\quad$ REPEAT (fill buffer with κ_2 first)

$\qquad$ Compute $b_k^\dagger(j) = \kappa_2(\chi)$

$\qquad$ Update $\chi \leftarrow f(\chi, b_k^\dagger(j)),\ j \leftarrow j+1$

$\quad$ UNTIL $j > \tau_k$

$\quad j \leftarrow 1$

$\quad$ REPEAT (fill buffer with κ_1)

$\qquad$ Compute $b_k^\dagger(j) = \kappa_1(\chi)$

$\qquad$ Update $\chi \leftarrow f(\chi, b_k^\dagger(j)),\ j \leftarrow j+1$

$\quad$ UNTIL $j > M_k$

OUTPUT $u_k = b_k^\dagger(1)$

$$\gamma_k = 2$$

$b_k^\dagger \leftarrow \mathbf{0}_\Lambda$

OUTPUT $u_k = 0$

$$\gamma_k = 0$$

$b_k^\dagger \leftarrow \Xi b_{k-1}^\dagger$

OUTPUT $u_k = b_k^\dagger(1)$

Fig. 3. Operation of $A2$, $b_k^\dagger$: buffer content at time instant k.

sequence is computed by iterating τ_k times the model (1) using κ_2, and then by iterating M_k times the model (1) using κ_1. When $|x_k| > d$ and computing resources are unavailable (due to dropouts or unavailable processor), the buffer is shifted. If $|x_k| \le d$ ($\Leftrightarrow \gamma_k = 2$), the buffer is cleared. The first element in the buffer will be used as the current input. We refer to $A2$ as *two-control law* E-SAC scheme. For ease of exposition, in this work, we only present in details the case of two control laws. The case of multiple control laws can be adapted directly.

Remark *1:* Processing units (in Assumption 1) represent the computational resources (e.g. memory units and given CPU time) available for computing the sequence of predicted control inputs. We assume that the processing time of the control task is significantly smaller than the

sampling time of the plant model. Here it is important to note that the control values written into the buffer at time k only use information about x_k (if available), but not x_{k+1}, or other future states. Merely predictions are used. Therefore, and assuming that processing and transmissions are "infinitely" fast, it is appropriate to use a time-invariant system model as (1). The overall system (including communications, and computations) turns out to be stochastically switching, leading to non-trivial dynamics.

Example 1. Suppose that $N_{\max} = 3, \eta = 2, \Lambda = 3$ and that the processor availability is such that $N_0 = 3, N_1 = 0, N_2 = 2$; the system state is such that $|x_k| > d, \ \forall k = 0, 1, 2$ and there are no dropouts.

If algorithm $A1$ is used with κ_1, then the buffer contents become:

$$\{b_0, b_1, b_2\} = \left\{ \begin{bmatrix} \kappa_1(x_0) \\ \kappa_1(f(x_0, \kappa_1(x_0))) \\ \kappa_1(\hat{x}_0^{[\kappa_1, 2]}) \end{bmatrix}^2, \begin{bmatrix} \kappa_1(f(x_0, \kappa_1(x_0))) \\ \kappa_1(\hat{x}_0^{[\kappa_1, 2]}) \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_1(x_2) \\ \kappa_1(f(x_1, \kappa_1(x_2))) \\ 0 \end{bmatrix}, \right\}$$

which gives the plant inputs $u_0 = \kappa_1(x_0), u_1 = \kappa_1(f(x_0, \kappa_1(x_0))), u_2 = \kappa_1(x_2)$.

If algorithm $A2$ is used, then the buffer contents at times $k \in \{0, 1, 2\}$ become:

$$\{b_0^\dagger, b_1^\dagger, b_2^\dagger\} = \left\{ \begin{bmatrix} \kappa_2(x_0) \\ \kappa_1(f(x_0, \kappa_2(x_0))) \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_1(f(x_0, \kappa_2(x_0))) \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \kappa_2(x_2) \\ 0 \\ 0 \end{bmatrix}, \right\}$$

which gives the plant inputs $u_0 = \kappa_2(x_0), u_1 = \kappa_1(f(x_0, \kappa_2(x_0))), u_2 = \kappa_2(x_2)$.

For the no-buffering schemes, the plant inputs are $u_0 = \kappa_1(x_0), \ u_1 = 0, \ u_2 = \kappa_1(x_2)$ for algorithm $B1$, and $u_0 = \kappa_2(x_0), \ u_1 = 0, \ u_2 = \kappa_2(x_2)$ for algorithm $B2$.

This example suggests that $A2$ outperforms $A1$ since κ_2 gives better control inputs than κ_1. The no-buffering schemes $B1$ and $B2$ cannot provide a control input when the processor is unavailable at time step $k = 1$.

C. State-dependent random-time drift condition approach for stability analysis of E-SAC

In [13], the state-dependent random-time drift (SRD) condition is developed to derive the stochastic stability of the one-control law scheme with buffering, i.e. the scheme $A1$. For deriving the stability condition, it requires one to calculate probability mass function (pmf)

$^2\hat{x}_0^{[\kappa_1, 2]} = f(f(x_0, \kappa_1(x_0)), \kappa_1(f(x_0, \kappa_1(x_0))))$

of random variable Δ_i, which denotes the time interval that the buffer becomes empty again. An analytical formulation of this pmf as well as the closed-form for stability boundary of $A1$ has been established in [13].

In [15], the derivation of stability condition for the two-control law with buffering, i.e. scheme $A2$, follows the same SRD method of [13]. Since there are two control laws in the buffer, the fine control law κ_2 and the coarse control law κ_1, there are not only Δ_i is a random variable, but also the number of times each control law is active, denoted by r_i, is also a random variable. Therefore, it is a combinatoric problem and quickly becomes intractable.

III. MARKOV JUMP SYSTEM WITH EVENT-TRIGGERING MODEL

In this section, we propose a different approach for the stability analysis of E-SAC based on Markov jump system ideas. We shall begin our analysis by developing a stochastic model of the buffer contents at any time k.

Remark 2: If $\eta = 1$ and $\kappa_2 \equiv \kappa_1$, algorithm $A2$ reduces to $A1$. In addition, $A2$ reduces to $B2$ when the buffer size $\Lambda = 1$. Therefore, in the sequel, we only present the stability analysis for algorithm $A2$, since results for $A1$ and $B2$ can be recovered as a special case of $A2$. ■

A. Markov state of the buffer content of $A2$

For two-control law scheme $A2$, we model the content of the buffer via $\theta_k = (F_k; C_k)$ where F_k and C_k indicate the number of κ_2 (fine control law) and κ_1 (baseline or coarse control law) in the buffer at time step k respectively. During periods when $|x_k| > d$, then the transition of θ_k only depends on N_k which is i.i.d. Hence, during periods when $|x_k| > d$, $\{\theta_k\}$ is a Markov chain. The corresponding state space is

$$\mathcal{S}^{(A2)} = \{S_i = (\lfloor \frac{i-1}{\eta} \rfloor; (i-1) \bmod \eta)\}, \tag{8}$$

$$i = 1, 2, ..., N_{\max} + 1$$

and is associated with the conditional $(|x_k| > d)$ transition probability matrix

$$\Pi^{(A2)} = \{\pi_{ij}^{(A2)}\}, \ i, j = 1, 2, ..., N_{\max} + 1.$$

where $\pi_{ij}^{(A2)} = \mathbf{Pr}\{\theta_k = S_j | \theta_{k-1} = S_i, \gamma_k \neq 2, \gamma_{k-1} \neq 2\}$ (see Appendix A for the derivation of the entries of the probability transition matrix). As an example, if $N_{max} = 4$, $\eta = 2$, we have
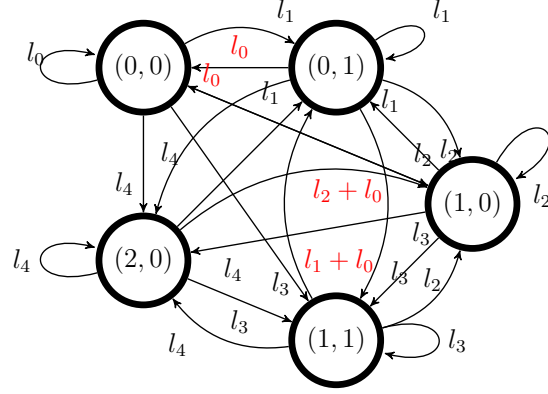
Fig. 4. State transition diagram for buffer contents of $A2$ ($N_{max} = 4$, $\eta = 2$), $|x_k| > d$. During periods when $|x_k| \leq d$, the buffer is always emptied, i.e. there is no change in the buffer content.

the state space of $A2$:

$$\mathcal{S}^{(A2)} = \{S_1 = (0;0),\ S_2 = (0;1),\ S_3 = (1;0),\ S_4 = (1;1), S_5 = (2;0)\}$$

and the state transition diagram shown in Fig. 4.

Remark *3:* In $A2$, for S_i that has the first entry greater than 0, κ_2 will be active, i.e., if θ_k is in the set $\{S_{\eta+1}, S_{\eta+2}, ..., S_{N_{\max}+1}\}$ we have $F_k > 0$, therefore, κ_2 is active. If θ_k is in the set $\{S_2, S_3, ..., S_\eta\}$, then we have $F_k = 0$ and $C_k > 0$, therefore, κ_1 is active. Finally if $\theta_k = S_1 = (0;0)$, then a zero control input is used. ∎

We note that, when $|x_k| > d$, the buffer and the control input change depending on external factors such as processor and measurement availability which are random. We call this as "stochastic mode". On the other hand, when $|x_k| \leq d$, the buffer is cleared and becomes empty. The control, for simplicity and without lost of generality, is set as $u_k = 0$. We call this the "deterministic mode", since the control is fixed.

B. Markov jump system model

For our subsequent analysis, it is convenient to introduce

$$y_k \triangleq [x_k; x_{k-1}; ...; x_{k-\Lambda+1}] \in \mathbb{R}^{n\Lambda} \tag{9}$$

Lemma *1:* $\{y\}_{\mathbb{N}_0}$ is Markovian.

Proof: See Appendix B. ∎

From (1) and (9), for the process $\{y_k\}$, there exists $\psi : \mathbb{R}^{n\Lambda} \times \mathbb{U} \to \mathbb{R}^{n\Lambda}$ such that $y_{k+1} = \psi(y_k, u_k)$.

The control u_k is determined by y_k and the random process θ_k describing the buffer contents (see Remark 3). Then, in general, we have $u_k = \hat{\kappa}_{(\theta_k)}(y_k)$ where $\theta_k \in \{S_i\}_{i=1,2,...}$ (see (8) for S_i) and set of control laws, associated with the contents of the buffer, $\hat{\kappa}_{(S_i)} : \mathbb{R}^{n\Lambda} \to \mathbb{U}$. This leads to

$$y_{k+1} = \psi(y_k, \hat{\kappa}_{(\theta_k)}(y_k)) \tag{10}$$

which is a Markov jump system during intervals when $|x_k| > d$. Here, we use $\hat{\kappa}$ to present the mappings from the domain of y_k (which is different from the domain of x_k) to the domain of u_k [3].

The mapping from a Markov state to a control law for $\{y_k\}$ process is

$$\hat{\kappa}_{(\theta_k=S_0)}(y_k) = 0,$$

$$\hat{\kappa}_{(\theta_k=S_1)}(y_k) = \kappa_1(x_k), \quad \cdots, \quad \hat{\kappa}_{(\theta_k=S_\eta)}(y_k) = \kappa_1(x_k), \tag{11}$$

$$\hat{\kappa}_{(\theta_k=S_{\eta+1})} = \kappa_2(x_k), \quad \cdots, \quad \hat{\kappa}_{(\theta_k=S_{\Lambda+1})}(y_k) = \kappa_2(x_k)$$

Figure. 5 shows an equivalent model of the E-SAC schemes $A2$ via the process $\{y_k\}$. We call this model the event-triggered Markov jump system model (E-MJS). We use $\beta_k \in \{1, 0\}$ to represent the threshold-based triggering event $|x_k| > d$ ($\gamma_k \in \{0, 1\}$) and $|x_k| \le d$ ($\gamma_k = 2$). In this model, the particular schemes such as $A1$ and $A2$ are encoded in the state space $\{S_1, S_2, ...\}$ (for e.g. (8)). The dropouts and processor availability are encoded in the transition probabilities of the state space.

C. General model

We now propose a general mathematical description for the E-MJS model of the E-SAC. Consider a non-linear system $y_{k+1} = \psi(y_k, u_k)$ controlled by two controllers: (1) stochastic controller and (2) deterministic controller. The loop is closed with either the stochastic or the deterministic controller (Fig. 5). When the triggering condition is met, the stochastic controller

---

[3]κ_1, κ_2 are the mapping from the domain of x_k to the domain of u_k.
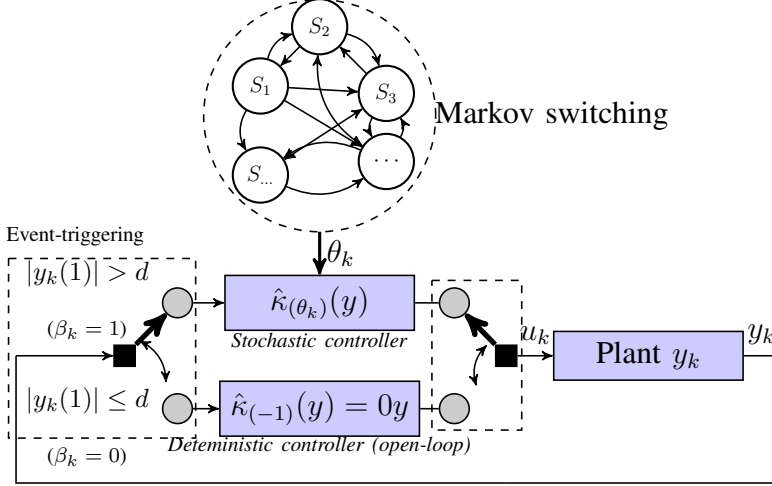
Fig. 5. E-MJS model of E-SAC schemes.

will be deployed. We use $\beta_k \in \{0, 1\}$ to indicate the triggering event at time k:

$$\beta_k = \begin{cases} 0 & \text{use deterministic controller} \\ 1 & \text{use stochastic controller} \end{cases}$$

1) Stochastic controller: Due to the external environment, such as time-varying processing powers or dropouts in the communication channels, the controller switches stochastically within a set of M control laws $\{\hat{\kappa}_{(i)}, \ i = 1, 2, ..., M\}$. In this case, the closed loop system model is

$$y_{k+1} = \psi(y_k, u_k) = \psi(y_k, \hat{\kappa}_{(\theta_k)}(y_k))$$

where $\{\theta_k\}_{\mathbb{N}}$ is a discrete Markov chain with state space

$$\mathcal{S} = \{1, 2, ..., M\} \tag{12}$$

and the (conditional) transition probability matrix

$$\Pi = \{\pi_{ij}\}, \quad i, j = 1, ..., M$$

where $\pi_{ij} = \mathbf{Pr}\{\theta_{k+1} = j | \theta_k = i, \beta_k = 1, \beta_{k+1} = 1\}$

Here, for ease of notation, we use numeric representation for the state space of the Markov chain in (12) instead of using S_i as (8).

2) Deterministic controller: This controller gives a fixed control policy $u_k = \hat{\kappa}_{(-1)}(y_k)$ and in this case, the closed loop is $y_{k+1} = \psi(y_k, \hat{\kappa}_{(-1)}(y_k))$.

For simplicity but without loss of generality, we set $\kappa_{(-1)}(y_k) = 0$.

IV. STOCHASTIC STABILITY OF E-MJS MODEL

In this section, we derive the stochastic stability condition for the proposed E-MJS model. First, we shall make the following assumptions:

Assumption *3:* There exists a non-negative function $\tilde{V} : \mathbb{R}^{n_y} \to \mathbb{R}_{\geq 0}$ (n_y is dimension of y) and coefficients $\sigma_{(i)} \in \mathbb{R}_{\geq 0}$, $i = -1, 1, 2, ..., M$ such that

$$\tilde{V}(\psi(y, \hat{\kappa}_{(i)}(y))) \leq \sigma_{(i)} \tilde{V}(y), \quad \forall i = -1, 1, 2, .., M \tag{13}$$

Assumption *4:* There exists a constant $D \in \mathbb{R}_+$ such that $\tilde{V}(y_k) \leq D$, if the deterministic controller setup is in operation. ∎

Remark *4:* Assumption 3 characterises each control law $\hat{\kappa}_{(i)}$ by a scalar $\sigma_{(i)}$, and bounds the rate of increase of $\tilde{V}(y)$ when a control law $\hat{\kappa}_{(i)}$ is active. In Section V.A we show that Assumptions 3 and 4 are satisfied in the E-SAC schemes $A1$ and $A2$, whenever Assumptions 2 is satisfied. However, Assumptions 2 is potentially conservative as a common Lyapunov function is required. ∎

Let $\tilde{V}_k = \tilde{V}(y_k)$, then we obtain the following stochastic model for $\{\tilde{V}_k\}_{\mathbb{N}_0}$:

$$\begin{cases} \tilde{V}_{k+1} \leq \sigma_{(\theta_k)} \tilde{V}_k, \quad (\underbrace{\theta_k \in \{1, 2, ..., M\}}_{Markov\ jump}) & \text{if } \beta_k = 1 \\ \tilde{V}_{k+1} \leq \sigma_{(-1)} \tilde{V}_k & \text{if } \beta_k = 0 \end{cases}$$

or in a compact form as:

$$\tilde{V}_{k+1} \leq \sigma_{(\theta_k)} \tilde{V}_k, \quad (\theta_k \in \underbrace{\{-1\}}_{\text{deterministic ctrl.}} \cup \underbrace{\{1, 2, ..., M\}}_{\text{Markov jump, stochastic ctrl.}}) \tag{14}$$

Note that we have extended the range of θ_k to include $\theta_k = -1$ (if $\beta_k = 0$, deterministic mode) to have the compact form as shown in (14).

Theorem *1:* If there exist positive real numbers $\nu_{(1)}, ..., \nu_{(M)}$, and $\zeta_{(1)}, ..., \zeta_{(M)}$ such that

$$\sum_{j=1}^{M} \pi_{ij} \sigma_{(i)} \zeta_{(j)} - \zeta_{(i)} = -\nu_{(i)}, \tag{15}$$

for all $i = 1, ..., M$, then $\mathbf{E}\{\tilde{V}_k\} < C_1 \xi^k \mathbf{E}\{\tilde{V}_0\} + C_2 < \infty$ where

$$\xi = 1 - \frac{\min_{1 \leq j \leq M}\{\nu_{(j)}\}}{\max_{1 \leq k \leq M}\{\zeta_{(k)}\}} \in (0, 1)$$

$C_1 = \frac{\zeta_{\max}}{\zeta_{\min}}$, $C_2 = \frac{1}{(\zeta_{\min})(1-\xi)}(\max\{\zeta_{\min}D, |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D\})$ ($\zeta_{\max} = \max\{\zeta_{(i)}\}_{i=1,...,M}$, $\zeta_{\min} = \min\{\zeta_{(i)}\}_{i=1,...,M}$).

Proof: The proof is essentially an adaptation of the general stability result of Markov jump linear systems from [21], [23] specialised for (14), which is a scalar and positive system with

event-triggering. See details in Appendix C. ∎

Theorem 1 provides a general condition for stochastic stability of $\{\tilde{V}_k\}$ in terms of the boundedness property of the expectation. Note that (15) represents a system of linear equations and can be represented as

$$(I - \Phi\Pi)\underline{\zeta} = \underline{\nu} \tag{16}$$

where

$$\Phi \triangleq \mathrm{diag}\{\sigma_{(1)}, \sigma_{(2)}, ..., \sigma_{(M)}\} \tag{17}$$

and $\underline{\zeta} \triangleq (\zeta_{(1)} \quad \zeta_{(2)} \quad ... \quad \zeta_{(M)})^T$, $\underline{\nu} \triangleq (\nu_{(1)} \quad \nu_{(2)} \quad ... \quad \nu_{(M)})^T$.

Then, Theorem 1 can be restated as follows:

Corollary *1:* Define the certification matrix $\mathcal{T} = \Phi\Pi$. If $\mathcal{T}$ is Schur stable, then $\mathbf{E}\{\tilde{V}_k\} < C_1\xi^k\mathbf{E}\{\tilde{V}_0\} + C_2 < \infty$ where $\xi \in (0, 1)$, $C_1, C_2 \in \mathbb{R}_{>0}$.

Proof: See Appendix D.

V. STOCHASTIC STABILITY OF E-SAC SCHEMES

In this section, we derive stochastic stability conditions for the E-SAC schemes by applying the results of Section IV.

A. *Existence and bounds of $\tilde{V}(y_k)$ of E-SAC*

For the process $\{y_k\}$ describing E-SAC (see (9)), we choose the following function

$$\tilde{V} : \mathbb{R}^{n\Lambda} \to \mathbb{R}_{\geq 0}, \quad \tilde{V}(y_k) = V([I \quad 0 \quad 0 \quad ... \quad 0]y_k) = V(x_k)$$

where $V(.)$ is defined as in Assumption 2.

The reason for choosing this $\tilde{V}$ is that it allows us to obtain the bound $\sigma_{(i)}$ in (13). This bound is related to a control law $\hat{\kappa}_{(i)}$ which is associated with a Markov state.

Assumptions 2 and equations (11), lead to the following bounds for $\tilde{V}(y_k)$

$$A2: \begin{cases} \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_1)}(y_k))) \leq \alpha\tilde{V}(y_k) \\ \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_i)}(y_k))) \leq \rho_1\tilde{V}(y_k), \ i = 2, 3, ..., \eta \\ \tilde{V}(\psi(y_k, \hat{\kappa}_{(S_i)}(y_k))) \leq \rho_2\tilde{V}(y_k), \\ \qquad i = \eta + 1, ..., N_{max} + 1 \end{cases} \tag{18}$$

where α, ρ_1 and ρ_2 are defined as in Assumptions 2.

To show that Assumption 4 is also satisfied, we recall that in the deterministic controller setup, $|x_k| \leq d$. Therefore $\tilde{V}(y_k) = V(x_k) \leq \varphi_2(|x_k|) \leq \varphi_2(d) \triangleq D$ (see Assumption 2). ∎

B. Stochastic stability for E-SAC schemes $A1$ and $A2$

We need the following Lemma to establish closed-loop stability, when $A1$ or $A2$ are used.

Lemma *2:* Consider a 2×2 block matrix $H = \begin{bmatrix} X & Y \\ Z & M \end{bmatrix}$, where $X \in \mathbb{R}^{1 \times 1}, Y \in \mathbb{R}^{1 \times m}, Z \in \mathbb{R}^{m \times 1}, M \in \mathbb{R}^{m \times m}$, and M is Schur stable with non-negative entries and $||M||_\infty < 1$ and $trace(M^2) < 1$. Then H is Schur stable if and only if $g(1) > 0$ where $g(\lambda) = (\lambda I_n - X) - Y(\lambda I_m - M)^{-1} Z$.

Proof: See Appendix E. ∎

Closed-loop stability when using algorithm $A2$ is then established as follows:

Corollary *2 (Stochastic stability of $A2$):*

The E-SAC scheme $A2$ yields a stochastically stable loop, in the sense that $V(x_k)$ satisfies the bound condition in Theorem 1, if the certification matrix

$$\mathcal{T}^{(A2)} \triangleq \mathrm{diag}\{\alpha, \underbrace{\rho_1, ..., \rho_1}_{\eta-1 \text{ times}}, \underbrace{\rho_2, ..., \rho_2}_{N_{max}-\eta+1 \text{ times}} \}\Pi^{(A2)}$$

is Schur stable.

Further, if $\rho_1, \rho_2 < 1$ the Schur stability of the certification matrix $\mathcal{T}^{(A2)}$ reduces to

$$\Psi \triangleq l_0\alpha + l_0\alpha\Theta_2^T(I - G_{\rho_1,\rho_2})^{-1}E_2 < 1 \tag{19}$$

where $E_2^T = [\rho_1 \underbrace{0 \cdots 0}_{\eta-2 \text{ elements}} \rho_2 \underbrace{0 \cdots 0}_{N_{max}-\eta+1 \text{ elements}}]$, $\Theta_2^T = [l_1 \ l_2 \ \cdots \ l_{N_{max}}]$, and $G_{\rho_1,\rho_2} = \mathcal{T}^{(A2)}_{2:(N_{max}+1),2:(N_{max}+1)}$ is the lower right block of $\mathcal{T}^{(A2)}$.

Proof: Appendix F. ∎

Remark *5:* Corollary 2 provides an analytical expression for the stability boundary of $A2$ which has not been obtained in the earlier works [15] or [13]. It is reassuring that (19) agrees with numerical results of [15].

Remark *6:* We see that using our approach, the condition that both ρ_1 and ρ_2 are strictly less than 1 is not necessary (which was needed in [15]).

C. Recover stability of $A1$

As aforementioned in Remark 2, $A2$ reduces to $A1$ when $\eta = 1$ and $\kappa_2 \equiv \kappa_1$. The probability transition of buffer content of $A2$ when $\eta = 1$ (i.e. $A1$ this case) is shown in Appendix as the matrix $\Pi^{(A1)}$ in (25). From Corollary 2, we obtain the stability for $A1$:

Corollary *3 (Stochastic stability of $A1$):*

The E-SAC scheme $A1$ yields a stochastically stable loop, in the sense that $V(x_k)$ satisfies the bound condition in Theorem 1, if $\mathcal{T}^{(A1)} \triangleq \text{diag}\{\alpha, \underbrace{\rho_1, ..., \rho_1}_{N_{max}-1 \text{ times}}\}\Pi^{(A1)}$ is Schur stable.

Futher, if $\rho_1 < 1$ the Schur stability condition for $\mathcal{T}^{(A1)}$ is

$$\Omega \triangleq l_0\alpha(1 + \rho_1\Theta^T(I - \rho_1 G)^{-1}E_1) < 1 \tag{20}$$

where $\Pi^{(A1)}$ is as (25), $\Theta^T = [l_1 \; l_2 \; \cdots \; l_{N_{max}}]$, $E_1 = [1 \; 0 \; 0 \; \cdots \; 0]^T \subset \mathbb{R}^{N_{max}\times 1}$, $G = \Pi^{(A1)}_{[2:(N_{max}+1);2:(N_{max}+1)]} \subset \mathbb{R}^{N_{max}\times N_{max}}$ is the lower right block of $\Pi^{(A1)}$ obtained by eliminating the first row and the first column.

Remark *7:* Interestingly, (20) is the same stability condition as already derived with different method in [13].

Remark *8:* The stability of $A1$ is independent of the triggering threshold d as showed in [13]. Similarly, the stability of $A2$ showed in Corollary 2 is also independent of d. The threshold d does however determine the size of the region that the system state converges to. In detail, in Theorem 1, the size of this region is $C_2 = \frac{1}{(\zeta_{\min})(1-\xi)}(\max\{\zeta_{\min}D, |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D\})$. For E-SAC schemes $A1$ and $A2$, $D = \varphi_2(d)$ (as defined in Section V.A.) influences C_2.

VI. NUMERICAL SIMULATION

We assume a plant with dynamics

$$x_{k+1} = -1.34x_k + 0.01sin(x_k) + u_k + w_k, \quad x_0 = 20 \tag{21}$$

where the disturbance w_k is i.i.d., normally distributed with zero mean and unit variance. For the proposed schemes with two control laws, we adopt

$$\kappa_1(x_k) = 1.34x_k - 0.01sin(x_k) + 0.9|x_k|, \tag{22}$$

$$\kappa_2(x_k) = 1.34x_k - 0.01sin(x_k) + c_2|x_k|. \tag{23}$$
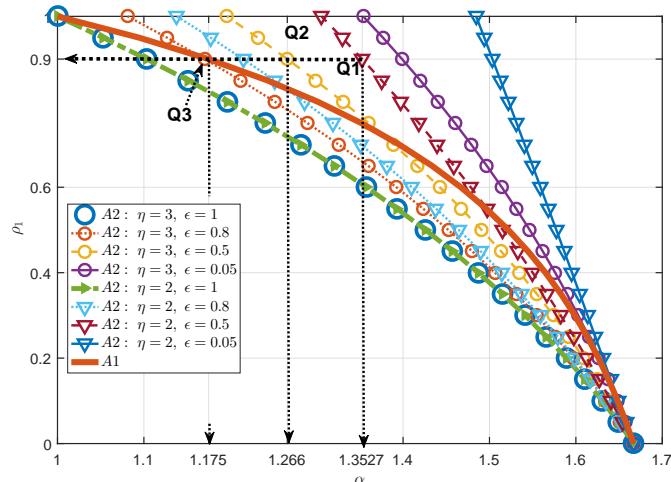
where c_2 is decided later.

Fig. 6. Stability guarantee region for $A2$ with respect to η (processing units needed to compute κ_2) and $\epsilon = \frac{\rho_2}{\rho_1}$. Two-control scheme $A2$ with configuration $Q1$ guarantees to yield a stable system if the open-loop bound satisfying $\alpha < 1.3527$. Two-control scheme $A2$ with configuration $Q2$ guarantees to yield a stable system if $\alpha < 1.266$. One-control scheme $A1$ with configurationn $Q3$ guarantees to yield a stable system if $\alpha < 1.175$. System (21) has open-loop bound $\alpha = 1.35$.
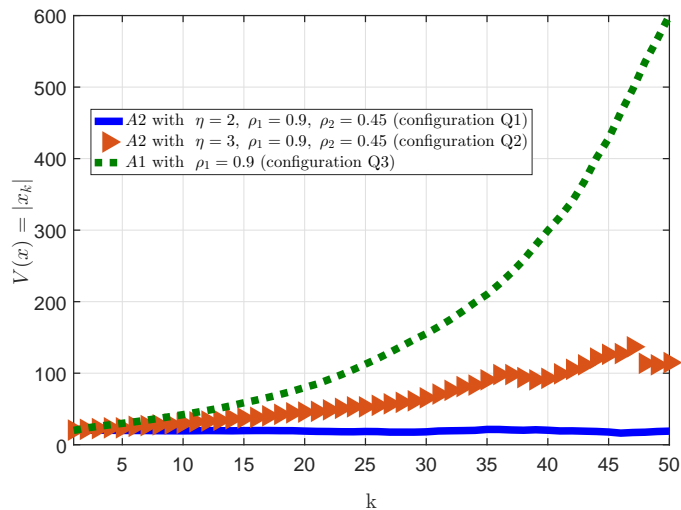

Fig. 7. Averaged value of $V(x) = |x_k|$ over 10^4 random realizations.

By choosing $V(x) = |x|$, we obtain the open-loop bound $\alpha = 1.35$, closed-loop contractions $\rho_1 = 0.9$ in (6) and $\rho_2 = c_2$ in (7).

We also assume that the buffer size $\Lambda = 4$ and that the maximum available processing units are $N_{max} = 4$.

The probability of successful transmission is given by:

$$q = \mathbf{Pr}\{\gamma_k = 1 | \ |x_k| > d\} = 0.5$$

For $N_k \in \{0, 1, 2, \cdots, 4\}$, we assume that the probabilities $p_j = \mathbf{Pr}\{N_k = j | \gamma_k = 1\}$ are equal for each $j \in \{0, 1, 2, \cdots, 4\}$, i.e., $p_0 = p_1 = ... = p_4 = 0.2$.

For ease of presenting the stability region, we introduce a parameter ϵ that represents the ratio between the closed-loop contractions of control laws κ_1 and κ_2 :

$$\epsilon = \frac{\rho_2}{\rho_1}$$

We see that $\epsilon \in [0, 1]$. It can be said that the smaller the ϵ is, the "better" the second control law κ_2 is.

Fig. 6 shows the region for the open-loop bound (α) and closed-loop contractions (ρ_1 & $\rho_2 = \epsilon \rho_1$) that $A2$ and $A1$ guarantee to yield a stochastically stable system, i.e. the region is represented by eq. (19) and (20) given that $\rho_1 \in [0, 1]$. For a specific value of ρ_1 and ϵ, one can figure out the maximum open-loop bound α that is allowed so that the system is guaranteed to be stochastically stable. The region when using $A2$ depends on parameters η and ϵ. Point $Q1 = (1.3525, 0.9)$ is on the curve with label "$A2 : \ \eta = 2, \ \epsilon = 0.5$" (dashed line with triangle). This implies that given two control laws: (1) the coarse control law as eq. (22) ($\rho_1 = 0.9$) and (2) the fine control law as eq. (23) with parameters $\eta = 2$ and $c_2 = \rho_2 = 0.5 * 0.9 = 0.45$, the algorithm $A2$ yields a stochastically stable system **if** the open-loop bound α satisfies $\alpha < 1.3527$. Since system (21) has $\alpha = 1.35 < 1.3527$, $A2$ yields a stochastically stable system with this configuration of two control laws. The blue line in Fig. 7 confirms this, as the averaged value of $V(x_k)$ is bounded.

Similarly, by looking at point $Q2$, it shows that $A2$ with $\rho_1 = 0.9$, $\rho_2 = 0.5*0.9 = 0.45$, $\eta = 3$ only guarantees to yield a stochastically stable system if the open-loop bound α satisfies $\alpha < 1.266$. And by looking at point $Q3$ on the curve labelled as $A1$, it shows that the one control law $A1$ with $\rho_1 = 0.9$ (i.e. using control law as (22)) only guarantees to be able to stochastically stabilise a system if the open-loop bound α satisfies $\alpha < 1.175$. Indeed, the averaged value of $V(x_k)$ can be very large in these two cases, see Fig. 7 the triangle points and dotted line, since the open-loop bound of (21) is $\alpha = 1.35$ bigger than the allowed open-loop bounds of these two configurations.

Remark *9:* Eq. (19) and (20) are sufficient conditions for stochastic stability of $A2$ and $A1$, respectively. Currently, necessary conditions are not available for these schemes.

VII. Conclusion

We propose a general model and a novel stability analysis method for event-triggered sequence-based anytime schemes based on Markov jump systems ideas. The proposed method is, unlike the State-dependent Random-time Drift condition approach, scalable for more sophisticated schemes. It also allows us to obtain an analytical expression for the stability boundary of two-control law schemes, as well as recover the existing stability results of one-control law. Future work and extensions are Markovian processor/sensor availability scenarios, processor scheduling, and the appearance of process noise and model uncertainty.

Appendix

A. Probability transition in $A2$

Firstly, we define

$$l_j \triangleq \mathbf{Pr}\{N_k = j | \gamma_k \neq 2, \gamma_{k-1} \neq 2\} = \mathbf{Pr}\{N_k = j | \gamma_k \neq 2\}$$

$$= \mathbf{Pr}\{N_k = j | \ \gamma_k = 1\}\mathbf{Pr}\{\gamma_k = 1 | \gamma_k \neq 2\} + \mathbf{Pr}\{N_k = j | \ \gamma_k = 0\}\mathbf{Pr}\{\gamma_k = 0 | \gamma_k \neq 2\} \quad (24)$$

satisfy $l_0 = p_0 q + (1 - q)$, $l_j = p_j q$ $(j = 1, 2, ..., N_{\max})$, where p_j and q are defined as in (2) and (3).

We obtain

$$\pi_{ij} = \mathbf{Pr}\{\theta_k = S_j | \theta_{k-1} = S_i, \gamma_k \neq 2, \gamma_{k-1} \neq 2\}$$

$$\pi_{i1} = l_0, \ i \in \{1, 2, \eta + 1\}, \ \pi_{i1} = 0 \ \forall i \neq \{1, 2, \eta + 1\}$$

$$\pi_{ij} = l_{j-1} + l_0, \ \forall \ j = i - \eta, \ i = \eta + 2, ..., N_{max}$$

$$\pi_{ij} = l_j \text{ otherwise}$$

Then, the probability transition matrix is as follows:

$$\Pi^{(A2)} = \begin{bmatrix} l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ l_0 & l_1 & l_2 & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 + l_0 & l_2 & \cdots & \cdots & l_{N_{max}} \\ \cdots & \cdots & \ddots & \cdots & \cdots & l_{N_{max}} \\ 0 & l_1 & \cdots & l_{N_{max}-2} + l_0 & \cdots & l_{N_{max}} \end{bmatrix},$$

when $\eta = 2$ as an example.

When $\eta = 1$ and $\kappa_2 \equiv \kappa_1$, A2 reduces to A1, and the corresponding probability transition matrix is

$$\Pi^{(A1)} = \begin{bmatrix} l_0 & l_1 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ l_0 & l_1 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ 0 & l_1 + l_0 & l_2 & l_3 & \cdots & l_{N_{\max}} \\ 0 & l_1 & l_2 + l_0 & l_3 & \cdots & l_{N_{\max}} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & l_1 & \cdots & \cdots & l_{N_{\max}-1} + l_0 & l_{N_{\max}} \end{bmatrix} \tag{25}$$

B. Proof of Lemma 1

If $N_k > 0$, then u_k is determined by the current state x_k. If the processor is not available, then u_k has been determined by the states which are at most Λ time stages old, or is given by $u_k = 0$. Since the processor availability is independent of the state, the stochastic process $\{y\}_{\mathbb{N}_0}$ is Markovian.

C. Proof of Theorem 1

We define $\zeta_{\max} = \max\{\zeta_{(i)}\}_{i=1,\ldots,M}$, $\zeta_{\min} = \min\{\zeta_{(i)}\}_{i=1,\ldots,M}$, i.e., we have

$$0 < \zeta_{\min} \leq \zeta_{(\theta_k)} \leq \zeta_{\max}, \quad \forall \, \theta_k = 1, \ldots, M. \tag{26}$$

Consider $J_k \triangleq J(y_k, \theta_k) = \zeta_{(\theta_k)} \tilde{V}(y_k) = \zeta_{(\theta_k)} \tilde{V}_k$ where θ_k is as in Eq. (14).

We will next establish the drift condition

$$\mathbf{E}\{J_{k+1} | y_k, \theta_k\} \leq \bar{D} + \xi J(y_k, \theta_k)$$

where $\bar{D}$ is a constant and is derived later (after eq. (32)) in the following.

Since $J(y_k, \theta_k)$ becomes undefined when $\theta_k = -1$ (deterministic mode, $\zeta_{(\theta_k=-1)}$ does not exist), without loss of generality, we take (extended value) $\zeta_{(-1)} = \zeta_{\min}$.

If $\theta_k = -1$, then $\beta_k = 0$, i.e., the deterministic mode is active. We have $\tilde{V}(y_k) \leq D$ (Assumption 4), and we also have $\zeta_{(\theta_{k+1})} \tilde{V}_{k+1} \leq \zeta_{\max} \sigma_{(-1)} \tilde{V}_k$. From $\tilde{V}(y_k) \leq D$, we obtain $\zeta_{\max} \sigma_{(-1)} \tilde{V}_k \leq \xi \zeta_{\min} \tilde{V}_k + |\zeta_{max} \sigma_{(-1)} - \xi \zeta_{\min}| D \leq \xi J(y_k, \theta_k) + |\zeta_{max} \sigma_{(-1)} - \xi \zeta_{\min}| D$. Then it

follows that

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | y_k, \theta_k = 0\} = \mathbf{E}\{\zeta_{(\theta_{k+1})} \tilde{V}_{k+1} | y_k, \theta_k = 0\}$$

$$\leq \xi J(y_k, \theta_k) + |\zeta_{max} \sigma_{(-1)} - \xi \zeta_{\min}| D \tag{27}$$

If $\theta_k = i > 0$, then $\beta_k = 1$. By denoting $\Upsilon_k \triangleq (y_k, \theta_k = i > 0)$, using the law of total expectation we have:

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | y_k, \theta_k = i > 0\} =$$

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 0\} \mathbf{Pr}\{\beta_{k+1} = 0 | \Upsilon_k\} +$$

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 1\} \mathbf{Pr}\{\beta_{k+1} = 1 | \Upsilon_k\}$$

$$\leq \mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 0\} + \mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 1\} \tag{28}$$

For $\beta_{k+1} = 0$, this implies that $\theta_{k+1} = -1$. And by Assumption 4, we have $\tilde{V}_{k+1} \leq D$, then

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 0\} = \mathbf{E}\{\zeta_{(-1)} \tilde{V}_{k+1} | \Upsilon_k, \beta_{k+1} = 0\} \leq \zeta_{\min} D, \qquad (\zeta_{(-1)} = \zeta_{\min}) \tag{29}$$

For the case $\beta_{k+1} = 1$, the stochastic controller is deployed, i.e., $\theta_{k+1} > 0$, we also have $\beta_k = 1$, since $\theta_k > 0$.

We have, using the law of total expectation and using (15),

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | \Upsilon_k, \beta_{k+1} = 1\}$$

$$= \sum_{j=1}^{M} \mathbf{Pr}\{\theta_{k+1} = j | \theta_k = i, y_k, \beta_k = 1, \beta_{k+1} = 1\}.$$

$$\mathbf{E}\{\tilde{V}_{k+1} \zeta_{(\theta_{k+1})} | \theta_{k+1} = j, y_k, \theta_k = i, \beta_k = 1, \beta_{k+1} = 1\}$$

$$\leq \sum_{j=1}^{M} (\pi_{ij})(\sigma_{(i)} \tilde{V}_k \zeta_{(j)}) = \tilde{V}_k \left(\zeta_{(i)} - \nu_{(i)}\right) \quad \text{(using (15))}$$

$$= \tilde{V}_k \zeta_{(i)} \left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) = J(y_k, \theta_k) \left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) \leq \xi J(y_k, \theta_k) \tag{30}$$

since

$$\left(1 - \frac{\nu_{(i)}}{\zeta_{(i)}}\right) \leq 1 - \frac{\min_{1 \leq j \leq M}\{\nu_{(j)}\}}{\max_{1 \leq k \leq M}\{\zeta_{(k)}\}} \triangleq \xi, \ \forall i = 1, .., M$$

and $0 < \xi < 1$ due to $\zeta_{(k)} > \nu_{(k)} > 0, \ \forall k = 1, 2, ..., M$ (from (15)).

Expressions (28)-(30) lead to:

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1}) | y_k, \theta_k > 0\} \leq \zeta_{\min} D + \xi J(y_k, \theta_k) \tag{31}$$

From (27) and (31), we obtain

$$\mathbf{E}\{J(y_{k+1}, \theta_{k+1})|y_k, \theta_k\} \leq \bar{D} + \xi J(y_k, \theta_k)$$

i.e., we have

$$\mathbf{E}\{J(z_{k+1})|z_k\} \leq \bar{D} + \xi J(z_k) \tag{32}$$

where

$$z_k \triangleq (y_k, \theta_k)$$

$$\bar{D} = \max\{\zeta_{\min}D, |\zeta_{\max}\sigma_{(-1)} - \xi\zeta_{\min}|D\}$$

From (32) and using the Markovian property of $\{z_k\}$, [1] we have

$$\mathbf{E}\{J(z_1)|z_0\} \leq \bar{D} + \xi J(z_0) \tag{33}$$

$$\mathbf{E}\{J(z_2)|z_1\} = \mathbf{E}\{J(z_2)|z_1, z_0\} \leq \bar{D} + \xi J(z_1) \tag{34}$$

Taking expectation $\mathbf{E}\{.|z_0\}$ of both sides of (34), using tower property of expectation and using (33) we have

$$\mathbf{E}\{J(z_2)|z_0\} = \mathbf{E}\{\mathbf{E}\{J(z_2)|z_1, z_0\}|z_0\} \leq \bar{D} + \xi\mathbf{E}\{J(z_1)|z_0\} \leq \bar{D} + \xi\bar{D} + \xi^2 J(z_0)$$

By iterating the above procedure, we obtain

$$\mathbf{E}\{J(z_k)|z_0\} \leq \bar{D}\left(\sum_{i=0}^{i=k-1} \xi^i\right) + \xi^k J(z_0) \leq \frac{\bar{D}}{1-\xi} + \xi^k J(z_0)$$

i.e.,

$$\mathbf{E}\{J(y_k, \theta_k)|y_0, \theta_0\} \leq \frac{\bar{D}}{1-\xi} + \xi^k J(y_0, \theta_0)$$

Using the law of total expectation [2], if y_0 is a discrete random variable we have

$$\mathbf{E}\{J(y_k, \theta_k)\} = \sum_{\forall(y_0, \theta_0)} \mathbf{E}\{J(y_k, \theta_k)|y_0, \theta_0\}\mathbf{Pr}\{y_0, \theta_0\}$$

$$\leq \sum_{\forall(y_0, \theta_0)} \left(\frac{1}{1-\xi}\bar{D} + \xi^k J(y_0, \theta_0)\right)\mathbf{Pr}\{x_0, \theta_0\}$$

$$= \frac{1}{1-\xi}\bar{D} + \sum_{\forall(y_0, \theta_0)} \left(\xi^k J(y_0, \theta_0)\right)\mathbf{Pr}\{y_0, \theta_0\}$$

$$= \frac{1}{1-\xi}\bar{D} + \xi^k\mathbf{E}\{J(y_0, \theta_0)\} \tag{35}$$

---

[1] Since $\{y_k\}$ and $\{\theta_k\}$ are Markovian, see Lemma 1 and (14).

[2] Here we assumed θ_0 is known.

If y_0 is a continuous random variable, we have[3]

$$\mathbf{E}\{J(y_k, \theta_k)\} = \int_{y_0} \mathbf{E}\{J(y_k, \theta_k)|y_0, \theta_0\}\mathbf{pdf}\{y_0, \theta_0\}dy_0$$

$$\leq \int_{y_0} \left(\frac{1}{1-\xi}\bar{D} + \xi^k J(y_0, \theta_0) \right) \mathbf{pdf}\{y_0, \theta_0\}dy_0$$

$$= \frac{1}{1-\xi}\bar{D} + \int_{y_0} \left(\xi^k J(y_0, \theta_0) \right) \mathbf{pdf}\{y_0, \theta_0\}dy_0$$

$$= \frac{1}{1-\xi}\bar{D} + \xi^k \mathbf{E}\{J(y_0, \theta_0)\} \tag{36}$$

Since $\zeta_{max}\tilde{V}_k \geq J(y_k, \theta_k) = \zeta_{(\theta_k)}\tilde{V}_k \geq \zeta_{min}\tilde{V}_k$, then $\mathbf{E}\{J(y_k, \theta_k)\} \geq \mathbf{E}\{\tilde{V}_k\}\zeta_{\min}$ and $\mathbf{E}\{J(y_0, \theta_0)\} \leq \zeta_{\max}\mathbf{E}\{\tilde{V}_0\}$. Then, from (35) (or (36)) we have

$$\zeta_{\min}\mathbf{E}\{\tilde{V}_k\} \leq \zeta_{\max}\xi^k\mathbf{E}\{\tilde{V}_0\} + \frac{1}{1-\xi}\bar{D}$$

D. Proof of Corollary 1

Since $\mathcal{T}$ is Schur stable, we have $(I - \mathcal{T})^{-1} = I + \sum_{i=1}^{\infty}(\mathcal{T})^i = I + \bar{\mathcal{T}}$ where $\bar{\mathcal{T}} = \sum_{i=1}^{\infty}(\mathcal{T})^i$. As all entries of $\mathcal{T}$ are non-negative, all entries of $\bar{\mathcal{T}}$ are non-negative. Thus, for any given $\underline{\nu} \succ 0$, the solution of (16) is $\underline{\zeta} = (I - \mathcal{T})^{-1}\underline{\nu} = \underline{\nu} + \bar{\mathcal{T}}\underline{\nu}$ satisfying $\underline{\zeta} \succ 0$. Then, by applying Theorem 1, we obtain the desired bound for $\mathbf{E}\{\tilde{V}_k\}$. ∎

E. Prove of Lemma 2

Consider the characteristic polynomial $P(\lambda) \triangleq \det(\lambda I - H) = \begin{vmatrix} \lambda - X & -Y \\ -Z & \lambda I_m - M \end{vmatrix}$. We denote $eig(\mathcal{M})$ as an eigenvalue of a matrix $\mathcal{M}$.

For λ not inside the unit circle $|\lambda| \geq 1$, we have $eig(\lambda I_m - M) = \lambda - eig(M) \neq 0$ (due to Schur stability of M) hence $\lambda I_m - M$ is invertible. Using the determinant result for a 2-by-2 block matrix, we have

$$P(\lambda) = \det(\lambda I_m - M)\det\left(\lambda - X - Y(\lambda I_m - M)^{-1}Z\right)$$

$$= h(\lambda)g(\lambda), \quad \forall |\lambda| \geq 1 \tag{37}$$

where $h(\lambda) = \det(\lambda I_m - M)$ and $g(\lambda) = \det\left(\lambda - X - Y(\lambda I_m - M)^{-1}Z\right)$.

---

[3]**pdf**: probability density function.

"⇒" We have H is Schur stable, now we need to proof $g(1) > 0$.

Since H is Schur stable, from the Schur-Cohn criteria (see [24], p.27) we have $P(1) > 0$. It follows that $h(1)g(1) > 0$. As M is Schur stable, we have $h(1) = det(I_m - M) > 0$. Hence, $g(1) > 0$.

"⇐" We have $g(1) > 0$, now we need to show that H is Schur stable. Assume that H is not Schur stable. Then there exists λ_0, $|\lambda_0| \geq 1$ such that $P(\lambda_0) = 0$. We then have $\lambda_0 I_m - M$ is invertible and then $P(\lambda_0) = h(\lambda_0)g(\lambda_0) = 0$ where $h(\lambda_0) = det(\lambda_0 I_m - M) \neq 0$. It follows that $g(\lambda_0) = 0$.

(1) If λ_0 is a real number: we can proof that $g(\lambda)$ is a strictly increasing function on $\mathbb{R}$. Then we have $0 = g(\lambda_0) > g(1) > 0$. This is a contradiction.

(2) If λ_0 is a complex number, we have λ_0^*, the conjugate of λ_0, is also an eigenvalue of H. Then λ_0^2 and $(\lambda_0^*)^2$ are eigenvalues of H^2. Then, $trace(H^2) = \sum(eig(H^2)) \geq \lambda_0^2 + (\lambda_0^*)^2 \geq 2$. It can also be shown that $trace(H^2) = X^2 + 2YZ + trace(M^2) < (X + YZ)^2 + trace(M^2) < 2$, since $g(1) > 0 \Rightarrow X + YZ < 1$ and $trace(M^2) < 1$ as in the assumption of Lemma 2. This is also a contradiction.

The two above contradictions establish the result. ∎

F. Proof of Corollary 2

Proof: The certification matrix of $A2$, $\mathcal{T}^{(A2)}$, is calculated as follows. From (18) and (17), we have $\Phi^{(A2)} = \text{diag}\{\alpha, \underbrace{\rho_1, ..., \rho_1}_{\eta-1 \text{ times}}, \underbrace{\rho_2, ..., \rho_2}_{N_{max}-\eta+1 \text{ times}}\}$. We then have $\mathcal{T}^{(A2)} \triangleq \Phi^{(A2)}\Pi^{(A2)}$. Then, the first statement of Corollary 2 is directly proved by Corollary 1.

To prove the second statement of Corollary 2, firstly we have $\mathcal{T}^{(A2)}$ can be written as:

$$\mathcal{T}^{(A2)} = \begin{bmatrix} l_0\alpha & \alpha\Theta^T \\ l_0\rho_1 E_1 & \rho_1 G \end{bmatrix}.$$

It is easy to prove that the (2,2) block of $\mathcal{T}^{(A2)}$, which is G_{ρ_1,ρ_2}, is Schur stable by proving that $||G_{\rho_1,\rho_2}||_\infty < 1$, hence $1 > ||G_{\rho_1,\rho_2}||_\infty \geq \sigma(G_{\rho_1,\rho_2})$ (spectral radius is less than or equal any matrix norm). It can also be shown that $trace(G_{\rho_1,\rho_2}^2) < 1$. Therefore, by applying Lemma 2 we have the Schur condition for $\mathcal{T}^{(A2)}$ is that

$$1 - l_0\alpha - l_0\alpha\Theta_2^T(I - G_{\rho_1,\rho_2})^{-1}E_2 > 0$$

which gives (19). ∎

REFERENCES

[1] R. Bhattacharya and G. J. Balas, "Anytime control algorithm: Model reduction approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 767–776, 2004.

[2] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI magazine*, vol. 17, no. 3, p. 73, 1996.

[3] L. Greco, D. Fontanelli, and A. Bicchi, "Design and stability analysis for anytime control via stochastic scheduling," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 571–585, 2011.

[4] V. Gupta and F. Luo, "On a control algorithm for time-varying processor availability," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 743–748, 2013.

[5] D. E. Quevedo and V. Gupta, "Sequence-based anytime control," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 377–390, Feb 2013.

[6] Y. V. Pant, H. Abbas, K. Mohta, T. X. Nghiem, J. Devietti, and R. Mangharam, "Co-design of anytime computation and robust control," in *2015 IEEE Real-Time Systems Symposium*, Dec 2015, pp. 43–52.

[7] K. J. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesque sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, vol. 2, 2002, pp. 2011–2016.

[8] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[9] D. P. Borgers, V. S. Dolk, and W. P. M. H. Heemels, "Riccati-based design of event-triggered controllers for linear systems with delays," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, June 2017.

[10] Y. Wang, W. X. Zheng, and H. Zhang, "Dynamic event-based control of nonlinear stochastic systems," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, May 2017.

[11] L. Ma, Z. Wang, and H. K. Lam, "Event-triggered mean-square consensus control for time-varying stochastic multi-agent system with sensor saturations," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3524–3531, July 2017.

[12] R. Postoyan, P. Tabuada, D. Nesic, and A. Anta, "A framework for the event-triggered stabilization of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 982–996, 2015.

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

[14] S. Yüksel and S. P. Meyn, "Random-time, state-dependent stochastic drift for Markov chains and application to stochastic stabilization over erasure channels," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 47–59, 2013.

[15] K. Q. Huang, T. V. Dang, K. V. Ling, and D. E. Quevedo, "Event-triggered anytime control with two controllers," in *Proceeding of The 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan*, Dec 2015, pp. 4157–4162.

[16] J. M. Maciejowski, *Predictive control: with constraints*. Prentice Hall, 2002.

[17] K. V. Ling, J. M. Maciejowski, A. Richards, and B. F. Wu, "Multiplexed model predictive control," *Automatica*, vol. 48, no. 2, pp. 396–401, 2012.

[18] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[19] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control - Theory and Algorithms*. Springer-Verlag, London, 2017.

[20] G. Constantinides, A. Kinsman, and N. Nicolici, "Numerical data representations for FPGA-based scientific computing," *IEEE Design & Test of Computers*, vol. 4, no. 28, pp. 8–17, 2011.

[21] Y. Ji and H. J. Chizeck, "Jump linear quadratic Gaussian control: steady-state solution and testable conditions," *Contr. Theory Adv. Tech.*, vol. 6, no. 3, pp. 289–319, 1990.

[22] S. Haykin, *Communication Systems*. John Wiley & Sons, 2008.

[23] Y. Fang and K. Loparo, "Stochastic stability of jump linear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1204–1208, 2002.

[24] J. P. LaSalle, *The stability and control of discrete processes*. Springer Science & Business Media, 1986.