Self-triggered and event-driven control for linear systems with stochastic delays

S. Prakash, E.P. van Horssen, D. Antunes, W.P.M.H. Heemels

***Abstract*— Delays are often present in embedded and networked control loops and represent one of the main sources of performance limitations. In this paper, we propose two aperiodic control strategies to optimize closed-loop performance in the presence of stochastic delays: (i) a self-triggered strategy, in which the deadline to drop data is decided on-line based on the current state; (ii) an event-driven strategy, whereby the control input is updated immediately after the delayed data becomes available, leading in general to faster but time-varying control loops. These schemes are designed and analyzed using a standard LQG framework, which allows for assessing and comparing closed-loop performance. We establish that our self-triggered strategy always achieves a better closed-loop performance than periodic control with an optimal sampling period. Moreover, we provide examples where the event-driven strategy outperforms the self-triggered strategy and examples where the opposite is observed.**

I. Introduction

In many control applications, delays in the control loop are inherent [15]. Prime examples are real-time embedded control systems, where delays result from the computational time needed by the underlying real-time platform to execute the control task [4], [11], and networked control systems where delays result from the time taken for the propagation of signals through the communication network [9], [11]. Another class of examples are applications requiring intensive data-processing algorithms (e.g. vision-based control) introducing processing delays in the control loop. These delays are mostly randomly varying depending on factors such as computational load, network traffic, and quality of sensor data. In several applications, especially those requiring high-performance, these random delays must be taken into account during control design.

In a traditional periodic control setting, these delays can be handled by picking a sufficiently large sampling time, such that even the largest possible delays can be accommodated, but this leads to very conservative designs. Alternatively, a sampling period can be selected according to the time-constants of the system dynamics; when a delay occurs that cannot be accommodated within a sampling period, the corresponding signal-data is 'lost', and typically replaced by the most recently available signal-data. Many control strategies have been proposed to cope with this 'packet-dropping' (see, e.g., [14]). The choice of sampling period obeys a trade-off: a small sampling period increases the packet drop rate deteriorating performance, while a large sampling period leads to a slow loop, also deteriorating performance. Some works in the literature find an optimal sampling period according to given performance specifications [7].

In this paper, we explore the possibility of using aperiodic control to cope with independent and identically distributed (i.i.d.) stochastic delays, and in particular we propose self-triggered and event-driven control strategies. Event-based and self-triggered control have been proposed in recent years as alternatives to standard periodic control. The main motivation behind these alternatives is to reduce the rate at which the control input is updated, either to reduce the communication or the computational burden [8]. Instead, in this paper, we propose the self-triggered and event-driven approaches with the purpose of improving closed-loop performance in the presence of stochastic delays. However, before presenting the aperiodic approaches, we consider a method similar to [7] to obtain the best performance by periodic control with packet drops, which will be used as a benchmark to evaluate the performance of the two aperiodic schemes.

In the self-triggered approach, the deadline to drop data, which in a periodic time-triggered scheme coincides with the sampling period, is decided on-line based on the current state. Here, the challenge is to design a policy for picking such a deadline on-line to optimize closed-loop performance. This problem of designing an on-line control policy as well as a scheduling policy is in general hard [13]. Furthermore, note that, contrarily to the event-driven and periodic schemes, for a particular policy for picking such a deadline on-line, the closed-loop model is non-linear.

In the proposed event-driven strategy, the actuation input is updated immediately after the control update becomes available. Event-driven strategies have been proposed to tackle stochastic delays in other works in the literature, for example, in [12], [10]. However, there, the sensor node is sampled periodically and a new control update is restricted from occurring at least until the next sampling instance of the sensor. Furthermore, in [12] it is assumed that the total delay in one complete loop is not more than one sampling period of the sensor, and in [10] it is limited by a certain

The authors are with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, the Netherlands. E-mail: s.prakash@tue.nl, e.p.v.horssen@tue.nl, d.antunes@tue.nl, m.heemels@tue.nl . This work is supported by the Dutch Technology Foundation (STW) under grant number 12697 Control based on Data-Intensive Sensing, and by the Innovational Research Incentives Scheme under the VICI grant Wireless control systems: A new frontier in automation (No. 11382) awarded by NWO (The Netherlands Organisation for Scientific Research) and STW (Dutch Technology Foundation).

multiple of the sampling period. To illustrate the approaches, we consider a control loop with a delayed actuation signal. In our approach, the sensor is allowed to sample immediately after the last actuation update occurs. Moreover, the delay is allowed to be arbitrarily large but finite. The actuation delay is considered to be significantly larger than the computational delay of the controller.

To model the system with stochastic delays or data-loss, we use a linear time-varying system model with stochastic parameters. The analysis for such systems becomes more intricate than for periodic control schemes, where the models are typically time-invariant. However, using the tools for systems with i.i.d. parameters provided in [6], we can design and analyze such systems. Note that the event-driven policy differs from event-triggered policies in the sense that in event-triggered control, the sampling occurs when the state of the plant deviates from a certain threshold while in event-driven control the sampling occurs at a system event which is not necessarily dependent on the value of its state.

These schemes are designed and analyzed in the standard linear quadratic Gaussian (LQG) framework, where closed-loop performance is evaluated by an average cost. Our main result establishes that the proposed self-triggered strategy outperforms the periodic control strategy with optimal sampling period in this average cost sense. This result highlights the benefits of aperiodic control in the context of control loops with stochastic delays. For certain systems the event-driven strategy can further improve performance with respect to the self-triggered strategy. This is illustrated by a simulation example. However, we also provide an example where the proposed self-triggered strategy outperforms the event-driven strategy, concluding that both aperiodic strategies are viable options for control loops with stochastic delays.

The remainder of the paper is organized as follows. The problem formulation is discussed in Section II. In Section III we discuss the best periodic control strategy to cope with stochastic delays in control loops. In Section IV the proposed aperiodic strategies are discussed and the main results are presented. Section V presents simulation examples comparing the performance of the three proposed methods. Section VI provides concluding remarks.

II. Problem formulation

We consider a continuous-time plant modeled by the following stochastic differential equation

$$dx_c = (A_c x_c + B_c u_c)dt + B_w dw(t),\ x_c(0) = x_0, \qquad (1)$$

where $x_c(t) \in \mathbb{R}^{n_x}$ is the state and $u_c(t) \in R^{n_u}$ is the applied control input at time $t \in \mathbb{R}_{\geq 0}$, and w is an n_w-dimensional Wiener process with incremental covariance $I_{n_w} dt$ [2]. We make the standard assumptions that (A_c, B_c) is controllable and B_c has full rank.

As in the standard linear quadratic Gaussian (LQG) framework, the average quadratic cost

$$J_c = \limsup_{T\to\infty} \frac{1}{T} \int_0^T \mathbb{E}[g_c(x_c(t), u_c(t))]dt, \qquad (2)$$

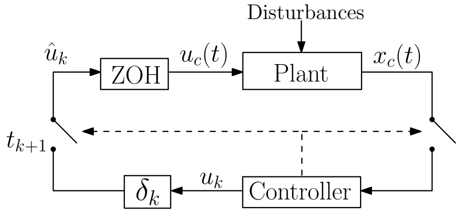

Fig. 1: Control loop with actuation delay δ_k, zero-order hold (ZOH) input holding, and the next control update and sampling instance t_{k+1}.

is chosen as the performance criterion where $g_c(x, u) = x^\intercal Q_c x + u^\intercal R_c u$ with positive definite matrices Q_c and R_c.

The plant is sampled at times t_k, $k \in \mathbb{N}$, with $t_{k+1} > t_k$, for all $k \in \mathbb{N}$. At every sampling instant t_k, we assume that the sensor provides a measurement of the full state $x_c(t_k)$. However, due to, e.g., communication delays, only after a delay $\delta_k \in \mathbb{R}_{\geq 0}$ can the control input be updated using sensor data acquired at time t_k. We assume that between control input updates the control input u_c is held constant and that the delays are independent and identically distributed with known cumulative delay distribution function $F : \mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}$ and thus $\Pr(\delta_k \leq \delta) = F(\delta)$. The corresponding probability density function is denoted by $f(\delta)$, $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The control loop is depicted in Figure 1.

A control strategy must specify not only *how* to update the control input based on the state measurements but also *when* to update the control input. In this paper, we consider three strategies for determining when to update the control input, leading to three different problems for minimizing the performance index (2). These are described next and illustrated in Figure 2.

1) Periodic control: This is the most commonly used strategy whereby the sensor data is acquired at a fixed sampling rate leading to $t_{k+1} - t_k = \tau$, for all k $\in \mathbb{N}$, where $\tau \in \mathbb{R}_{>0}$ and control updates also occur at times t_k, $k \in \mathbb{N}$, with a one-interval delay, i.e, the applied control input $u_c(t_k)$ at time t_k is a function of the state samples and previous control inputs at times t_ℓ, $\ell \in \mathbb{N}_{[0,k)}$. Moreover, when $\delta_k > \tau$ we assume the sensor data acquired at time t_k is dropped and the control input is not updated. Hence, in addition to holding the control input constant between updates, i.e.,

$$u_c(t) = u_c(t_k), \quad t \in [t_k, t_{k+1}), \qquad (3)$$

we also have for $k \in \mathbb{N}$ that

$$u_c(t_{k+1}) = \begin{cases} u_k & \text{if } \delta_k < \tau, \\ u_c(t_{k+1}^-) & \text{otherwise,} \end{cases} \qquad (4)$$

where u_k is the control update designed at time t_k and $u_c(t_{k+1}^-)$ is the limit value of $u_c(t)$ as time t approaches t_{k+1} from below. We assume $u_c(t_1^-) = u_c(t_0)$ is given as part of the initial conditions. The problem that we are interested in this case is first, for fixed τ, to find a control policy

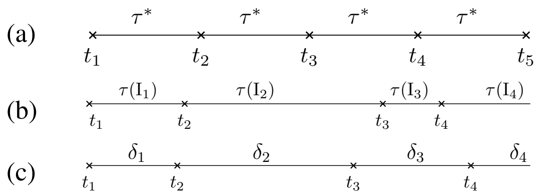

Fig. 2: Illustration of the scheduling schemes of the three proposed methods. (a) Periodic control with data loss (with τ^* being the optimal sampling period) (b) Self-triggered control (c) Event-driven control.

μ specifying $u_k = \mu(\mathrm{I}_k)$ as a function of the information available for control at time t_k being

$$\mathrm{I}_k = \{\big(x_c(t_l), u_c(t_l)\big) \mid l \in \mathbb{N}_{[0,k]}\} \tag{5}$$

to minimize the performance index (2) and then to pick the optimal τ. As we will see in Section III, we can solve this problem optimally, see also [7].

2) Self-triggered control: In this case, instead of picking a constant deadline for dropping data when the delay is too large, which coincides with the sampling period for periodic control, we allow the next deadline to be a function of the information available for control, i.e.,

$$t_{k+1} - t_k = \tau(\mathrm{I}_k), \tag{6}$$

where τ is now a policy to be designed. The equations (3), (4) still hold taking into account that τ is a function of I_k. Note that the closed-loop system is aperiodic in this case. Now, the problem is to design a control policy μ specifying $u_k = \mu(\mathrm{I}_k)$, $k \in \mathbb{N}$ and a policy τ specifying $t_{k+1} = t_k + \tau(\mathrm{I}_k)$, $k \in \mathbb{N}$ as in (6) to minimize the performance index (2). We will provide in Section IV a policy, which leads to a guaranteed better performance than that of the periodic control strategy presented in Section III.

3) Event-driven control: In event-driven control, the control input is updated immediately when the new data becomes available, i.e., $t_{k+1} - t_k = \delta_k$. Furthermore, a new sensor measurement is taken at that same instant. The artificial delays introduced by the periodic and self-triggered strategies when δ_k is less than the deadline are not present. Therefore, the actuation signal may be updated (significantly) faster than in the previous two cases. Although, there are no data drops in this case, we can have longer waiting times when the delays are long while in the previous two cases the packets can be dropped to start computing a new control input. Since the times at which the control updates occur depend on the realization of the stochastic delays and are not assigned by the controller, the problem is simply to find a control policy μ specifying $u_k = \mu(\mathrm{I}_k)$, $k \in \mathbb{N}$, as a function of the information I_k, defined as in (5) available to the controller at time t_k, to minimize the performance index (2). We will be able to solve this control design problem optimally in Section IV, building upon the work in [6].

Remark 1: The above problem formulation assumes that the sensors can be sampled at any time. However, the formulation can also capture the more realistic scenario where the sensors can be sampled at a fast rate, in the sense that the possible sampling period is much smaller than typically delay values. In fact, considering that the actuation updates are delayed to the next sampling instant, the delays take values in a countable set, which can be captured by a piecewise constant cumulative probability distribution.

In order to analyze the three methods, it is convenient to obtain a discrete-time description of the system, which we provide next.

A. Discretization

By discretization of system (1) at times t_k, $k \in \mathbb{N}$, we obtain

$$x_{k+1} = A(\tau_k)x_k + B(\tau_k)\hat{u}_k + w_k, \tag{7}$$

where $x_k := x_c(t_k)$ and $\hat{u}_k := u_c(t_k)$ are the state of the plant and the applied control input, respectively, at t_k, and we define $\tau_k = t_{k+1} - t_k$, $k \in \mathbb{N}$, as the k-th intersampling time (in general varying). For $\tau \in \mathbb{R}_{\geq 0}$, we have

$$A(\tau) = e^{A_c\tau}, \quad B(\tau) = \int_0^\tau e^{A_c s} B_c ds. \tag{8}$$

The disturbance is a sequence of zero-mean independent random vectors $w_k \in \mathbb{R}^{n_w}$, $k \in \mathbb{N}$, with covariance $\mathbb{E}[w_k(w_k)^\intercal] = W(\tau_k)$ with $W(\tau)$, $\tau \in \mathbb{R}_{\geq 0}$, given by

$$W(\tau) = \int_0^\tau e^{A_c s} B_w B_w^\intercal e^{A_c^\intercal s} ds. \tag{9}$$

The average cost can be written as

$$J = \limsup_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{k=0}^{N(T)-1} g(\xi_k, \tau_k)\right], \tag{10}$$

where $N(T)$ is the number of sampling instants up to time T, $\xi_k := [x_k^\intercal \ \hat{u}_k^\intercal]^\intercal$ is an augmented state,

$$g(\xi_k, \tau_k) := \xi_k^\intercal Q(\tau_k)\xi_k, \tag{11}$$

where for $\tau \in \mathbb{R}_{\geq 0}$

$$Q(\tau) = \int_0^\tau e^{\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix}^\intercal s} \begin{bmatrix} Q_c & 0 \\ 0 & R_c \end{bmatrix} e^{\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} s} ds. \tag{12}$$

We will use a Bernoulli random variable γ_k to capture the occurrence of packet-drops in the sense that $\gamma_k = 1$ will denote that the control input u_k has been successfully applied to the system while $\gamma_k = 0$ will denote that u_k has been dropped. As a consequence, (4) can be written as

$$\hat{u}_k = \gamma_{k-1}u_{k-1} + (1 - \gamma_{k-1})\hat{u}_{k-1}, \quad k \in \mathbb{N}, \tag{13}$$

with $\hat{u}_0 = u_c(t_0)$. We combine this equation and (7) and write

$$\xi_{k+1} = \mathcal{A}_{\gamma_k}(\tau_k)\xi_k + \mathcal{B}_{\gamma_k}u_k + \widehat{w}_k, \tag{14}$$

$\widehat{w}_k = [w_k^\intercal \ 0]^\intercal$ and for $\tau \in \mathbb{R}_{\geq 0}$ and $\gamma \in \{0, 1\}$, we have

$$\mathcal{A}_\gamma(\tau) = \begin{bmatrix} A(\tau) & B(\tau) \\ 0 & (1-\gamma)I_{n_u} \end{bmatrix}, \quad \mathcal{B}_\gamma = \begin{bmatrix} 0 \\ \gamma I_{n_u} \end{bmatrix}. \tag{15}$$

This open-loop model is used for the three methods but differs in the manner the variables τ_k and γ_k, $k \in \mathbb{N}$, are determined. In periodic control, τ_k is fixed with the value of the sampling period and γ_k is a Bernoulli random variable; in self-triggered control, τ_k is a variable that is chosen online at t_k based on the information I_k and γ_k is a Bernoulli random variable; in event-driven control, τ_k is a stochastic random variable that takes the value of the control delay δ_k at discrete time k, and $\gamma_k = 1$ for every $k \in \mathbb{N}$.

III. Periodic control with data losses

The optimal periodic control strategy is found in two steps. The optimal control policy is first deduced for a fixed sampling period τ. This is done using the dynamic programming algorithm, which gives the solution in the form of a Riccati equation from which the optimal control gain K_τ and the corresponding optimal cost $J^{p*}(\tau)$ is obtained. Then, the optimal sampling period, denoted by τ^* is deduced from the function $J^{p*} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, i.e.,

$$\tau^* = \arg\min_{\tau > 0} J^{p*}(\tau). \tag{16}$$

The optimal control policy for the system (14) to minimize the cost function (10) can be obtained by applying the standard dynamic programming algorithm [3] and is given by

$$u_k = -K_\tau \xi_k, \tag{17}$$

where K_τ is the optimal control gain given by

$$K_\tau = G_\tau{}^\dagger(\overline{\mathcal{B}^\mathsf{T}_{\gamma_k}(\tau) P_\tau \mathcal{A}_{\gamma_k}(\tau)}). \tag{18}$$

Here, the notation $M^\dagger$ is used to denote the Moore-Penrose pseudo-inverse of the matrix M, and P_τ and G_τ are obtained from the solution to the algebraic Riccati equation

$$\begin{aligned} P_\tau &= \overline{\mathcal{A}_{\gamma_k}(\tau)^\mathsf{T} P_\tau \mathcal{A}_{\gamma_k}(\tau)} + Q(\tau) - K_\tau{}^\mathsf{T} G_\tau K_\tau \\ G_\tau &= \overline{\mathcal{B}^\mathsf{T}_{\gamma_k} P_\tau \mathcal{B}_{\gamma_k}}. \end{aligned} \tag{19}$$

The notation $\overline{X_{\gamma_k} P Y_{\gamma_k}}$ indicates the mean value $\mathbb{E}[X_{\gamma_k} P Y_{\gamma_k}]$, i.e.,

$$\overline{X_{\gamma_k} P Y_{\gamma_k}} = p_\tau(X_1 P Y_1) + (1 - p_\tau)(X_0 P Y_0), \tag{20}$$

where X_{γ_k} and Y_{γ_k} are random matrices that depend on the Bernoulli random variable γ_k and p_τ is the probability of success, which is given by $p_\tau = \Pr[\gamma_k = 1] = F(\tau)$.

The Riccati equation (19) leads to a stabilizing solution (17) for system (15) if the system is ms-stabilizable and $Q(\tau)$ is a positive definite matrix i.e. $Q(\tau) > 0$. Here, the notion of stability considered is mean square stability (mss), c.f. [5, p. 58].

As will be seen in Section V, Figure 3, there is a minimum success rate $\underline{p}$ for p_τ (which in our case translates into a minimum for the sampling period $\underline{\tau}$) below which the system will not have a stabilizing solution, c.f. [14].

It can be shown (see, e.g., [3]) that applying the control policy as in (17), results in the cost

$$J^{p*}(\tau) = \frac{1}{\tau}\,\mathrm{tr}(P_\tau \widehat{W}_\tau), \tag{21}$$

where $\widehat{W}_\tau = \begin{bmatrix} W(\tau) & 0 \\ 0 & 0 \end{bmatrix}$.

IV. Aperiodic control and main results

In this section, we present and discuss the two proposed aperiodic methods. In Subsection IV-A, we introduce a novel self-triggered control strategy guaranteed to outperform the optimal periodic control strategy discussed in Section III. In Subsection IV-B, we provide the optimal control input policy for the event-driven strategy.

A. Self-triggered control

In the self-triggered strategy, the times elapsed between sampling times τ_k are selected based on the information available to the controller, i.e., according to a state-dependent policy, and the control input policy is also to be designed. Note that, the average cost (10) takes the form

$$J^s = \limsup_{L \to \infty} \mathbb{E}\left[\frac{\sum_{k=0}^{L-1} g(\xi_k, \tau_k)}{\sum_{k=0}^{L-1} \tau_k} | \xi_0\right]. \tag{22}$$

We propose the following method to determine these policies for the intervals between sampling times and for the control input, which builds upon the optimal sampling period τ^* and the matrices P_{τ^*}, defined for the periodic control strategy.

$$t_{k+1} - t_k = \tau(\xi_k) \tag{23}$$

$$\tau(\xi_k) = \mathrm{argmin}_{\alpha \in [\underline{\alpha}, \bar{\alpha}]} \xi_k^\mathsf{T} Z(\alpha) \xi_k + \beta(\alpha), \tag{24}$$

$$u_k = -L(\tau(\xi_k))\xi_k, \tag{25}$$

where $\underline{\alpha}$ and $\bar{\alpha}$ are given constants such that $0 < \underline{\alpha} < \tau^* < \bar{\alpha}$ and

$$\beta(\alpha) = \mathrm{tr}(P_{\tau^*} \widehat{W}_\alpha) - \frac{\alpha}{\tau^*}\,\mathrm{tr}(P_{\tau^*} \widehat{W}_{\tau^*}) \tag{26}$$

$$\begin{aligned} Z(\alpha) = {} & \overline{\mathcal{A}_{\gamma_k}(\alpha)^\mathsf{T} P_{\tau^*} \mathcal{A}_{\gamma_k}(\alpha)} + Q(\alpha) \\ & - L(\alpha)^\mathsf{T}(\overline{\mathcal{B}^\mathsf{T}_{\gamma_k}(\alpha) P_{\tau^*} \mathcal{B}_{\gamma_k}(\alpha)})L(\alpha) \end{aligned} \tag{27}$$

$$L(\alpha) = \overline{\mathcal{B}^\mathsf{T}_{\gamma_k}(\alpha) P_{\tau^*} \mathcal{B}_{\gamma_k}(\alpha)}^\dagger (\overline{\mathcal{B}^\mathsf{T}_{\gamma_k}(\alpha) P_{\tau^*} \mathcal{A}_{\gamma_k}(\alpha)}). \tag{28}$$

The rationale behind this policy is to add an optimization for the current sampling interval and current control at each time t_k over a search space that includes the optimal periodic control policy. In fact, in the optimization (24) it is possible to pick $\tau(\xi_k) = \tau^*$ in which case $u_k = K_{\tau^*}\xi_k$, where K_{τ^*} is the gain of the optimal periodic control policy. The following result is the main result of the paper establishing that this policy achieves an average cost less than or equal to the average cost of the optimal control policy.

Theorem 1: Let $J^*_{per} := J^{p*}(\tau^*)$ be the periodic control cost for the optimal control input policy and for the optimal sampling time, and J^{s} be the average cost of policy (23)-(25). Then

$$J^{\mathrm{s}} \leq J^*_{\mathrm{per}}.$$

Proof: The proof is omitted for brevity. ■

Note that the inclusion of τ^* in the set of allowable deadlines does not directly guarantee better performance. It is also required to derive an appropriate one-stage cost argument in (24) whilst the expected cost in the performance index is taken over multiple stages. In fact, the trigger policy (24) is deduced from the proof. The proof builds upon the proof for a result in [1] established in a different context. However, the fact that the intervals between decision times are time-varying and that a continuous parameter (τ_k) is picked instead of a number of discrete options as in [1] makes the proof (more) challenging. In particular, this is reflected in the definition of the self-triggered policy (24). The last term is needed to cope with time-varying intervals between deadline decisions.

B. Event-driven control

To obtain the optimal control input policy for the event-driven case, we consider the discretized system (14) and use the fact that we can write the cost (10) as

$$J^e = \frac{1}{\overline{\delta}} \limsup_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}[g(\xi_k, \delta_k)], \tag{29}$$

where $\overline{\delta} = \mathbb{E}[\delta_k]$. The proof of this fact is omitted for brevity. Since the delays in the loop will result in varying sampling intervals, the randomness in the system due to the delays in the loop results in randomness in the system matrices of the discretized system, and in the matrices of this cost function. As shown in, for example [6], dynamic programming can now be applied on such systems which are characterized by randomly time-varying system parameters. In fact, the optimal solution will be in the form of a stochastic Riccati equation from which a constant control gain is deduced.

The optimal control policy for the system (14) to minimize this cost function is obtained by applying the dynamic programming algorithm [6] and is given by

$$u_k = -K^e \xi_k, \tag{30}$$

where K^e is the optimal control gain given by

$$K^e = G^{e\dagger}(\mathcal{B}_1^\intercal P^e \overline{\mathcal{A}_1(\delta)}), \tag{31}$$

and P^e and G^e are obtained from the solution to the algebraic Riccati equation

$$\begin{aligned} P^e &= \overline{\mathcal{A}_1(\delta)^\intercal P^e \mathcal{A}_1(\delta)} + \overline{Q(\delta)} - K^{e\intercal} G^e K^e, \\ G^e &= \mathcal{B}_1^\intercal P^e \mathcal{B}_1. \end{aligned} \tag{32}$$

In the above Riccati equation, the notation $\overline{U(\delta)PV(\delta)}$ and $\overline{Z(\delta)}$ are used to indicate the mean values $\mathbb{E}[U(\delta)PV(\delta)]$ and $\mathbb{E}[Z(\delta)]$, respectively, given by

$$\begin{aligned} \overline{U(\delta)PV(\delta)} &= \int_0^\infty [U(s)PV(s)]dF(s), \\ \overline{Z(\delta)} &= \int_0^\infty Z(s)dF(s). \end{aligned} \tag{33}$$

The mean cost corresponding to the control law will be given by

$$J^e = \frac{1}{\overline{\delta}} \operatorname{tr}(P^e \widehat{W}^e), \tag{34}$$

where $\widehat{W}^e = \begin{bmatrix} \overline{W}_\delta & 0 \\ 0 & 0 \end{bmatrix}$ with $\overline{W}_\delta$ being the mean covariance of the disturbance w_k in (14), i.e.,

$$\overline{W}_\delta = \int_0^\infty W(s)dF(s),$$

where $W(s)$ is obtained from (9).

	Periodic	Self-triggered	Event-driven
cost	1050.7	993.2(-5.46 %)	768.73(-26.83 %)

TABLE I: Comparison of performance for event-driven and periodic control for the probability distribution of delays δ_k given by $f = f_1$.

V. Example

In this section, we compare the performance of (optimal) periodic control to that of self-triggered and event-driven control on a second-order system given by (1) with

$$A_c = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & -\frac{d}{ml} \end{bmatrix} \quad B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad B_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{35}$$

This model represents a simple inverted pendulum system with force input with the state vector being $x_c = [x \ \dot{x}]^\intercal$, where x and $\dot{x}$ are the angular displacement and the angular velocity, respectively. The system matrices in equation (1) take the form as shown in (35) with the gravitational acceleration $g = 10ms^{-2}$, mass of pendulum $m = 0.25kg$, length $l = 0.5m$ and damping co-efficient $d = 1Nm/rads^{-1}$.

The cost function matrices in (2) are taken as,

$$Q_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad R_c = \begin{bmatrix} 1 \end{bmatrix} \tag{36}$$

The delay δ is set to be randomly varying with Gamma distribution having shape and scale parameters $k = 10$ and $\theta = 0.01$, respectively. We denote the corresponding probability distribution function by $f_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and the cumulative distribution function by $F_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}$.

We start with the evaluation of the periodic control strategy with packet drops. For this, we take τ^{min} and τ^{max} described in Section III as $2.2 \cdot 10^{-16}$ and 0.25, respectively. A set of $n = 500$ values s_i, $i \in \mathbb{N}_{[1,500]}$, are selected in $[\tau^{min}, \ \tau^{max}]$ at regular intervals, for the sampling period τ. Following the procedure described in Section III, it is observed that the system is not ms-stabilizable for sampling periods less than $\underline{\tau} = 0.079$s, below which the packet drop rate is above 72.7%. The plot of (21), i.e., $J^{p*}(\tau)$, against τ is obtained for the remaining values of s_i as shown in Figure 3. The optimal sampling period and the corresponding optimal cost for periodic control are found from the graph as $\tau^* = 0.1327s$ and $J^{p*}(\tau^*) = 1050.7$, respectively, and is validated by Monte-Carlo (MC) simulations.

Next, we consider the self-triggered control method. The average cost, when the self-triggered control method is applied, is found by MC simulations. For this, $\underline{\alpha}$ and $\overline{\alpha}$ as described in Section IV-A are chosen as 0.01 and 0.25, respectively. The average cost for self-triggered control is obtained as $J^s = 993.2$.

Lastly, we consider the event-driven control method. The average cost, when the proposed event-driven control method is applied, is obtained by solving the Riccati equation (32) and using equation (34) as $J^e = 768.7$, and is validated by MC simulations.

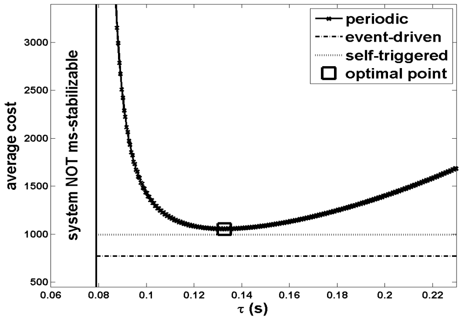

Fig. 3: Comparison of the performance of the three methods on an example system for the probability distribution of delays δ_k given by $f = f_1$ and $F = F_1$.

	Periodic	Event-driven
cost	2053.9	5541.3

TABLE II: Performance of event-driven and periodic control for the probability distribution of delays δ_k given by $f = f_2$.

The average costs resulting from event-driven and self-triggered control methods are compared with that of the optimal periodic control in Figure 3. It is seen that with event-driven control an improvement of 26.83% was achieved against periodic control while an improvement of 5.46% was observed with self-triggered control as shown in Table I. Note that, the performance of event-driven and self-triggered control methods is compared with the best possible performance by periodic control.

The average sampling interval for self-triggered control (as obtained by Monte-Carlo simulations) and for event-driven control (the mean of the delay δ), are 0.092s and 0.1s, i.e., an increase in control rate by 31.4% and 24.6%, respectively, from periodic control. Though the aperiodic methods result in significant increase in control rate, with periodic control, at such high control rates, the performance deteriorates from the optimal, as can be seen in Figure 3.

Although the improvement of event-driven control is significantly larger than that of periodic and self-triggered control in this particular example, it is not necessarily always the case. To show this, we consider the same system with the parameter $d = 5\ Nm/rads^{-1}$ and a different probability distribution function for the delays δ_k given by

$$f_2(\delta) = \begin{cases} 6.25, & \text{if} \quad \delta \in [0.01, 0.09) \cup (0.91, 0.99] \\ 0, & \text{otherwise.} \end{cases}$$

In this setting, the resulting cost for event-driven control $J^e = 5541.3$ is higher than that of periodic control $J^{p*}(\tau^*) = 2053.9$ as also shown in Table II. Note that, for this case, the event-driven performance is worse than that of periodic control and therefore also worse than self-triggered control by Theorem 1. The occurrences of large delays deteriorate the performance of the event-driven scheme. Hence, performance analysis, made possible by the methods presented in this paper, is essential when deciding to apply aperiodic control schemes.

VI. Conclusion

In this paper, we considered control loops where the presence of stochastic delays limit performance of traditional control strategies. To enhance performance, a-priori knowledge of the probability distribution of the delay is used in the controller design. Three control strategies to handle the delays were analyzed. The first method deduces the best possible performance that can be obtained by conventional periodic control with packet drops and an optimal periodic sampling interval. The resulting performance is used as a benchmark for two subsequently proposed aperiodic strategies, namely self-triggered and event-driven control. It is shown, with an example of a simple second order system, that the self-triggered and event-driven control strategies were able to achieve better performance (26.8% and 5.52%, respectively) than periodic control. A second example showed that event-driven control is not necessarily better than periodic control. However, with self-triggered control, performance improvement is always guaranteed, as formally established in our main result.

References

[1] D. Antunes and W.P.M.H. Heemels. Rollout event-triggered control: Beyond periodic control performance. *IEEE Transactions on Automatic Control*, 59:3296–3311, August 2014.

[2] K. J. Åström. *Introduction to stochastic control theory*. Elsevier, 1970.

[3] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 2005.

[4] G. C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Publishing Company, Incorporated, 2011.

[5] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. *Discrete-time Markov jump linear systems*. Springer Science & Business Media, 2006.

[6] W. L. de Koning. Infinite horizon optimal control of linear discrete time systems with stochastic parameters. *Automatica*, 18(4):443–453, July 1982.

[7] Burak Demirel, Zhenhua Zou, Pablo Soldati, and Mikael Johansson. Modular co-design of controllers and transmission schedules in wirelesshart. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5951–5958. IEEE, 2011.

[8] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. Event-triggered and self-triggered control. *Encyclopedia of Systems and Control*, pages 384–391, 2015.

[9] J.P. Hespanha, P. Naghshtabrizi, and Yonggang X. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1), 2007.

[10] Bo Lincoln and Bo Bernhardsson. Optimal control over networks with long random delays. In *Proceedings CD of the 14th International Symposium on Mathematical Theory of Networks and Systems*. Laboratoire de Théorie des Systèmes (LTS), University of Perpignan, 2000.

[11] J. Nilsson. *Real-time control systems with delays*. PhD thesis, Lund institute of Technology Lund, Sweden, 1998.

[12] J. Nilsson, B. Bernhardsson, and B. Wittenmark. Stochastic analysis and control of real-time systems with random time delays. *Automatica*, 34(1), 1998.

[13] S. Reimann, S. Al-Areqi, and S. Liu. Output-based control and scheduling codesign for control systems sharing a limited resource. In *52nd IEEE Conference on Decision and Control*, pages 4042–4047. IEEE, 2013.

[14] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1):163–187, Jan 2007.

[15] B. Wittenmark, J. Nilsson, and M. Torngren. Timing problems in real-time control systems. In *American Control Conference, Proceedings of the 1995*, 1995.