2015 European Control Conference (ECC)
July 15-17, 2015. Linz, Austria

# Frequency-domain analysis of real-time and networked control systems with stochastic delays and data drops

D. Antunes, W. Geelen, W. P. M. H. Heemels

*Abstract*— **We present a novel frequency-domain analysis framework for a closed-loop model capturing a wide range of real-time and networked control systems with stochastic delays and packet drops. Our results allow for inferring the mean and variance of the output response to deterministic inputs, based on a new frequency response plot. We illustrate the usefulness of our results in the context of real-time control systems with input-to-output delays resulting from the use of a shared processor.**

## I. Introduction

Increasingly many control applications rely on shared computational and communication resources. The use of shared resources in real-time and networked control systems reduces costs and increases flexibility. However, it also leads to time-varying effects, such as computational and communication delays and data drops. Taking into account these effects in the control design requires a significantly more elaborate analysis than that for traditional sampled-data systems.

Several works in the literature consider different models (often stochastic) of these effects providing control design methods coping with packet drops [1]–[6], time-varying transmission intervals [7]–[10], and time-varying delays [11]–[17]. Such methods include: (i) optimal control and estimation, using the time-varying Kalman filter [1]; (ii) model-based design [9]; (iii) optimal control [2]; (iv) emulation observer-based designs [18]; (v) adaptive sampling period design [15], [16]; (vi) multi-rate design [14].

However, frequency-domain based tools, widespread in industry in the context of time-invariant control design, are almost absent for coping with varying delays and data drops in the loop. This results from the fact that the models capturing delays and data drops are typically time-varying. Two exceptions in the literature are [19], [20] both exploiting probabilistic descriptions of these time-varying features. In [19] the analysis is carried out for a broad class of real-time and networked control systems, based on the power spectral density of the output response to white noise. A different approach is followed in [20], where the analysis is based on computing the expected value and the variance of the output response to *deterministic signals*. However, [20] considers only models with packet drops.

The purpose of this paper is to show that the analysis in [20], considering packet drops and single-input-single-output systems, can be extended to consider a broad range of real-time and networked control models. In fact, we show that with proper adaptations one can consider general linear systems with independent and identically distributed parameters. We illustrate how this framework allows to capture both stochastic delays in real-time and networked control settings and data losses, arising, e.g., from deadline misses or communication drops. Moreover, we generalize the results in [20] to multiple-input multiple-output control loops.

Our approach builds upon the fact that the maps between the input of the loop and the statistical moments of the state and of the output are *time-invariant*. This allows for plotting the amplitudes of the mean and variance of a scalar output response to sinusoidal input signals as a function of the input frequency. Note that this parallels the classical frequency response (Bode) plot. Moreover, much like the classical analysis, this plot allows for inferring the behavior of the output response, characterized now by its mean and variance, to an arbitrary deterministic input. Through a graphical method, and for a given scalar output, the mean can be exactly computed and the variance can be upper bounded.

We illustrate the usefulness of our results in the analysis of a real-time control system with control delays resulting from the use of a shared processor. In this setting the delay variability may be large and the choice of sampling period is often dictated by the worst-case delays, depending on the scheduling policies and worst-case completion times. With our tools, one can study the influence of reducing the sampling period. This leads to a faster control loop, but results in deadline misses, increasing the uncertainty on the plant's responses, a trade-off which we can capture with our tools.

The paper is organized as follows. Section II introduces the model framework and discusses several settings in real-time and networked control, which can be captured by this model. Section III presents the main results, and Section IV provides an illustrative example. Section V provides concluding remarks. The proofs of the results are omitted for the sake of brevity, but can be obtained via the suitable adaptation of the arguments provided in [20].

The authors are with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, the Netherlands. {D. Antunes,M.Heemels}@tue.nl, w.geelen@student.tue.nl.

This work was supported by STW (Dutch Technology Foundation) and NWO (The Netherlands Organisation for Scientific Research) through the Innovational Research Incentives Scheme under the VICI grant Wireless control systems: A new frontier in automation (11382) and the projects Control based on data-intensive sensing (12697) and Novel unmanned aerial system for industrial inspection: An innovative design, a robust solution (14119).

## II. Modeling Real-time and Networked Control Systems

In this paper we provide a frequency-domain analysis framework for the following model

$$\xi_{k+1} = E(\sigma_k)\xi_k + H(\sigma_k)w_k, \tag{1}$$

where for $k \in \mathbb{Z}$, $\xi_k \in \mathbb{R}^n$ is the state, which may include plant, controller and auxiliary states, $w_k \in \mathbb{R}^{n_w}$ is a deterministic input, which either belongs to the class of signals with bounded energy, i.e., $\sum_{k=-\infty}^{\infty} \|w_k\|^2 < \infty$, or to the class of periodic bounded signals, and $\{\sigma_k\}_{k\in\mathbb{Z}}$ is a sequence of random vectors. Each random vector $\sigma_k$ may belong to a continuous or a discrete space $\mathcal{X}$. A general output of the system is given by

$$z_k = F\xi_k + Qw_k, \tag{2}$$

for given matrices $F$ and $Q$.

We assume that for every $k_1 \in \mathbb{Z}$, $k_2 \in \mathbb{Z}$, $k_1 \neq k_2$, $\sigma_{k_1}$ and $\sigma_{k_2}$ are statistically independent (or orthogonal) random vectors, an assumption which we denote by

$$\sigma_{k_1} \perp \sigma_{k_2}, \quad k_1 \neq k_2. \tag{3}$$

Moreover, we assume that every $\sigma_k$ follows the same distribution $\mu$, i.e., for every $k \in \mathbb{Z}$,

$$\text{Prob}[\sigma_k \in A] = \mu(A), \quad \forall A \subseteq \mathcal{X}.$$

Considering models for which $\sigma_k$, $k \in \mathbb{Z}$, are correlated over time warrants further research (see Section IV). Furthermore, we assume that the (unforced) system (1) with $w_k = 0$, for all $k \in \mathbb{Z}$, is mean square stable, i.e., for every $\xi_0$, $\lim_{k\to\infty} \mathbb{E}[\|\xi_k\|^2] = 0$. Since the unforced system is a special case of a Markov jump linear system, equivalently we assume the following condition (see [21])

$$\mathsf{r}(M) < 1, \tag{4}$$

where $\mathsf{r}$ denotes the spectral radius and

$$M := \mathbb{E}[E(\underline{\sigma}) \otimes E(\underline{\sigma})], \tag{5}$$

for a random vector $\underline{\sigma}$ with probability distribution $\mu$, where $\otimes$ denotes the Kronecker product.

Next, we discuss three scenarios of real-time and networked control systems, which can be captured by (1)-(3), discussing also variants. See [19] for more scenarios captured by (1)-(3).

### A. Input-output (I/0) delays in real-time (RT) control

Consider a standard real-time control loop depicted in Figure 1, in which a linear controller, designed in continuous-time assuming no communication and computational constraints, should be implemented on a shared processor (executing other tasks possibly pertaining to other control loops). In this context, the use of dynamic task scheduling polices and the uncertainty in the duration of each task introduce considerable uncertainty in the start (and completion) time of each task execution. This leads to a significant variability in the delay between the time at which the input of the controller (sensor measurement) is obtained and the time at which the controller output is available to the actuators.

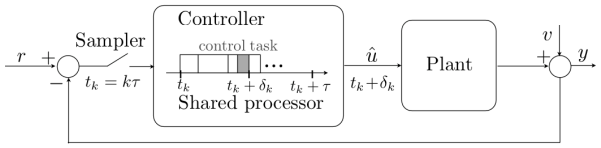



Fig. 1. Real-time control loop with I/0 delays due to task scheduling.

We assume that the sensors are sampled at times

$$t_k = k\tau, \quad k \in \mathbb{Z},$$

where $\tau \in \mathbb{R}_{>0}$ is the sampling period. The input-output delay to implement the control update based on the sensor measurement obtained at time $t_k$ is denoted by $\delta_k$, $k \in \mathbb{Z}$. We assume that the delays are independent and identically distributed with a known probability distribution. We consider two cases, which we discuss next: (I) either the delays are always smaller than the sampling period, i.e., $\delta_k < \tau$, $k \in \mathbb{Z}$, or (II) the delays can be larger than $\tau$ (see Figure 2).

The plant and controller are assumed to be described by

$$\dot{x}(t) = Ax(t) + B\hat{u}(t), \quad y(t) = Cx(t) + v(t), \tag{6}$$

and

$$\begin{pmatrix} \dot{x}^c(t) \\ u(t) \end{pmatrix} = \begin{pmatrix} A_c & B_c \\ C_c & D_c \end{pmatrix} \begin{pmatrix} x^c(t) \\ r(t) - \hat{y}(t) \end{pmatrix}, \tag{7}$$

respectively, where $x(t) \in \mathbb{R}^{n_x}$ and $x^c(t) \in \mathbb{R}^{n_c}$ denote the state of the plant and of the controller, respectively, at time $t \in \mathbb{R}$, $\hat{u}(t) \in \mathbb{R}^{n_u}$ and $y(t) \in \mathbb{R}^{n_y}$ are the input and the output of the plant at time $t$, respectively, $\hat{y}(t) \in \mathbb{R}^{n_y}$ denotes the input of the controller, and $u(t) \in \mathbb{R}^{n_u}$ the output. Moreover, $r(t) \in \mathbb{R}^{n_y}$ is a reference signal and $v(t) \in \mathbb{R}^{n_y}$ is output noise. The input of the controller is hold constant in the time interval $[t_k, t_{k+1})$, in the sense that,

$$r(t) = r(t_k), \quad \hat{y}(t) = y(t_k), \quad t \in [t_k, t_{k+1}). \tag{8}$$

An important special case is when the state is available and (7) is a static state-feedback controller

$$u(t) = K(r(t_k) - x(t_k)), \quad t \in [t_k, t_{k+1})$$

for some gain $K$.

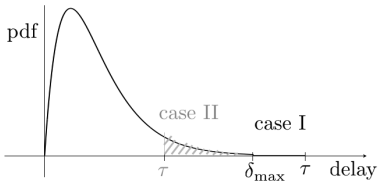



Fig. 2. Illustration of the delay probability density function and connection between sampling period $\tau$ and maximum delay $\delta_{\max}$.

*1) Delay smaller than the sampling period $\tau$:* In this case, the controller task is always executed within a sampling interval. When such task is executed the controller can run or integrate (7) for constant input (8), providing the control update $u_k := u(t_k + \delta_k)$, and the state $x^c(t_{k+1})$. At the actuation side, the control input is hold constant between two actuation update times

$$\hat{u}(t) = u_k, \quad t \in [t_k + \delta_k, t_{k+1} + \delta_{k+1}).$$

For the state-feedback case we can capture this setting with model (1) by considering the state $\xi_k = (x(t_k), u_{k-1})$, the random vector $\sigma_k = \delta_k$, the exogenous input $w_k = (r(t_k), v(t_k))$, $k \in \mathbb{Z}$, and the matrices

$$E(\underline{\delta}) = \begin{pmatrix} e^{A\tau} - B_1(\underline{\delta})K & B_2(\underline{\delta}) \\ K & 0 \end{pmatrix},$$
$$H(\underline{\delta}) = \begin{pmatrix} B_1(\underline{\delta})K \\ K \end{pmatrix} \begin{pmatrix} I & -I \end{pmatrix},$$

where

$$B_1(\underline{\delta}) := \int_0^{\tau-\underline{\delta}} e^{As} ds B, \quad B_2(\underline{\delta}) := e^{A(\tau-\underline{\delta})} \int_0^{\underline{\delta}} e^{As} ds B,$$

for $\underline{\delta} \in [0, \tau)$. Moreover, the outputs $y(t_k)$ or $e(t_k) := y(t_k) - r(t_k)$ can be captured by (2). For example for $y$ we can take $F = \begin{pmatrix} C & 0 \end{pmatrix}$ and $Q = \begin{pmatrix} 0 & I \end{pmatrix}$. We can also model the output feedback case in the framework of (1) by considering $\xi_k = (x(t_k), x^c(t_k), u_{k-1})$, and, for $\underline{\delta} \in [0, \tau)$,

$$E(\underline{\delta}) = \begin{pmatrix} e^{A\tau} - Q_1(\underline{\delta}) & B_1(\underline{\delta})C_c e^{A\underline{\delta}} & B_2(\underline{\delta}) \\ -B_{c,1}(\tau)C & e^{A_c\tau} & 0 \\ -(D_c + C_c B_{c,1}(\underline{\delta}))C & C_c e^{A\underline{\delta}} & 0 \end{pmatrix}, \tag{9}$$

and

$$H(\underline{\delta}) = \begin{pmatrix} B_1(\underline{\delta})(D_c + C_c B_{c,1}(\underline{\delta})) \\ B_{c,1}(\tau) \\ C_c B_{c,1}(\underline{\delta}) + D_c \end{pmatrix} \begin{pmatrix} I & -I \end{pmatrix}, \tag{10}$$

where

$$Q_1(\underline{\delta}) = B_1(\underline{\delta})(D_c + C_c B_{c,1}(\underline{\delta}))C, B_{c,1}(\underline{\delta}) = \int_0^{\underline{\delta}} e^{A_c s} ds B.$$

Note that (3) is satisfied, since the delays $\{\delta_k\}_{k\in\mathbb{Z}}$ are assumed to be independent and identically distributed.

*2) Delay possibly larger than the sampling period $\tau$:* Due to the variability of the task execution times, the I/O delays can be larger than a sampling period $\tau$, i.e. task deadlines may not be met at given execution periods. We assume that if a task has not been executed within the sampling period, it is simply canceled and the processing cycle restarts at time $t_{k+1}$. The next time that the controller task is executed, the controller runs (7) holding constant at its input the last obtained measurement. Then we can change the model for the output feedback case to

$$E(\underline{\delta}) = \begin{cases} (9), \text{ if } \underline{\delta} < \tau, \\ \begin{pmatrix} e^{A\tau} & 0 & B_2(\tau) \\ -B_{c,1}(\tau)C & e^{A_c\tau} & 0 \\ 0 & 0 & I \end{pmatrix} \text{otherwise}, \end{cases} \tag{11}$$

and

$$H(\underline{\delta}) = \begin{cases} (10), \text{ if } \underline{\delta} < \tau, \\ \begin{pmatrix} 0 \\ B_{c,1}(\tau) \\ 0 \end{pmatrix} \begin{pmatrix} I & -I \end{pmatrix} \text{otherwise}. \end{cases} \tag{12}$$

A similar model can be obtained for the state-feedback case.

Note that this problem is also relevant when the controller is specified in discrete time. In fact, consider, for simplicity, a strictly proper discrete-time controller

$$\begin{pmatrix} x_{k+1}^d \\ u_k \end{pmatrix} = \begin{pmatrix} A_d & B_d \\ C_d & 0 \end{pmatrix} \begin{pmatrix} x_k^d \\ r_k - \hat{y}(t_k) \end{pmatrix}. \tag{13}$$

In this case there is no need to cope with delays smaller than the sampling period, since $u_k$ should only be applied at time $t_{k+1}$ and can be computed before $t_{k+1}$ based on $r_k - \hat{y}(t_k)$. Let $\sigma_k = 1$ if $\delta_k \leq \tau$ and $\sigma_k = 0$ if $\delta_k > \tau$ for every $k \in \mathbb{Z}$. Assume that the actuation equals the previous actuation value if the delay is larger than the sampling period, i.e., $u_k = u_{k-1}$ if $\delta_{k-1} > \tau$. Moreover, assume for simplicity that there is no output noise. Then (1) can capture this scenario by considering the state $\xi_k = (x(t_k), x_k^d, u_{k-1})$, $w_k = r_k$,

$$E(\bar{\sigma}) = \begin{cases} \begin{pmatrix} e^{A\tau} & B_2(\tau)C_d & 0 \\ -B_d C & A_d & 0 \\ 0 & C_d & 0 \end{pmatrix}, \text{ if } \bar{\sigma} = 1, \\ \begin{pmatrix} e^{A\tau} & 0 & B_2(\tau) \\ -B_d C & A_d & 0 \\ 0 & 0 & I \end{pmatrix}, \text{ if } \bar{\sigma} = 0, \end{cases} \tag{14}$$

$$H(\bar{\sigma}) = \begin{pmatrix} 0 \\ B_d \\ 0 \end{pmatrix}, \text{ for all } \bar{\sigma} \in \{0, 1\}. \tag{15}$$

We will provide a numerical example for this case in Section IV.

### *B. Networked control systems*

Consider a distributed networked control system in which the plant and the controller communicate over a network. The plant's sensors are sampled at times $t_k = k\tau$, $k \in \mathbb{Z}$, and sent to the controller arriving after a communication delay $\delta_{s-c,k}$. After a computational delay $\delta_{c,k}$, the controller sends the control input of the plant's actuators, which arrives after a delay $\delta_{c-s,k}$. We assume that $\delta_{s-c,k} + \delta_{c-s,k} + \delta_{c,k} < \tau$ for every $k \in \mathbb{Z}_0$. We suppose that a model-based control design is pursued and to assure that the controller knows the actual plant's input $\hat{u}(t)$ we assume that the message from the plant to the controller at time $t_k$ contains not only the sensor measurement obtained at time $t_k$ but also the time at which the actuation was updated in the interval $[t_{k-1}, t_k)$.

Let the plant be described by (6), and consider now for simplicity that the reference signals are absent, i.e., only the impact of noise signals in the networked control system is to be analyzed. Conceptually, the controller mimics the evolution of the plant between consecutive sensor measurements

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B\hat{u}(t), \quad t \in \mathbb{R}_{\geq 0} - \{t_k\}_{k\in\mathbb{Z}}, \tag{16}$$

and, when it receives an output measurement $y(t_k)$, it updates its state by computing

$$\hat{x}(t_k) = \hat{x}(t_k^-) + L(y(t_k) - C\hat{x}(t_k^-)). \tag{17}$$

In practice the controller only needs to execute at times $t_k + \delta_{s-c,k}$ at which it receives $y(t_k)$, computes (17), where $\hat{x}(t_k^-)$ is obtained by running (16), (17) at previous steps, and integrates (16) to obtain the estimated state

$$\hat{x}(t), \quad t \in [t_k + \delta_{s-c,k}, t_k + 2\tau). \tag{18}$$

The control input to be sent to the actuator is computed as

$$u(t) = K(r(t_k) - \hat{x}(t)), \quad t \in [t_k + \delta_{s-c,k}, t_k + 2\tau). \tag{19}$$

The actuators of the plant apply this control input between two messages received from the controller, i.e., in the interval $t \in [t_k + \delta_{s-c,k} + \delta_{c-s,k} + \delta_{c,k}, t_{k+1} + \delta_{s-c,k+1} + \delta_{c-s,k+1} + \delta_{c,k+1})$. Note that in practice the controller sends a discrete-time version of (19) sampled at a very high frequency to the actuators.

We can capture this networked control setting with the model (1) by considering an auxiliary variable $b_k$ which holds the value of the sampled output $b_k = y(t_k)$ in the interval $[t_k, t_{k+1})$ and making $\xi_k := (x(t_k), \hat{x}(t_k), b_k)$, $\sigma_k = (\delta_{s-c,k}, \delta_{c,k}, \delta_{c-s,k})$, $w_k = v(t_k)$,

$$E(\delta_1, \delta_2, \delta_3) = e^{\underline{A}(\tau - \delta_2 - \delta_3)} \underline{J}(\delta_1) e^{\underline{A}\delta_1} \underline{N}$$
$$H(\delta_1, \delta_2, \delta_3) = e^{\underline{A}(\tau - \delta_2 - \delta_3)} \underline{J}(\delta_1) e^{\underline{A}\delta_1} \underline{L}$$

for positive scalars $\delta_1, \delta_2, \delta_3$ and,

$$\underline{A} := \begin{pmatrix} A & -BK & 0 \\ 0 & (A - BK) & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\underline{J}(\delta_1) := \begin{pmatrix} I & 0 & 0 \\ 0 & (I - LC) & L \\ 0 & 0 & I \end{pmatrix},$$
$$\underline{N} := \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ C & 0 & 0 \end{pmatrix}, \underline{L} := \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix},$$

where $\delta_1$ is a positive scalar.

### C. Data losses in an MIMO control loop

Consider a multiple-input multi-output (MIMO) plant in which the data sent from each sensor and actuator link can be lost. The plant and the controller are considered in discrete time (e.g. obtained by discretizing (6), (7)) and described by

$$x_{k+1} = Ax_k + B\tilde{u}_k, \quad y_k = Cx_k, \tag{20}$$

and

$$x_{k+1}^c = A_c x_k^c + B_c(r_k - \tilde{y}_k), \quad u_k = C_c x_k^c + D_c(r_k - \tilde{y}_k), \tag{21}$$

respectively, where $x_k \in \mathbb{R}^{n_x}$ and $x_k^c \in \mathbb{R}^{n_c}$ denote the state of the plant and of the controller at time $k \in \mathbb{Z}$, respectively. Moreover, $\tilde{u}_k \in \mathbb{R}^{n_u}$ and $y_k \in \mathbb{R}^{n_y}$ are the input and the output of the plant at time $k$, respectively. Similarly, $e_k := r_k - \tilde{y}_k$ and $u_k \in \mathbb{R}$ are the input and the output of the controller at time $k$, respectively, where $\tilde{y}_k \in \mathbb{R}^{n_y}$ is the latest received output of the plant and $r_k \in \mathbb{R}^{n_r}$ is the reference signal.

To model the lossy link between the sensor and the controller we use

$$\tilde{y}_k = (1 - \Theta_k)\tilde{y}_{k-1} + \Theta_k y_k,$$

for $k \in \mathbb{Z}$, where $\Theta_k$ is a diagonal matrix with entries $\theta_{i,k}$, $1 \le i \le n_y$, at the diagonal and each $\theta_{i,k}$ equals one if the controller receives the component $i$ of the output of the plant at time $k$ and zero if this data is lost. Similarly, to model the lossy link between the controller and the actuator of the plant we use

$$\tilde{u}_k = (1 - \Omega_k)\tilde{u}_{k-1} + \Omega_k u_k,$$

for $k \in \mathbb{Z}$, where $\Omega_k$ is a diagonal matrix with entries $\rho_{j,k}$, $1 \le j \le n_u$, and each $\rho_{j,k}$ equals one if the actuator receives the component $j$ of the output of the controller at time $k$ and zero otherwise.

Note that there may exist $2^{n_y + n_u}$ data loss possibilities at each time $k$. Let $\sigma_k \in \{1, \ldots, 2^{n_y + n_u}\}$ indicate which of the data loss possibilities occurred at time $k$. We can capture this setting with the model (1), (3) with $\xi_k = (x_k, x_k^c, \hat{y}_k, \hat{u}_k)$, $w_k = r_k$ and

$$E_{\underline{\sigma}} := \begin{pmatrix} A - \underline{\Theta}B\underline{\Omega}D_cC & B\underline{\Omega}C_c & B(I - \underline{\Omega}) & -(I - \underline{\Theta})BD_c \\ -\underline{\Omega}B_c\underline{\Theta}C & A_c & 0 & -B_c(I - \underline{\Theta}) \\ -\underline{\Omega}D_c\underline{\Theta}C & \underline{\Omega}C_c & (I - \underline{\Omega}) & -\underline{\Omega}D_c(I - \underline{\Theta}) \\ \underline{\Theta}C & 0 & 0 & (I - \underline{\Theta}) \end{pmatrix},$$
$$H_{\underline{\sigma}} := \begin{pmatrix} B\underline{\Omega}D_c^\intercal & B_c^\intercal & \underline{\Omega}D_c^\intercal & 0 \end{pmatrix}^\intercal,$$

for $\underline{\sigma} \in \{1, \ldots, 2^{m+p}\}$, and $\underline{\Theta}$, $\underline{\Omega}$ are diagonal matrices whose entries are functions of $\underline{\sigma}$ (this dependence is omitted). These functions specify if the diagonal entries are 0 or 1, depending of which sensors and actuators correspond to data losses in the possibility indicated by $\underline{\sigma}$.

## III. Frequency-domain analysis

Although (1) is time-varying, the expected value and the variance of each component of the output can be determined from the solution to a linear time-invariant system, as stated in the next two propositions.

*Proposition 1:* Consider (1), (2) for a deterministic input $\{w_k\}_{k \in \mathbb{Z}}$ and suppose that (3), (4) hold. Then

$$\mathbb{E}[z_k] = F\beta_k + Qw_k,$$

where the $\beta_k := \mathbb{E}[\xi_k]$, $k \in \mathbb{Z}$, satisfy

$$\beta_{k+1} = \bar{E}\beta_k + \bar{H}w_k, \qquad k \in \mathbb{Z}, \tag{22}$$

and $\bar{E} := \mathbb{E}[E(\underline{\sigma})]$, $\bar{H} := \mathbb{E}[H(\underline{\sigma})]$ for a random vector $\underline{\sigma}$ with probability distribution $\mu$. Moreover, $\bar{E}$ is Schur. □

We now consider that $z_k \in \mathbb{R}$ for every $k \in \mathbb{Z}$ and show how to compute its variance, given by

$$\text{var}(z_k) := \mathbb{E}[(z_k - \mathbb{E}[z_k])^2] = \mathbb{E}[(F\xi_k - \mathbb{E}[F\xi_k])^2]. \tag{23}$$

Note also that

$$\text{var}(z_k) = \mathbb{E}[(F\xi_k)^2] - \mathbb{E}[F\xi_k]^2. \tag{24}$$

For a general vector $z_k$, one can compute the variance of each of its components. The second term on the right hand side of (24) can be computed from Proposition 1. The next result shows how to compute the first term. Let $T$ be the unique matrix such that $Ta \otimes b = b \otimes a$ for $a \in \mathbb{R}^{n_x}$ and $b \in \mathbb{R}^{n_w}$.

*Proposition 2:* Consider (1), a scalar output $z_k \in \mathbb{R}$, $k \in \mathbb{Z}$, given by (2), and a deterministic input $\{w_k\}_{k\in\mathbb{Z}}$. Suppose that (3) holds. Then,

$$\mathbb{E}[(F\xi_k)^2] = G\zeta_k, \tag{25}$$

where $G := F \otimes F$ and $\zeta_k$ satisfies

$$\zeta_{k+1} = M\zeta_k + L\beta_k \otimes w_k + Nw_k \otimes w_k, \tag{26}$$

where $M$ is defined in (5), and for a random vector $\underline{\sigma}$ with probability distribution $\mu$,

$$N := \mathbb{E}[H(\underline{\sigma}) \otimes H(\underline{\sigma})]$$
$$L := \mathbb{E}[E(\underline{\sigma}) \otimes H(\underline{\sigma}) + H(\underline{\sigma}) \otimes E(\underline{\sigma})T].$$

□

Since (22) is linear, using standard arguments for linear time-invariant systems (see, e.g., [22]), we can relate the z-transforms of the input $\hat{w}(z) := \sum_{k=-\infty}^{\infty} w_k z^{-k}$ and the expected value of the output $\hat{z}(z) := \sum_{k=-\infty}^{\infty} \mathbb{E}[z_k] z^{-k}$ by

$$\hat{z}(z) = a(z)\hat{w}(z), \tag{27}$$

where $a(z) := F(zI - \bar{E})^{-1}\bar{H} + Q$, for $z$ in the intersection of the regions of convergence of $\hat{r}(z)$ and $a(z)$. However, (24), (25), (26) depend non-linearly on the input $w_k$, $k \in \mathbb{Z}$. Yet, as we show next, we can compute (26), and (24), when $w$ is a scalar sinusoidal input,

$$w_k = \Im\{ve^{j\omega_c k}\} = |v|\sin(\omega_c k + \psi_v), \tag{28}$$

where $v = |v|e^{j\psi_v} \in \mathbb{C}$ is the complex amplitude, $\omega_c \in (-\pi, \pi]$ is the frequency, and $\Im$ denotes the imaginary part.

*Proposition 3:* Consider (1), a scalar output $z_k \in \mathbb{R}$, $k \in \mathbb{Z}$, given by (2), and the scalar input (28). Suppose that (3), (4) hold. Then,

$$\text{var}(z_k) = b(e^{j\omega_c})|v|^2 - \Re\{c(e^{j\omega_c})v^2 e^{2\omega_c k}\}, \tag{29}$$

for every $k \in \mathbb{Z}$, where, for $z \in \mathbb{C}$,

$$b(z) := \frac{1}{2}\Re\{G(I - M)^{-1}(N + L(zI - \bar{E})^{-1}\bar{H})\} - \frac{|a(z)|^2}{2},$$
$$c(z) := \frac{1}{2}G(z^2 I - M)^{-1}(N + L(zI - \bar{E})^{-1}\bar{H}) - \frac{a(z)^2}{2}.$$

Moreover, $b(1) = 0$ and $c(1) = 0$, and thus $\text{var}(z_k) = 0$ for every $k \in \mathbb{Z}$ when $\omega_c = 0$.

□

Based on Proposition 3 and (27), we define a new frequency domain plot, in which $a(e^{j\omega})$, $b(e^{j\omega})$, $c(e^{j\omega})$ are plotted as a function of frequency $\omega \in [0, \pi)$. Based on this plot, as we show next, we can give a bound for the variance (24) to an arbitrary input signal characterized by its Fourier transform $\hat{w}(e^{j\omega})$, $\omega \in (-\pi, \pi]$. This is main result of our novel approach. Note that $\hat{w}(e^{j\omega})$ exists in $\omega \in (-\pi, \pi]$ for signals with bounded energy [23]. For periodic signals with period $T$, $\hat{w}(e^{j\omega})$ is a generalized function given by

$$\hat{w}(e^{j\omega}) = \frac{2\pi}{T}\sum_{\ell=0}^{T-1} v_\ell \delta(t - \frac{2\pi k}{T}), \tag{30}$$

where the Fourier coefficients $v_\ell$ are given by

$$v_\ell := \sum_{k=0}^{T-1} w_k e^{-j\frac{2\pi\ell}{T}k}, \tag{31}$$

and $\delta$ denotes the Dirac function [23].

*Theorem 4:* Consider (1), a scalar output $z_k \in \mathbb{R}$, $k \in \mathbb{Z}$, given by (2), and a scalar input $w_k \in \mathbb{R}$, $k \in \mathbb{Z}$ with Fourier transform $\hat{w}(e^{j\omega})$. Then, for $w$ with bounded energy it holds that

$$\text{var}(z_k) \le \frac{2}{\pi}\int_0^\pi \big(|b(e^{j\omega})| + |c(e^{j\omega})|\big)|\hat{w}(e^{j\omega})|^2 d\omega, \tag{32}$$

for every $k \in \mathbb{Z}$. Moreover, for $T-$ periodic $w$ it holds that

$$\text{var}(z_k) \le \frac{4}{T}\sum_{\ell=1}^{\lfloor \frac{T}{2} \rfloor} \big(|b(e^{j\omega_\ell})| + |c(e^{j\omega_\ell})|\big)|v_\ell|^2, \tag{33}$$

for every $k \in \mathbb{Z}$, where $\omega_\ell := \frac{2\pi\ell}{T}$, the $v_\ell$ are described in (31), and $\lfloor \frac{T}{2} \rfloor$ denotes the floor of $\frac{T}{2}$.

□

As for LTI systems, in order to obtain the expected value of the output, it suffices to multiply $a(e^{j\omega})$ and $\hat{w}(e^{j\omega})$ and obtain the expected value of the output by inverting the Fourier transform $a(e^{j\omega})\hat{w}(e^{j\omega})$, $\omega \in (-\pi, \pi]$. To obtain a bound for the variance it suffices to multiply $|\hat{w}(e^{j\omega})|^2$ by $|b(e^{j\omega})| + |c(e^{j\omega})|$, for $\omega \in [0, \pi]$, and computing (32). Graphically (see Figure 6), the computation of (32) amounts to plotting $(|b(e^{j\omega})| + |c(e^{j\omega})|)|\hat{w}(j\omega)|^2$, for $\omega \in [0, \pi]$, and computing the average over frequency multiplied by a factor 2.

## IV. Example

We consider the scenario described in Section II-A.2. The plant (6) is considered to be a double integrator described by the transfer function $p_1(s) := \frac{1}{s^2}$ and the controller is described by the transfer function $c_1(s) := \frac{10(s+1)}{s+5}\frac{20}{s+20}$. The plant and the controller are discretized using a sampling period $\tau$ with the zero-order hold method [22]. Note that this results in a strictly proper linear time-invariant controller of the form (13). Let $p_{d,1}(z)$ and $c_{d,1}(z)$ denote the resulting transfer functions in discrete time.

The delay distribution is assumed to be a truncated gamma distribution depicted in Figure 3(a). and described by the probability density $\text{Prob}[\delta \in [a, b)] = \int_a^b f(s)ds$, with

$$f(s) = \begin{cases} \alpha_1 s^2 e^{-\lambda s}, \text{ if } s < 0.1 \\ 0, \text{ if } s \ge 0.1 \end{cases}$$

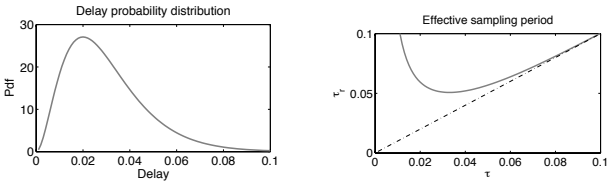
for $\lambda = 100$ and a normalization constant $\alpha_1$, where the maximum delay is assumed to be $0.1$. A conservative design would then pick the sampling period $\tau = 0.1$, which would avoid deadline misses, resulting in a time-invariant system and thus could be analyzed by classical control methods for LTI systems, and in particular by frequency domain analysis tools. However, this leads to a significant performance degradation compared to the ideal continuous-time case. To illustrate this, we plot the step responses of the closed-loop with the ideal continuous-time controller and the discretized one in Figure 4(a), as well as the sensitivity functions $\frac{1}{1+p_{d,1}(z)c_{d,1}(z)}$ and $\frac{1}{1+p_1(s)c_1(s)}$ in Figure 5(b).

Using our tools, we investigate the possibilities of reducing the sampling period $\tau$. By properly choosing $\tau$, this will result in a closed-loop system with an increased control update rate, but time-varying, since deadline misses occur with probability $q_{\text{miss}}(\tau) = \int_{\tau}^{0.1} f(s)ds$. The average control update period is then given by $\frac{\tau}{1-q_{\text{miss}(\tau)}}$. This function is plotted in Figure 3(b), achieving its minimum for $\tau = 0.033$. Note that this plot should serve as a mere indicator on how to choose $\tau$ and does not directly translate into closed-loop properties. For instance, if the delay distribution were uniform, the effective control update rate $\frac{\tau}{1-q_{\text{miss}(\tau)}}$ would be constant as a function of $\tau \in [0, 0.1)]$, although the variability of the response would increase when $\tau$ decreases. However, it can serve as a guideline. For example, it is intuitively clear that there should be no advantage on reducing $\tau$ to values smaller than $0.033$, since this reduces the average sampling period while still creating variability in the response.

The matrices from the model (1) can be obtained from (14), (15). We start by analyzing the system in the time domain. Figure 4(b)-(d) plots realizations of the output response to a step, and the expected value of the output for several values of $\tau$. It also plots two signals obtained by adding and subtracting the standard deviation to the expected value. The mean is obtained by running (22) for a step input. The standard deviation is obtained by running (26) for a step input and taking the square root of (29). Note that indeed there is no advantage in making $\tau$ very small (e.g. $\tau = 0.01$). The choice of $\tau$ is dictated by the following trade-off: decreasing $\tau$ may lead to faster average control update rates and thus better control performance but it also increases the uncertainty about the system which is visible in the two plots of Figure 4 corresponding to $\tau = 0.066$ and $\tau = 0.033$.

The time-domain analysis is insightful but limited, as it allows to reason only in terms of a single input (in this case a step). One of the advantages of the frequency domain analysis is that is allows to reason in terms of any input.

Figures 5(b)-(c) plot the frequency response of the mean and the variance of the error $e = y - r$ proposed in Section II-C, when $\tau = 0.033$ and when $\tau = 0.066$ and Figures 5(d) provides a closer look to a frequency range of interest of Figures 5(a)-(c). As we can see, for $\tau = 0.033$ the sensitivity plot associated with $a(e^{j\omega})$ improves but




(a) Pdf (b) effective sampling period

Fig. 3. Delay probability density and average (effective) control update period as a function of sampling period $\tau$

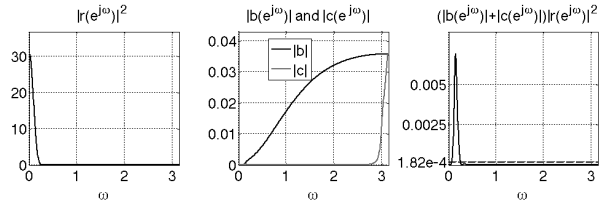



Fig. 6. Computation of the bound for the variance.

the variance increases (see plots of $b(e^{j\omega})$, $c(e^{j\omega})$) when compared to $\tau = 0.066$.

Using these plots and the tools provided in Section IV one can infer how the expected value of the response is and bound the variance for any control input with a given frequency content. For instance, for the following input

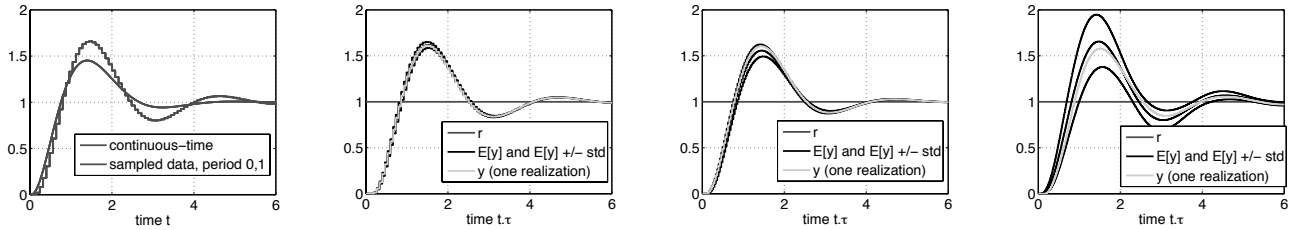
$$r_t = \begin{cases} \gamma \frac{1}{2}(1 - \cos(\frac{2\pi t}{T_r})), & t \in \{0, 1 \ldots, T_r\} \\ 0, \text{ otherwise,} \end{cases} \tag{34}$$

where $\gamma$ is a normalization factor such that $r$ has unitary energy, i.e., $\sum_{t=-\infty}^{\infty} r_t^2 = 1$ and $T_r = 10$, the bound for the variance obtained from (32) is given by $2 \times 1.82 \times 10^{-4} = 3.64 \times 10^{-4}$, which corresponds to a standard deviation of $0.0190$. The maximum value of the actual standard deviation obtained via simulation is $0.0027$, which indicates that the given bound provides a fairly close estimation of the variance. The computation of this bound is illustrated in Figure 6, considering linear scales. As mentioned in Section II-C, we start by computing $|\hat{r}(e^{j\omega})|^2$ and multiplying by $|b(j\omega)| + |c(j\omega)|$. The variance bound is obtained by computing the average value across frequency and multiplying by 2.

## V. Concluding remarks

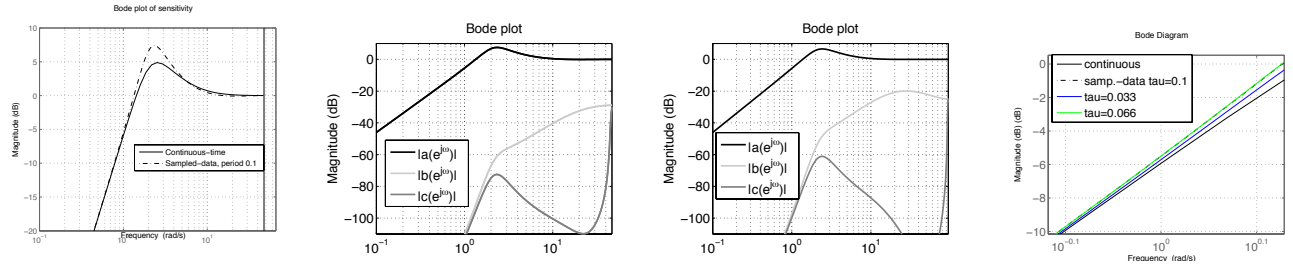
In this paper, we have shown that the frequency-domain analysis framework provided in [20] can be extended to capture a wide range of real-time and networked control systems with delays and data losses. We illustrated how these tools can be useful for sampling period selection in the context of real-time systems subject to delays due to task scheduling.

A future research direction is to extend the results considering models taking the form (1), but for which the assumption given in (3) does not necessarily hold and, in particular consider scenarios where the random vectors $\sigma_k$ can be correlated.




(a) Continous-time and sampled-data$\tau = 0.1$ (b) $\tau = 0.066$ (c) $\tau = 0.033$ (d) $\tau = 0.011$

Fig. 4. Step responses: one realization, mean and standard deviation for several values of the sampling period $\tau$.




(a) Continous-time and $\tau = 0.1$ (b) $\tau = 0.066$ (c) $\tau = 0.033$ (d) $\tau = 0.011$

Fig. 5. Frequency responses for several values the sampling period $\tau$.

## REFERENCES


[1] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, Jan 2007.

[2] O. C. Imer, S. Yksel, and T. Baar, "Optimal control of lti systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429 – 1439, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109806001397

[3] D. E. Quevedo and D. Nesic, "Robust stability of packetized predictive control of nonlinear systems with disturbances and markovian packet losses," *Automatica*, vol. 48, no. 8, pp. 1803 – 1811, 2012.

[4] V. Gupta, A. Dana, J. Hespanha, R. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807–1819, Aug 2009.

[5] T. Gommans, W. Heemels, N. Bauer, and N. v. d. Wouw, "Compensation-based control for lossy communication networks," *International Journal of Control*, vol. 86, no. 10, pp. 1880–1897, 2013.

[6] B. Demirel, "Performance analysis of wireless networked control systems," Ph.D. dissertation, KTH, 2015.

[7] M. Tabbara and D. Nesic, "Input-output stability of networked control systems with stochastic protocols and channels," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1160–1175, June 2008.

[8] D. Antunes, J. P. Hespanha, and C. Silvestre, "Volterra integral approach to impulsive renewal systems: Application to networked control," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 607 –619, march 2012.

[9] L. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, Sept 2004.

[10] W. De Koning and L. Van Willigenburg, "Randomized digital optimal control," in *Non Uniform Sampling Theory and Practice*. New York: Kluwer Acadamic/Plenum Publishers, 2001.

[11] S. Hirche, T. Matiakis, and M. Buss, "A distributed controller approach for delay-independent stability of networked control systems," *Automatica*, vol. 45, no. 8, pp. 1828 – 1836, 2009.

[12] P. Naghshtabrizi and J. Hespanha, "Analysis of distributed control systems with shared communication and computation resources," in *American Control Conference, 2009. ACC '09.*, June 2009, pp. 3384–3389.

[13] S. Al-Areqi, D. Gorges, and S. Liu, "Receding-horizon control and scheduling of systems with uncertain computation and communication delays," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, Dec 2012, pp. 2654–2659.

[14] A. Cuenca, J. Salt, A. Sala, and R. Piza, "A delay-dependent dual-rate PID controller over an ethernet network," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 18–29, Feb 2011.

[15] P. Martí, M. Velasco, R. Villa, J. M. Fuertes, and J. Ayza, "A probabilistic approach to the stability analysis of real-time control systems," in *16th IFAC World Congress, Praha, Czech Republic*, July 2005.

[16] E. Bini and A. Cervin, "Delay-aware period assignment in control systems," in *Real-Time Systems Symposium, 2008*, Nov 2008, pp. 291–300.

[17] M. Donkers, W. Heemels, D. Bernardini, A. Bemporad, and V. Shneer, "Stability analysis of stochastic networked control systems," *Automatica*, vol. 48, no. 5, pp. 917 – 925, 2012.

[18] R. Postoyan and D. Nesic, "On emulated nonlinear reduced-order observers for networked control systems," *Automatica*, vol. 48, no. 4, pp. 645 – 652, 2012.

[19] B. Lincoln and A. Cervin, "Jitterbug: a tool for analysis of real-time control performance," in *2002, Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, Dec 2002, pp. 1319–1324 vol.2.

[20] D. Antunes and W. P. M. H. Heemels, "Frequency-domain analysis of control loops with intermittent data losses," 2014, submitted to journal publication.

[21] O. L. V. Costa, M. Fragoso, and R.P.Marques, *Discrete-Time Markov Jump Linear Systems*. Springer, 2005.

[22] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[23] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. New Jersey, USA: Prentice Hall, 1997.