The Impact of Deadline Misses on the Control Performance of High-End Motion Control Systems

W. Geelen, D. Antunes, *Member, IEEE*, J. P. M. Voeten, R. R. H. Schiffelers, and W. P. M. H. Heemels, *Senior Member, IEEE*

***Abstract*—In high-end motion control systems, the real-time computational platform must execute tasks from multiple control loops operating at high sampling rates. In recent years, traditional special-purpose platforms have been replaced by general-purpose multiprocessor platforms, which introduce significant fluctuations in execution times. While considering worst-case execution times would severely reduce the sampling rates, accepting deadline misses and assuring that the control system still meets the desired specifications is challenging. In this paper, we provide a framework to model and assert the impact of deadline misses in a real-time control loop. We consider stochastic models for deadline misses and characterize the mean and the variance of closed-loop output variables based on a time-domain analysis. We illustrate the usefulness of our framework in the control of a benchmark motion control experimental setup and in the control of a wafer stage in a lithographic machine.**

***Index Terms*—Cyber-physical systems, data losses, deadline misses, hybrid systems, industrial case study, packet drops, performance analysis, real-time systems, stochastic analysis.**

I. Introduction

HIGH-PERFORMANCE real-time control systems require a tight integration between embedded systems and control engineering [1]–[4]. This is the case, e.g., in the context of photolithographic systems for the semiconductor industry, which require nanometer-level positioning precision. These systems incorporate multiple control loops having very high sampling rates to meet the desired precision. This imposes strict constraints on the input–output (IO) delays of a control task schedule, which should be met by the real-time computational platform. Conversely, the design of the controller should take into account the potential and limitations of the computational platform.

In the past decade, traditional special-purpose platforms have been replaced by general-purpose multiprocessor platforms [5], mainly due to reasons of cost and flexibility. This replacement leads to the introduction of significant variability in the task execution times [6, p. 12]. Dimensioning the platform for worst-case execution times is either very costly or results in a large sample-rate reduction and thereby possibly unacceptable performance degradation in the context of high-performance motion control. An alternative is to still use high sampling rates, but allow task deadline misses, i.e., events where tasks are not completed within the sampling period, which in general result in data losses in the control loop. However, deadline misses are often ignored in the analysis and design of the control loops because 1) realistic models of deadline misses suitable for controller design are typically not available; and 2) it is hard to quantify the impact of data losses in the closed-loop performance with state-of-the-practice control methods. Although quite some results on the analysis of control systems subject to data losses are available in the literature (see, e.g., [7]–[17] and the references therein), these results typically lead to (asymptotic) guarantees only on (mean square) stability or quadratic costs and do not immediately provide insights on various important performance indicators such as overshoot, settling time, and rise time, of the time responses. Clearly, the latter performance indicators are of high importance for the controller design and are frequently used in industrial practice for controller tuning. Besides focusing mainly on stability guarantees, the analyses in the networked control systems (NCSs) literature sometimes rely on techniques (e.g., linear matrix inequalities (LMIs) that are not so common in engineering practice. In control engineering practice, one often adopts time-domain or frequency-domain methods (e.g., loop-shaping techniques) for the analysis and synthesis of control systems. As a result, LMI-based tools are not so easily embraced by control engineering in industry. Since deadline misses can significantly degrade the performance of high-end motion control systems, it is extremely important to quantify

Manuscript received April 22, 2015; revised September 29, 2015 and November 10, 2015; accepted November 15, 2015. Date of publication November 30, 2015; date of current version January 8, 2016. This work was supported in part by the Dutch Science Foundation (STW) and in part by the Dutch Organization for Scientific Research (NWO) under the VICI grant "Wireless controls systems: A new frontier in automation" (11382) and under the Robust Cyber-Physical Systems (RCPS) project (12694).

W. Geelen is with the Control Systems Group, Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: wouter.geelen@gmail.com).

D. Antunes and W. P. M. H. Heemels are with the Control Systems Technology Group, Department of Mechanical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: d.antunes@tue.nl; m.heemels@tue.nl).

J. P. M. Voeten is with the Electronic Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: j.p.m.voeten@tue.nl).

R. R. H. Schiffelers is with the Model Driven Software Engineering Group, Department of Mathematics and Computer Science, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: ramon.schiffelers@asml.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2015.2504339

their impact on the control performance in terms of, e.g., overshoot, settling time, or rise time. Understanding this impact provides important information for (re)designing the control system and/or the real-time platform.

Motivated by this lack of tools, the objective of this paper is to develop an analysis and design framework for motion control systems incorporating deadline misses that 1) provides results on important performance indicators such as overshoot, settling time, and rise time; and 2) does connect to the motion control design methods adopted in industry. In particular, we consider control-oriented models of deadline misses and propose a method to assess the impact of deadline misses on the control performance of high-end motion control systems, motivated by an industrial case study for the real-time control of a waferstage of a photolithographic machine produced by ASML.[1] We model the occurrence of deadline misses by random variables resulting from the uncertainty in the execution times of real-time tasks [6, p. 249], [18]. Exploiting the techniques provided in [5], we can obtain the probability distribution of the completion times of control-related tasks. We assume that control-related tasks start to be processed immediately after new sensor measurements are acquired, which occurs at times $\{kh \mid k \in \mathbb{N}\}$, where h is the sampling period. If the completion time of the tasks exceeds the IO delay, denoted by τ and defined as the time from when a sensor measurement is acquired up until the time at which the control signals must be sent to the actuators, a dealine miss occurs. Using the probability distribution of the completion times, we can then compute the probability that a deadline miss occurs as a function of the IO delay τ, as illustrated in Fig. 1. Considering the stochastic model resulting from these considerations, we show how to use time-domain techniques for systems with data losses (see [19], [20]) and the techniques for Markov jump linear systems (MJLS) (see [21]) to analyze the impact of deadline misses on the control performance of motion control systems. Interestingly, our control design framework can incorporate feedforward signals as well as disturbance signals, and complements other existing tools such as [2], [22], [23].

The proposed analysis framework will be applied to two case studies. The first case study pertains to the control of a fourth-order experimental benchmark motion system, used for educational purposes [24]. The second case study considers the control of a waferstage of a lithographic system used at ASML. We will show how our analysis framework provides important information for the (re)design of the control algorithm and/or the real-time platform.

This paper is organized as follows. In Section II, the problem formulation is presented, and we discuss two models for taking into account deadline misses in a control loop. Section III provides methods to analyze the models presented in the previous section. Section IV and Section V illustrate the applicability of the proposed analysis. Section IV considers a benchmark motion control experimental setup, while Section V provides an industrial case study. Section VI provides concluding remarks.

[1]ASML is the world leading manufacturer of photolithography systems for the semiconductor industry.

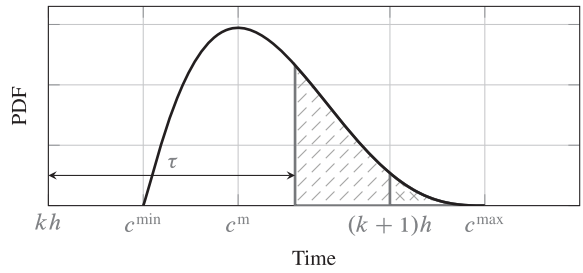

Fig. 1. Probability distribution of the completion time. Herein, $c^{\min} \leq c^{\mathrm{m}} \leq c^{\max}$ denote the minimal, modal, and maximum completion time, being the time at which the schedule is finished. Furthermore, h denotes the sampling period, $k \in \mathbb{N}$ the counter, and τ is the IO delay. We will often assume $\tau = h$, but these might in general be different. In the single hatched region, the task completion time is longer than the specified IO delay and in the doubly hatched region, the task completion time is longer than the sampling period. A deadline miss occurs if the completion time is longer than the IO delay. Therefore, the probability of a deadline miss occurring coincides with the area of the total hatched region.

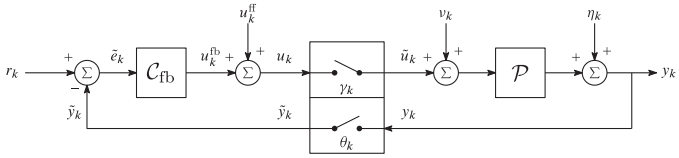

Fig. 2. $\mathcal{P}$ is the plant, $\mathcal{C}_{\mathrm{fb}}$ is the feedback controller, and $\gamma_k \in \{0, 1\}$ and $\theta_k \in \{0, 1\}$ indicate if data losses occur at discrete time $k \in \mathbb{N}$.

II. Problem Formulation

In this paper we consider industrial high-end motion control systems in which several control tasks are executed on a general-purpose multiprocessor platform. The main tasks that typically have to be executed by a control loop are: 1) acquire sensor data from the IO board; 2) compute the control inputs for the actuators; and 3) output the control inputs to the IO board. Due to the variability on the execution times of the tasks that the real-time platform must perform (see Fig. 1), deadline misses may occur in any of these tasks. These deadline misses may cause data losses in the control loop. This is portrayed in Fig. 2 considering a single-input single-output (SISO) control loop for which data losses can either occur at the plant's input or at the output.

The plant $\mathcal{P}$ and the controller $\mathcal{C}_{\mathrm{fb}}$ are assumed to be described by the linear models

$$\begin{aligned} x_{k+1}^p &= A_p x_k^p + B_p(\tilde{u}_k + \nu_k) \\ y_k &= C_p x_k^p + \eta_k \end{aligned} \tag{1}$$

and

$$\begin{aligned} x_{k+1}^c &= A_c x_k^c + B_c(r_k - \tilde{y}_k) \\ u_k^{\mathrm{fb}} &= C_c x_k^c + D_c(r_k - \tilde{y}_k) \\ u_k &= u_k^{\mathrm{ff}} + u_k^{\mathrm{fb}} \end{aligned} \tag{2}$$

respectively, where $x_k^p \in \mathbb{R}^{n_p}$ and $x_k^c \in \mathbb{R}^{n_c}$ denote the state of the plant and of the feedback controller at time $t_k := kh$, with h being the sampling period and $k \in \mathbb{N}$ denoting the discrete time. Moreover, u_k^{fb}, u_k^{ff}, u_k, and $\tilde{u}_k \in \mathbb{R}^{n_u}$ denote the output of the feedback controller, the feedforward control input

signal, the control output (sum of the feedback control and feedfoward control input signal), and the available control signal at the plant, respectively. The available control input at the plant would coincide with the output of the feedback controller in the absence of data losses. We will model shortly how the two are related in the presence of data losses. The output of the plant and the available output of the plant at the controller are denoted by y_k and $\tilde{y}_k \in \mathbb{R}^{n_y}$, respectively. Furthermore, $\nu_k \in \mathbb{R}^{n_u}$ and $\eta_k \in \mathbb{R}^{n_y}$ denote the disturbance at the input and the output of the plant, respectively, at discrete time $k \in \mathbb{N}$, with mean $\mu_\nu \in \mathbb{R}^{n_u}$ and $\mu_\eta \in \mathbb{R}^{n_y}$ and covariance $\Sigma_\nu \in \mathbb{R}^{n_u \times n_u}$ and $\Sigma_\eta \in \mathbb{R}^{n_y \times n_y}$, respectively. In addition, $r_k \in \mathbb{R}^{n_y}$ is the reference signal, at discrete time $k \in \mathbb{N}$. Finally, the real and available error are defined as $e_k := r_k - y_k$ and $\tilde{e}_k := r_k - \tilde{y}_k$, respectively, at discrete time $k \in \mathbb{N}$. For the remainder of this paper, we assume that for every $k \in \mathbb{N}$, r_k, and u_k^{ff} are deterministic and bounded signals.

A. Modeling Data Losses Due to Deadline Misses

Data losses can occur in the controller-to-actuator channel (c–a) and/or in the sensor-to-controller channel (s–c). Consider for now that data losses only occur in the controller-to-actuator channel (c–a). When a task misses its deadline, there are several possible scenarios for the processor to proceed. We consider two scenarios.

1) The processor aborts the control task and the actuator uses the latest available control output.
2) The actuator uses the latest available control output (scenario 1) and the processor continues processing the control task. The computed control output is not used in the current sampling period, but is prepared and saved for the next period. As a result, in the event of another deadline miss in the next period, a more recent control signal is available.

1) Scenario 1: In this scenario, the control task is aborted. As a consequence, the actuators hold their previous input values. If we assume that the controller has a single output, this behavior can be captured by

$$\tilde{u}_k = (1 - \gamma_k)\tilde{u}_{k-1} + \gamma_k u_k \tag{3}$$

where γ_k equals zero if data losses occur and one if no data losses occur at discrete time $k \in \mathbb{N}$. Herein, $\gamma := \{\gamma_k \mid k \in \mathbb{N}\}$ is a discrete-time stochastic process for which we adopt the following assumption.

Assumption 1: The discrete time stochastic process γ is independent and identically distributed (i.i.d.), i.e., for every $k_1, k_2 \in \mathbb{N}$ with $k_1 \neq k_2$, γ_{k_1} and γ_{k_2} are statistically independent random variables with a common probability distribution.

As a result, the probability $\mathbb{P}[\gamma_k = 1] = p_\gamma$ does not depend on k. In particular, γ is a Bernoulli process, which is a special Markov chain as depicted in Fig. 3 and described by a transition matrix

$$P_\gamma = \begin{bmatrix} p_\gamma & 1 - p_\gamma \\ p_\gamma & 1 - p_\gamma \end{bmatrix}$$

where the entries of the first column indicate the probabilities $\mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 0]$ and $\mathbb{P}[\gamma_{k+1} = 1 \mid \gamma_k = 1]$ and

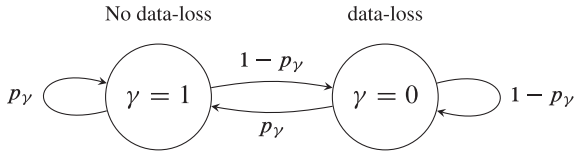

Fig. 3. Markov chain modeling the i.i.d. process γ.

the entries of the second column indicate the probabilities $\mathbb{P}[\gamma_{k+1} = 0 \mid \gamma_k = 0]$ and $\mathbb{P}[\gamma_{k+1} = 0 \mid \gamma_k = 1]$. This Bernoulli process is able to model data losses at the plant's input. In a similar fashion, data losses can be modeled at the plant's output. To model these, we use

$$\tilde{y}_k = (1 - \theta_k)\tilde{y}_{k-1} + \theta_k y_k \tag{4}$$

where, as before, θ_k equals zero if data losses occur and equals one if no data losses occur at discrete time $k \in \mathbb{N}$. Likewise, θ is a Bernoulli process corresponding to the chain with a transition matrix P_θ of the following special structure

$$P_\theta = \begin{bmatrix} p_\theta & 1 - p_\theta \\ p_\theta & 1 - p_\theta \end{bmatrix}$$

where p_θ denotes $\mathbb{P}[\theta_k = 1]$.

We introduce the mode σ_k to indicate combinations of data losses that occur at discrete time $k \in \mathbb{N}$, as

$$\sigma_k := \begin{array}{l} 1, \text{ if } (\gamma_k, \theta_k) = (1, 1) \\ 2, \text{ if } (\gamma_k, \theta_k) = (1, 0) \\ 3, \text{ if } (\gamma_k, \theta_k) = (0, 1) \\ 4, \text{ if } (\gamma_k, \theta_k) = (0, 0). \end{array} \tag{5}$$

Similar to γ and θ, σ is a Markov chain. Its transition matrix is obtained by the Cartesian product of the two Markov chains of γ and θ, i.e., $P_\sigma = P_\gamma \otimes P_\theta$, where $\otimes$ denotes the Kronecker product [25]. Note that the proposed model for scenario 1 can be easily extended to capture multiple input multiple output (MIMO) systems as well. In fact, in Section II-B, we will provide the complete model for the MIMO case where each entry in the plant's input and each entry in the plant's output satisfy a model as in (3) and (4), respectively.

2) Scenario 2: In contrast to scenario 1, in this scenario a control task is not aborted when it exceeds its deadline. This behavior can be caused by a static-order task scheduler, which schedules the tasks in a certain order after which the tasks cannot be deleted or stopped.

A consequence of this scenario is that a finite number of control values need to be memorized in the model up to a certain horizon $n_{\hat{u}}$, which is the worst-case number of consecutive deadline misses, assuming it is finite. This behavior can be captured by

$$\tilde{u}_k = \begin{cases} u_k, & \text{if } \psi_k = \mathsf{e}_1, \text{no lag} \\ u_{k-1}, & \text{if } \psi_k = \mathsf{e}_2, 1 \text{ sample lag} \\ \vdots & \\ u_{k-n_{\hat{u}}}, & \text{if } \psi_k = \mathsf{e}_{n_{\hat{u}}+1}, n_{\hat{u}} \text{ samples lag} \end{cases}$$

where $\psi_k := \begin{bmatrix} \psi_{1,k} & \dots & \psi_{n_{\hat{u}}+1,k} \end{bmatrix} \in \{\mathsf{e}_1, \dots, \mathsf{e}_{n_{\hat{u}}+1}\}$ indicates which input is applied to the plant; e_i denote the vector of the

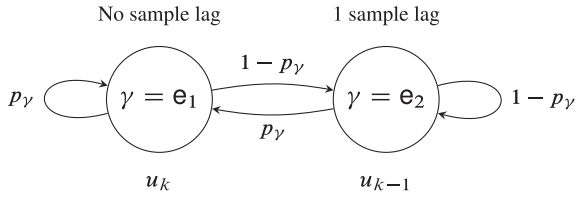

Fig. 4. Markov chain modeling the case of one sample lag.

standard basis in $\mathbb{R}^{n_{\hat{u}}+1}$, e.g., $\mathsf{e}_2 = \begin{bmatrix}0 & 1 & 0 & \cdots & 0\end{bmatrix}$. In particular, when $\psi_k = \mathsf{e}_i$, $\tilde{u}_k = u_{k-i+1}$ is applied. The available input at the plant can now be described by the following linear model:

$$\hat{u}_k = A_u \begin{bmatrix} u_k \\ \hat{u}_{k-1} \end{bmatrix}, \quad \tilde{u}_k = \psi_k \begin{bmatrix} u_k \\ \hat{u}_{k-1} \end{bmatrix} \tag{6}$$

where

$$A_u := \begin{bmatrix} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(n_{\hat{u}}+1)\times(n_{\hat{u}}+1)}$$

and $\hat{\boldsymbol{u}}_{k-1} := \begin{bmatrix} u_{k-1} & \ldots & u_{k-n_{\hat{u}}} \end{bmatrix}^\intercal \in \mathbb{R}^{n_{\hat{u}}}$ is a vector of previously computed control outputs.

In the industrial case study of Section V, we focus on at most 1 sample lag, i.e., $n_{\hat{u}} = 1$. However, as indicated above, higher values for $n_{\hat{u}}$ can be described as well. Assuming again that the data losses resulting from deadline misses are independent, we can model this particular scenario using a Markov chain as illustrated in Fig. 4. Hence, we obtained a Markov chain similar to scenario 1, which is able to model data losses in the c–a channel.

The s–c channel and also MIMO systems can be studied by direct extensions of these ideas.

B. MJLS Model

The closed-loop system (see, e.g., Fig. 2) can have different modes due to data losses, see (5). These mode switches are governed by the discrete-time stochastic process σ. A crucial step in our analysis is to note that, when the stochastic process σ is a Markov chain, the closed-loop system can be described by a discrete-time MJLS [21]. A discrete-time MJLS is a class of models taking the general form

$$\begin{aligned} \rho_{k+1} &= M_{\pi_k}\rho_k + N_{\pi_k}\omega_k \\ \upsilon_k &= Q_{\pi_k}\rho_k + R_{\pi_k}\omega_k \end{aligned} \tag{7}$$

where ρ_k, υ_k, ω_k denote the state, output, and input signals, respectively, and $\pi_k \in \mathcal{M}$ is described by a Markov chain taking values in a finite set $\mathcal{M}$ with transition probabilities $P_{ij} = \mathbb{P}\left[\pi_{k+1} = i \mid \pi_k = j\right]$ for $i, j \in \mathcal{M}$. We show next how to model the closed-loop system with deadline misses by an MJLS, with: 1) the state ρ incorporating the plant and controller states and auxiliary states; 2) π coinciding with the stochastic process σ; and 3) ω_k and υ_k being the external inputs and output of the closed-loop system, respectively. We also specify next the matrices appearing in the model (7).

We consider here for reasons of generality an MIMO system assuming that each entry of the plant's input and each entry of the plant's output satisfy a model as in (3) and (4), respectively. Naturally, it can also happen that certain entries in the plant's output and plant's input are gathered in one node and all of them miss the deadline or all of them do not miss the deadline. However, here we consider that each entry can have a deadline miss independently of the other entries. In this case assuming the behavior as in scenario 1, the closed-loop system consisting of (1)–(5) leads to

$$\begin{aligned} \xi_{k+1} &= A_{\sigma_k}\xi_k + B_{\sigma_k}r_k + E_{\sigma_k}u_k^{\mathrm{ff}} + G_{\sigma_k}w_k \\ \zeta_k &= C_{\sigma_k}\xi_k + D_{\sigma_k}r_k + F_{\sigma_k}u_k^{\mathrm{ff}} + H_{\sigma_k}w_k \end{aligned} \tag{8}$$

where $\xi_k := \begin{bmatrix} x_k^p & x_k^c & \tilde{y}_{k-1} & \tilde{u}_{k-1} \end{bmatrix}^\intercal$, $\zeta_k := y_k$ is the output of the plant, and $w_k := \begin{bmatrix} \nu_k^\intercal & \eta_k^\intercal \end{bmatrix}^\intercal$. The matrices for the state equation are

$$A_{\underline{\sigma}} := \begin{bmatrix} A_p - B_p\underline{\Gamma}D_c\underline{\Theta}C_p & B_p\underline{\Gamma}C_c & -B_p\underline{\Gamma}D_c(I-\underline{\Theta}) & B_p(I-\underline{\Gamma}) \\ -B_c\underline{\Theta}C_p & A_c & -B_c(I-\underline{\Theta}) & 0 \\ \underline{\Theta}C_p & 0 & (I-\underline{\Theta}) & 0 \\ -\underline{\Gamma}D_c\underline{\Theta}C_p & \underline{\Gamma}C_c & -\underline{\Gamma}D_c(I-\underline{\Theta}) & (I-\underline{\Gamma}) \end{bmatrix}$$

$$B_{\underline{\sigma}} := \begin{bmatrix} B_p\underline{\Gamma}D_c \\ B_c \\ 0 \\ \underline{\Gamma}D_c \end{bmatrix}, E_{\underline{\sigma}} := \begin{bmatrix} B_p\underline{\Gamma} \\ 0 \\ 0 \\ \underline{\Gamma} \end{bmatrix}, G_{\underline{\sigma}} := \begin{bmatrix} B_p & -B_p\underline{\Gamma}D_c\underline{\Theta} \\ 0 & -B_c\underline{\Theta} \\ 0 & \underline{\Theta} \\ 0 & -\underline{\Gamma}D_c\underline{\Theta} \end{bmatrix}$$

where $\underline{\sigma} \in \{1, \ldots, 2^{n_u+n_y}\}$, $\underline{\Gamma} \in \{\mathrm{diag}(\underline{\gamma}^1, \ldots, \underline{\gamma}^{n_u}) \mid \underline{\gamma}^i \in \{0,1\}\}$, and $\underline{\Theta} \in \{\mathrm{diag}\left(\underline{\theta}^1, \ldots, \underline{\theta}^{n_y}\right) \mid \underline{\theta}^i \in \{0,1\}\}$. The matrices for the output equation depend on the output of interest. For example, if we are interested in the output of the plant y_k we have

$$C_{\underline{\sigma}} := \begin{bmatrix} C_p & 0 & 0 & 0 \end{bmatrix}, \quad D_{\underline{\sigma}} := 0, \quad F_{\underline{\sigma}} := 0, \quad H_{\underline{\sigma}} := \begin{bmatrix} 0 & I \end{bmatrix}$$

and if we are interested in the error $y_k - r_k$ we have

$$C_{\underline{\sigma}} := \begin{bmatrix} C_p & 0 & 0 & 0 \end{bmatrix}, \quad D_{\underline{\sigma}} := -I, \quad F_{\underline{\sigma}} := 0, \quad H_{\underline{\sigma}} := \begin{bmatrix} 0 & I \end{bmatrix}.$$

Note that in this case, the matrices $C_{\underline{\sigma}}$, $D_{\underline{\sigma}}$, $F_{\underline{\sigma}}$, and $H_{\underline{\sigma}}$ do not depend on the mode $\underline{\sigma}$, but for reasons of generality we allow $C_{\underline{\sigma}}$, $D_{\underline{\sigma}}$, $F_{\underline{\sigma}}$, and $H_{\underline{\sigma}}$ to depend on $\underline{\sigma}$, as well. For instance, if the output ζ_k of (8) is chosen as u_k, then these matrices depend on $\underline{\sigma}$.

Due to the assumption that each entry of the plant's input and plant's output follows its own Bernoulli-type model (independent of the other entries), we have that each of the modes $\underline{\sigma} \in \{1, 2, \ldots, 2^{n_u+n_y}\}$ has a probability $p_{\underline{\sigma}}$ of occurring (which is independent of time, or what happened at the previous time step). This is basically the modeling setup we will be working with in this paper, which is a special form of an MJLS.

Also in the case of scenario 2, a similar special form of an MJLS can be formulated.

III. Analysis

The analysis of a motion control loop in industrial practice is often pursued by either using time-domain or frequency-domain techniques for time-invariant systems. The time-domain analysis evaluates the output responses to reference signals of interest, e.g. step functions, via simulation and is concerned with properties such as settling time, rise time, and overshoot. The frequency-domain analysis relies typically on Bode plots of (complementary) sensitivity (transfer) functions [26] to determine how the closed-loop tracks desired input signals and rejects disturbance signals.

However, the model (8) that we have obtained to capture deadline misses in the control loop is in general a *time-varying* model, for which such time-domain and frequency-domain techniques are not available, as they rely heavily on the property of time invariance. In fact, concepts such as step responses and sensitivity plots [26, p. 151, 279] are only useful or even applicable to time-invariant models, which is a condition not satisfied in our setup of Sections II-A1 and II-A2 if deadline misses would occur.

In this section, we present an approach which still allows to define concepts as step responses for the system of interest. To this effect, we exploit the stochastic structure in (8), i.e., the fact that the data losses caused by deadline misses are captured by an MJLS [21].

Key to our analysis is to notice that for such models, the statistical moments such as the mean and the variance are described by *time-invariant* systems. This enables us to provide similar techniques for the considered class of models (8) as for time-invariant systems. Although recently a frequency-domain analysis for time-varying systems as in (8) has been developed in [19] and [20], in this work we focus on time-domain techniques for reasons of compactness.

We introduce the following definitions pertaining to the matrices $A_{\underline{\sigma}}$ and $B_{\underline{\sigma}}$ in (8), which exhibit 1 to N different modes, i.e., $\underline{\sigma} \in \{1, 2, \ldots, N\}$, according to the Markov chain σ described by a transition matrix P_σ

$$\overline{A} := \mathbb{E}\left[A_\sigma\right] = \textstyle\sum_{i=1}^{N} p_i A_i$$
$$\underline{A} := \mathbb{E}\left[A_\sigma \otimes A_\sigma\right] = \textstyle\sum_{i=1}^{N} p_i (A_i \otimes A_i).$$

Herein, $\mathbb{E}\left[\cdot\right]$ is the expectation operator and $p_i = \mathbb{P}\left[\sigma_k = i\right]$. In addition, for A and B of appropriate sizes, let T be the unique matrix such that $TA \otimes B = B \otimes A$. Then, we define

$$\begin{aligned}\underline{N}_B^A &:= \mathbb{E}\left[A_\sigma \otimes B_\sigma + B_\sigma \otimes A_\sigma T\right] \\ &:= \textstyle\sum_{i=1}^{N} p_i (A_i \otimes B_i + B_i \otimes A_i T)\end{aligned}$$

We assume the following stability notion, typically considered for MJLS [21, Ch. 3].

Assumption 2: The (unforced) system (8) with $r_k = 0$ and $u_k^{\mathrm{ff}} = 0$ for all $k \in \mathbb{N}$ is mean square stable, i.e., for every ξ_0, $\lim_{k\to\infty} \mathbb{E}\left[\|\xi_k\|^2\right] = 0$.

The MJLS is mean square stable (MSS) iff $\underline{A}$ is Schur [21, p. 36], i.e., all eigenvalues λ_i of $\underline{A}$ satisfy $|\lambda_i| < 1$.

Consider the MJLS in (8). As we show next, although the system is time varying, the expected value and the second-order statistical moments of the state can be described by linear time-invariant systems. Let $\mathrm{vec}\,(A)$ be the vectorization operation as defined in [25, p. 60] for square matrices, and let its inverse be defined by $\mathrm{vec}^{-1}\left(\mathrm{vec}\,(A)\right) := A$.

Theorem 1: Consider the MJLS in (8) and suppose that Assumptions 1 and 2 hold. Then,

$$\begin{aligned}\beta_{k+1} &= \overline{A}\beta_k + \overline{B} r_k + \overline{E} u_k^{\mathrm{ff}} + \overline{G}\mu_{\nu\eta} \\ \alpha_k &= \overline{C}\beta_k + \overline{D} r_k + \overline{F} u_k^{\mathrm{ff}} + \overline{H}\mu_{\nu\eta}\end{aligned} \tag{9}$$

where $\beta_k = \mathbb{E}\left[\xi_k\right]$ is the expected value of the state, $\alpha_k = \mathbb{E}\left[\zeta_k\right]$ is the expected value of the output and $\mu_{\nu\eta} = \left[\mu_\nu^{\mathsf{T}} \ \mu_\eta^{\mathsf{T}}\right]^{\mathsf{T}}$ is the expected value of w_k, and $\overline{A}$ is a Schur matrix. Moreover, the covariance of the output ζ_k is given by

$$\begin{aligned}\mathrm{covar}\,(\zeta_k) &= \mathbb{E}\left[(\zeta_k - \mathbb{E}\left[\zeta_k\right])(\zeta_k - \mathbb{E}\left[\zeta_k\right])^{\mathsf{T}}\right] \\ &= \mathbb{E}\left[\zeta_k \zeta_k^{\mathsf{T}}\right] - \alpha_k \alpha_k^{\mathsf{T}}\end{aligned} \tag{10}$$

where $\mathbb{E}\left[\zeta_k \zeta_k^{\mathsf{T}}\right] = \mathrm{vec}^{-1}\left(\varphi_k\right)$ and $\varphi_k := \mathbb{E}\left[\zeta_k \otimes \zeta_k\right]$ is obtained in terms of $\vartheta_k = \mathbb{E}\left[\xi_k \otimes \xi_k\right]$ as follows

$$\begin{aligned}\vartheta_{k+1} = {} & \underline{A}\vartheta_k + \underline{B}(r_k \otimes r_k) + \underline{E}(u_k^{\mathrm{ff}} \otimes u_k^{\mathrm{ff}}) + \underline{G}\Sigma_{\nu\eta} \\ & + \underline{N}_B^A(\beta_k \otimes r_k) + \underline{N}_E^A(\beta_k \otimes u_k^{\mathrm{ff}}) + \underline{N}_G^A(\beta_k \otimes \mu_{\nu\eta}) \\ & + \underline{N}_E^B(r_k \otimes u_k^{\mathrm{ff}}) + \underline{N}_G^B(r_k \otimes \mu_{\nu\eta}) \\ & + \underline{N}_G^E(u_k^{\mathrm{ff}} \otimes \mu_{\nu\eta}) \\ \varphi_k = {} & \underline{C}\vartheta_k + \underline{D}(r_k \otimes r_k) + \underline{F}(u_k^{\mathrm{ff}} \otimes u_k^{\mathrm{ff}}) + \underline{H}\Sigma_{\nu\eta} \\ & + \underline{N}_D^C(\beta_k \otimes r_k) + \underline{N}_F^C(\beta_k \otimes u_k^{\mathrm{ff}}) + \underline{N}_H^C(\beta_k \otimes \mu_{\nu\eta}) \\ & + \underline{N}_F^D(r_k \otimes u_k^{\mathrm{ff}}) + \underline{N}_H^D(r_k \otimes \mu_{\nu\eta}) \\ & + \underline{N}_H^F(u_k^{\mathrm{ff}} \otimes \mu_{\nu\eta})\end{aligned} \tag{11}$$

where $\Sigma_{\nu\eta} = \mathrm{diag}\,(\Sigma_\nu, \Sigma_\eta) = \left[\Sigma_\nu \ 00 \ \Sigma_\eta\right]$.

Proof: Equation (9) can be obtained by taking expected values in the state equation in (8) and using the fact that $\mathbb{E}[A_{\sigma_k}\xi_k] = \mathbb{E}[A_{\sigma_k}]\mathbb{E}[\xi_k]$. This latter property follows from the fact that $\mathbb{E}[\xi_k]$ is only a function of the random variables $\sigma_{k-1}, \sigma_{k-2}, \ldots$ which are independent of σ_k due to Assumption 1. The statement that (9) is stable in the sense that $\overline{A}$ is Schur is a consequence of Assumption 2 and follows from the fact that $\mathbb{E}\left[\xi_k\right]\mathbb{E}\left[\xi_k\right]^{\mathsf{T}} \leq \mathbb{E}\left[\xi_k \xi_k^{\mathsf{T}}\right]$. That is, when the variance converges the expected value converges as well, i.e., when $\underline{A}$ is Schur than $\overline{A}$ is Schur as well. The equations for the variance can be obtained by using the properties of the Kronecker product and appealing again to Assumption 1 in a similar fashion to the argument used to obtain (9). ■

The LTI systems (9) and (11) provide us the means to infer information about the behavior of the time-varying system (MJLS) (8) in terms, of the first and second moments and thus also the covariance of the output ζ_k (10). This gives valuable information about the behavior (e.g., step responses) of the system by using classical tools for LTI systems. In a similar manner, also information about higher order moments can be obtained, if needed. Note that this analysis can be carried out when different reference signals are applied (e.g, sinusoids, triangle, square, or periodic signals). For each signal, we can quantify if the data losses lead to acceptable performance degradation in terms of the time responses. It is also

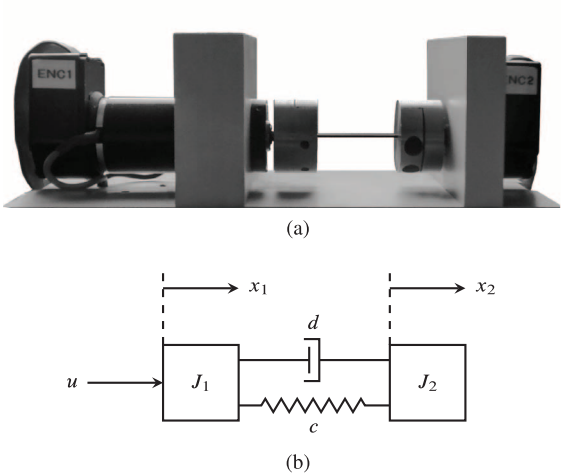

Fig. 5. PATO system. (a) Front view picture of the PATO system. (b) Schematic of the dynamical model.

possible to perform a frequency-domain analysis, as proposed in [19], and reason in terms of this analysis on the behavior of the output to any reference signal, in a similar fashion to the frequency-domain analysis for LTI systems. However, we do not pursue this analysis here for the sake of brevity. From this frequency-domain analysis, we can also conclude that the data losses do not affect steady-state errors, i.e., the asymptotic errors obtained when the input signal becomes a constant, possibly after a given time. This means that the asymptotic variance of the output response to a constant signal is zero and the asymptotic expected value coincides with the steady-state value obtained in the absence of data losses (see [19] for further details).

In the next two sections, we illustrate the usefulness of the proposed methods for two relevant case studies.

IV. Academic Case Study

The PATO system is a dual rotary fourth-order single input multiple output (SIMO) motion system. It consists of two loads, which are connected to each other by a thin, flexible bar. One of the loads is directly connected to the motor and two encoders measure the rotation of each load. The physical system and a schematic of the dynamical model are shown in Fig. 5.

The dynamics of the PATO system are described by

$$J_1\ddot{x}_1 = c(x_2 - x_1) + d(\dot{x}_2 - \dot{x}_1) + u$$
$$J_2\ddot{x}_2 = c(x_1 - x_2) + d(\dot{x}_1 - \dot{x}_2).$$

Herein, J_1 and J_2 are the moments of inertia in [kg m^2] of the two loads, d is the damping factor in [N m s] and c is the spring constant in [N m]. We will focus in this study on controlling the first load. This results in the continuous time-invariant model described by

$$\begin{aligned} \dot{x}^p(t) &= A_p x^p(t) + B_p u(t) \\ y(t) &= C_p x^p(t) \end{aligned} \tag{12}$$

where $t \in \mathbb{R}_{\geq 0}$ and

$$A_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -c/J_1 & -d/J_1 & c/J_1 & d/J_1 \\ 0 & 0 & 0 & 1 \\ c/J_2 & d/J_2 & -c/J_2 & -d/J_2 \end{bmatrix}, \quad B_p = \begin{bmatrix} 0 \\ 1/J_1 \\ 0 \\ 0 \end{bmatrix}$$
$$C_p = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

By a three-point frequency measurement [27, p. 11], the frequency response of the system was first obtained. The model (12) was then fitted in the frequency domain, leading to the parameters

$$J_1 := 2.4561 \cdot 10^{-4}, \quad J_2 := 2.4642 \cdot 10^{-4}$$
$$d := 7.7954 \cdot 10^{-4}, \text{ and } c := 16.069.$$

A controller was designed such that the closed-loop system has a bandwidth of approximately 10 (Hz). It consists of the series of a lead-lag controller, with transfer function having a zero at 3 Hz and pole at 30 Hz and with gain 0.05, and a second-order notch filter described by the transfer function $\frac{n_{f_1,\beta_1}(s)}{n_{f_2,\beta_2}(s)}$ where $n_{f,\beta} = \frac{1}{(2\pi f)^2}s^2 + \frac{2\beta}{2\pi f}s + 1$ and $f_1 = 57$, $f_2 = 58$, $\beta_1 = 0.01$, $\beta_2 = 0.1$. The feed-forward control input signal is set to zero. Both the PATO model and the controller are discretized using the bilinear (Tustin) method [28] with a sampling period of $h = 1$ (ms), leading to the plant and controller models (1) and (2).

Remark 1: The setup has a limited control input range and as a result, the control effort is saturated. This causes nonlinear behavior, which is not captured by our model. As a consequence, we designed our controller such that the control effort stays within the boundaries of the saturation. Furthermore, the setup is under the influence of other nonlinear phenomena such as quantization and coulomb and viscous friction.

We investigate the behavior of the PATO system with emulated data losses in the c–a and s–c channels corresponding to the behavior described in scenario 1, see Section II-A1. Hence, $n_u = 1$ and $n_y = 1$ result in $2^{1+1} = 4$ modes. Fig. 6(a) and (b) shows the results for 250 simulations and experiments for the case where a step function is applied as a reference. On top of the simulation and experimental results, the analytically computed expected value $\mathbb{E}[y_k]$ and standard deviation σ_k are shown using the methodology based on (9) and (11) described in Section III. Fig. 7 shows the analytically computed variance and the variance computed based on the results from the simulation and experiments.

From Fig. 6(a), with data losses in the c–a channel, we observe that the simulations of the model closely follow the results, which we obtained analytically. The experimental results show the same behavior, although the results seem to be more concentrated. This is verified by the variance given in the left plot of Fig. 7. Looking at Fig. 6(b), in which data losses occur in the s–c channel, both the model and the experiments closely follow the results that we obtained analytically. In addition, from Fig. 6 we observe that data losses that occur in the c–a channel have a more significant impact on the responses than data losses that occur at the s–c channel.

Looking at the left plots of Fig. 6(a) and (c), which show the simulation results for 25%, 50%, and 75% data losses, it is

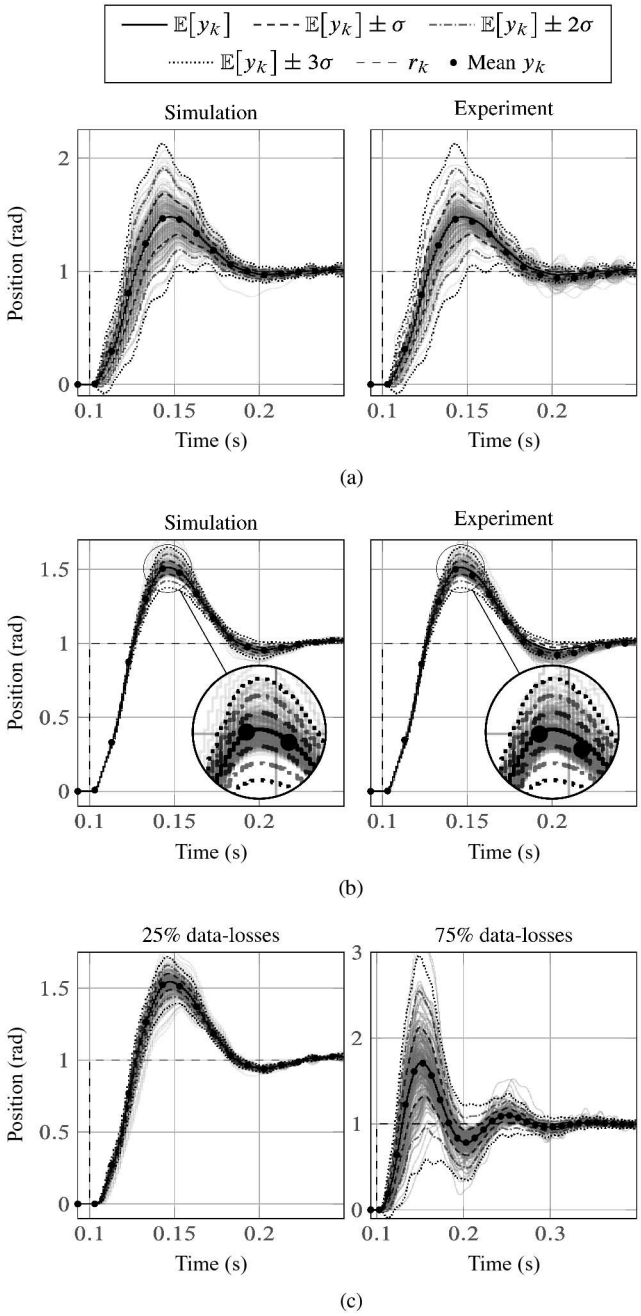

Fig. 6. Step response of the PATO system and its model for 250 simulations, 250 experiments and its analytically computed expected value $\mathbb{E}[y_k]$ and standard deviations σ. (a) 50% chance of data losses in the controller-to-actuator channel, i.e., $\mathbb{P}[\gamma_k = 0] = 0.5$ and $\mathbb{P}[\theta_k = 0] = 0$. (b) 60% chance of data losses in the sensor-to-controller channel, i.e., $\mathbb{P}[\gamma_k = 0] = 0$ and $\mathbb{P}[\theta_k = 0] = 0.6$. (c) Two simulations for 25% and 75% chance of data losses in the controller-to-actuator channel.

clear that an increase in the deadline miss probability increases the uncertainty in the output responses to deterministic inputs such as steps responses. The proposed analysis allows to quantify this impact of the deadline misses on the output response. In this specific experimental study, one can conclude that 25% of data losses does not degrade the output responses significantly when compared to the case where deadline misses are absent. However, the degradation of 50% data losses is quite notorious and a degradation of 75% data losses is not acceptable. Using

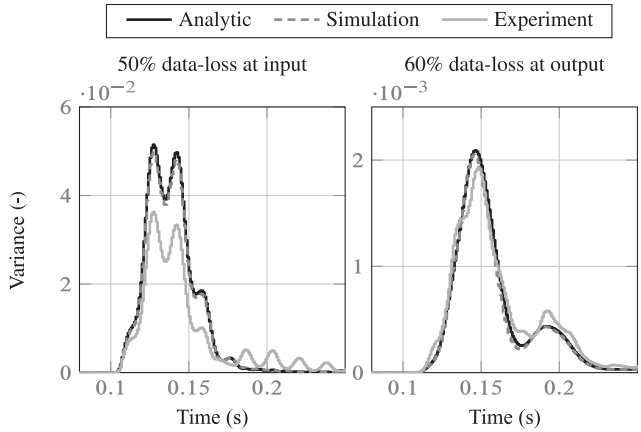

Fig. 7. Comparison of the variance computed analytically, by the model simulations and by the experiment results. Left plot corresponds to 50% chance of data losses at the controller-to-actuator channel. Right plot corresponds to 60% chance of data losses at sensor-to-controller channel.

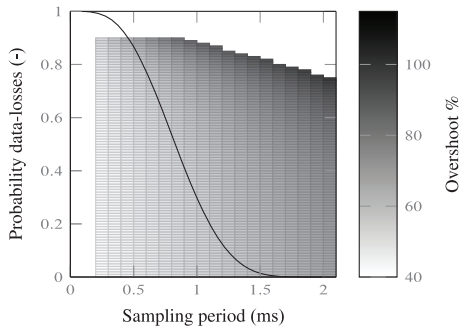

Fig. 8. Percentage of the expectation of overshoot, shown in a heat-plot, as a function of the probability of data losses occurring in the c–a channel and the sampling period. On top, the probability of data losses as a function of the sampling period is plotted, which is obtained from Fig. 1.

our methods we are able to quickly determine the impact of data losses. As a result, we are able to assess, for instance, the overshoot as a function of the probability that data losses occur in the c–a channel and the sampling period, the result of which is a three-dimensional (3-D) plot in Fig. 8. Indeed, based on Fig. 1, we can also plot the probability of data losses as a function of the sampling period (solid line in 8). The area underneath the probability distribution in Fig. 1 defines the probability that data losses will occur. By plotting the probability of data losses as a function of the sampling period on top of the 3-D plot Fig. 8, we are able to identify which sampling period is to be chosen to achieve the largest performance (expressed in terms of smallest overshoot here). This demonstrates how our tools can be used effectively in the determination of the sampling period, even though data losses occur. A traditional design in this case would choose a sampling period conservatively to avoid data losses which would correspond to a sampling period of 1.6 (ms) just to be able to apply classical LTI design techniques. Clearly in terms of percentage of overshoot, this is not the optimal choice and by using our new methodology, which is still based on LTI analysis techniques, much better designs and selection of the sampling period can be obtained.

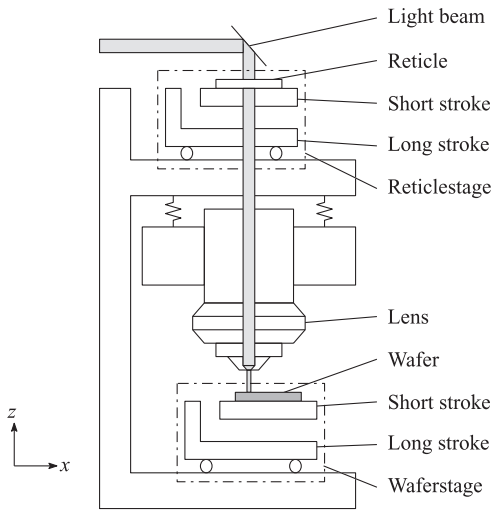

Fig. 9. Schematics of a photolithography system and its main components [30]. Light passes through a mask, reticle, a lens and onto a die on the silicon wafer. Both reticle and wafer are positioned by accurately controlled stages that perform repeated scanning motions while exposing subsequent dies.

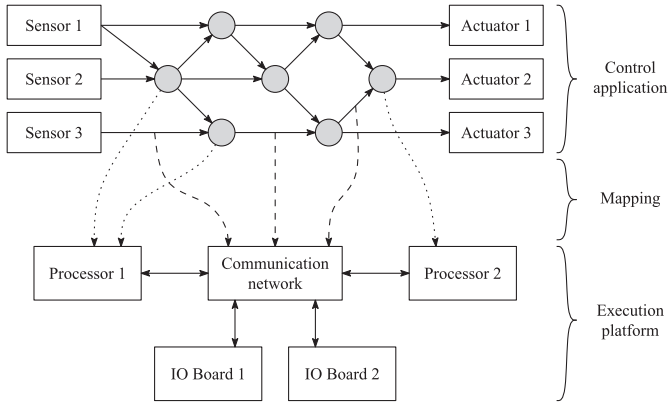

Fig. 10. Control application mapped on execution platform. Note that there is no mapping from the sensors, or actuators, to the IO boards since the mapping which sensor, or actuator, is connected to which IO board is already *priori* known.

V. Industrial Case Study

ASML Holding N.V. [29] is the world leading manufacturers of photolithography systems for the semiconductor industry. These machines have extremely high requirements regarding accuracy and throughput. Fig. 9 shows the schematics of the system and its main components.

One of the critical components of the system is the waferstage, a platform, which transports a silicon wafer in the system during exposure. The waferstage consists of the long stroke and the short stroke. While the long stroke has a positioning accuracy of micrometers, the short stroke has a positioning accuracy of nanometers, which is required for the photolithography process. Both the long and the short stroke have six degrees of freedom—the position along the three spatial axis, which we denote by x, y, z and three rotation angles around each axis, which we denote by R_x, R_y, R_z,—and are controlled by multiple processors. The processors communicate with multiple IO boards, which are connected to a number of sensors and actuators.

Fig. 10 illustrates a control application and the mapping on a multiprocessor execution platform. The control application consists of a number of tasks, which are scheduled on the processors. The task schedule is divided into two sets of tasks: 1) critical tasks that have to be executed before the specified IO delay; and 2) noncritical tasks that perform preparatory work for the next sampling period. The IO delay as well as the sampling period of the system are specified as $\tau = h = 50$ [μs], i.e., control loops run at 20 (kHz).

The system uses a static-order task scheduler and the task scheduling is computed when the system is initialized. Therefore schedules are always fully executed and tasks in these schedule cannot be aborted. As a result, the task schedule of the next period can be influenced by the previous period if deadlines were not met, i.e., tasks had a longer execution time than expected. This behavior coincides, assuming that only one sample lag is possible, with the behavior described by scenario 2 in Section II-A.

Because ASML is able to derive a probability distribution for the completion time of the task schedule, we are able to use our presented methods and investigate how deadline misses affect the ASML system. For our industrial case study, we will focus on the short stroke of the waferstage with data losses in the c–a channel.

ASML has several performance criteria for their control systems, including overshoot and settling time in the time domain and gain margin, phase margin, and modules margin in the frequency domain [26, p. 151, 279]. Besides these common performance criteria, ASML also uses the moving average error (MA) as an important performance criterion. This is defined by $\text{MA}_k := \sum_{i=0}^{N-1} w_i e_{k-i}$, where N is the window size, w_i is the weight associated with the error e_{k-i} at discrete time $k \in \mathbb{N}$, and $i \in \{0, 1, \ldots, N-1\}$. The MA is only calculated during the photolithography process, e.g., during the exposure of a die. In this time span the error must be in the nanometer range. The short stroke performs repeated motions trajectories over the wafer. Fig. 11 shows the repeated motion trajectories of the short stroke and the window for the settling-time, the window in which the exposure of a die takes place and the window for which the calculated MA should be small, meaning that the MA is computed continuously, although its values only have importance in this window. By formulating the MJLS (8) from the reference to the error, we are able to compute the expected value of the error. Because the MA filter is linear, we are able to compute the expected moving average error, i.e., $\mathbb{E}[MA_k] = \mathbb{E}\left[\sum_{i=0}^{n-1} w_i e_{k-i}\right] = \sum_{i=0}^{n-1} w_i \mathbb{E}[e_{k-i}]$. The short stroke is modeled by an MIMO state-space system, which has 93 states and the controller has 82 states. Both have six inputs and six outputs. The linear controllers were designed based on loop-shaping techniques to meet the desired specifications. The frequency response of the open-loop and sensitivity transfer functions from the first input to the first output pertaining to the control of the x-axis are depicted in Fig. 12(a) and (b), respectively. The requirements for the sensitivity frequency response are also depicted in Fig. 12(b).

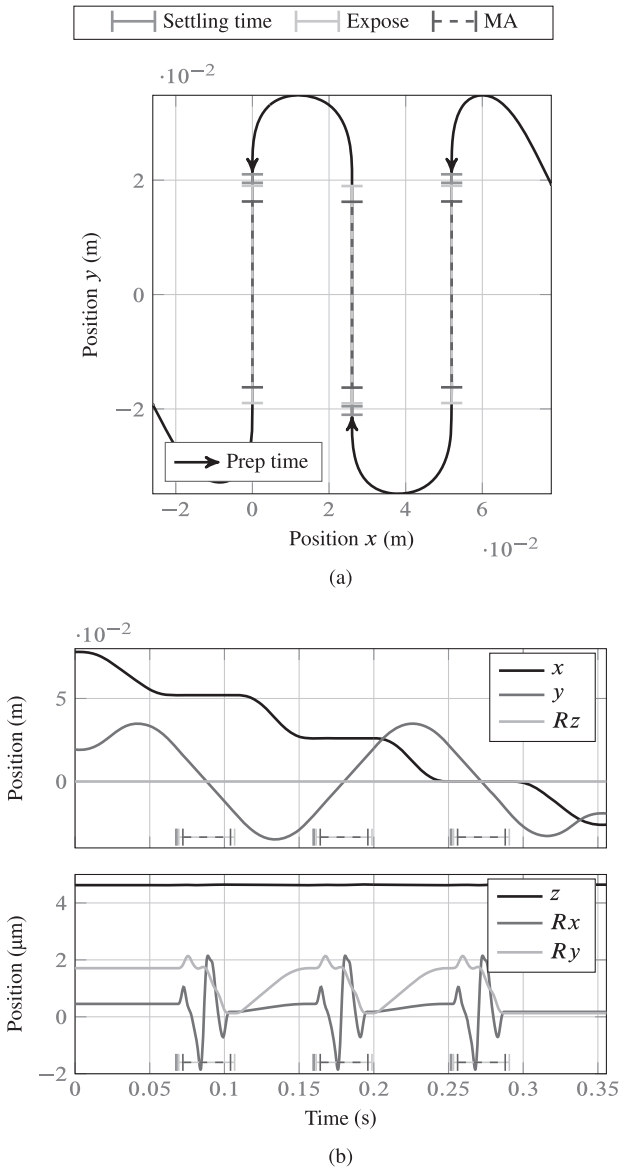

Fig. 11. Trajectory of the short stroke, along with the windows for the settling time (solid dark gray), exposure (solid light gray), and moving average (dashed). (a) Top view of the trajectory of the short stroke. (b) Trajectory for the six degrees of freedom of the short stroke.

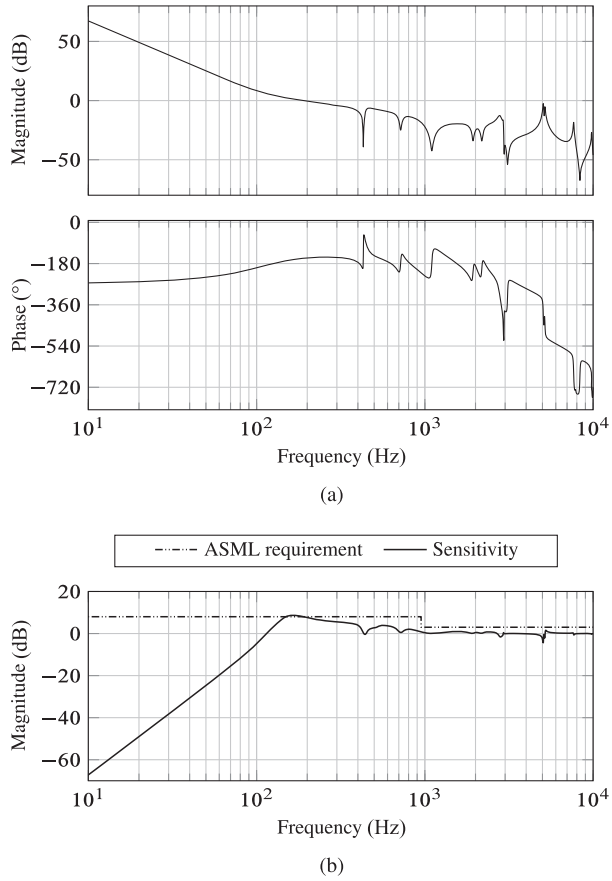

Fig. 12. Frequency response of the open-loop and sensitivity transfer function from the first input to the first output of the plant pertaining to the control of the degree of freedom x. (a) Frequency-response of the open-loop transfer function. (b) Frequency-response of the sensitivity transfer function.

Together with the six states required to model data losses in the c–a channel, this brings a total of 181 states of the MJLS resulting model. The system, which computes the expected output (9), has the same number of states, while the model, which computes the variance of the output (10), has $181 \times 181 = 32\,761$ states. Some elements from the state matrix of the plant A_p or the controller A_c are very small. This caused numerical problems when computing $\underline{A}$. As a result, we were not able to compute the variance of the system. No numerical problems were observed when computing the expected output.

The time-domain simulation of the short stroke consists of two separate simulations: 1) horizontal simulation in which only the x, y, and Rz axis have a setpoint and z, Rx, and Ry are set to zero; and 2) vertical simulation in which only the z, Rx and Ry axis have a setpoint and x, y, and Rz are set to zero. The vertical simulation pertains to the case where the system has to compensate for small height differences of the wafer. The simulations are performed separately to asses their contributions to the overall error. This is necessary because the horizontal movements affect the vertical movements and vice versa. In the discussion below, we will focus on the error caused by the horizontal displacement and only show the results of the x-axis for reasons of compactness.

Fig. 13(a) shows a simulation of the short stroke for the given trajectory along with the feedforward and the feedback controller but without any data losses. When comparing Fig. 13(a) and (b), observe that data losses in the feedback and feedforward controller combined are far more severe than data losses from the feedback controller only. When we focus on the control system without the feedforward controller [Fig. 13(b)], we can notice that in the plot with 50% data losses, the mean from the simulations and likewise the moving average error coincide with the analytically calculated mean and the analytically calculated moving average. The same results were observed when we had 100% data loss, i.e., when the system, without feedforward, lags one sample behind every period. We also investigated the behavior of the system when applying scenario 1, instead of scenario 2, in which case the system behaves very similarly for 50% data losses. Even at 80% data loss, the system behaves

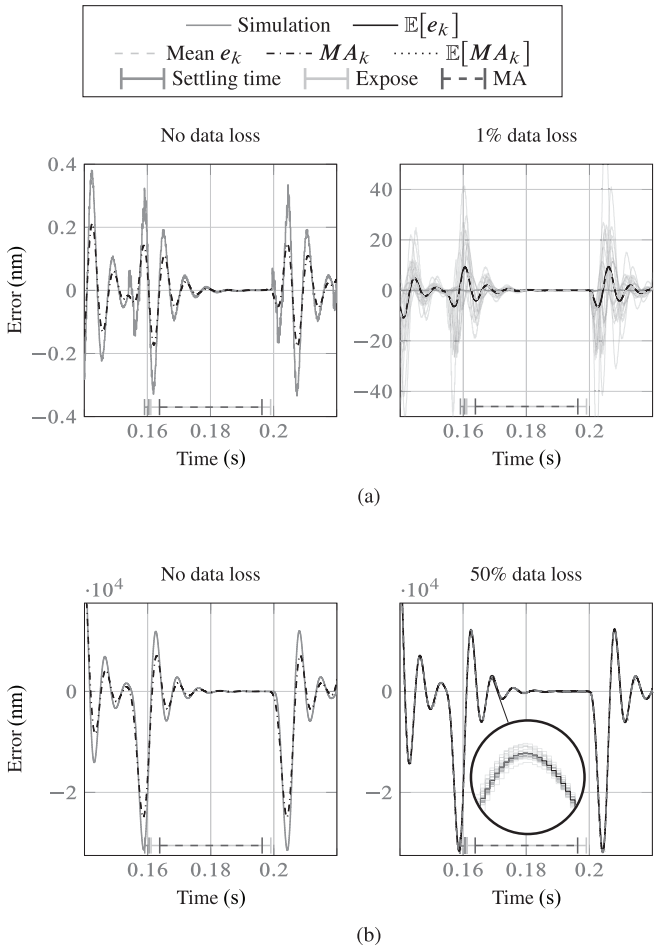

Fig. 13. Time response of the short stroke with and without feedforward controller and data losses modeled using scenario 2, respectively. (a) Simulation in which data losses occur in both the feedback and feedforward c–a channel. (b) Simulation in which data losses occur in the feedback c–a channel. No feedforward is present.

similarly. However, at approximately 85% data loss the system, exhibiting the behavior of scenario 1, becomes unstable.

From the results, we conclude that the performance of the short stroke is highly affected by data losses when they occur in the feedforward path. This was also verified by performing simulations in which only the feedback controller was affected by data losses and the feedforward was not. In fact, for both scenarios 1 and 2 with 50% data losses in the feedback controller, the behavior is very similar to that of the system without any data losses. Hence, the responses are hardly influenced by up to 50% data loss in the feedback path and no data losses in feedforward path. This may also lead to an approach on how to cope with deadline misses that causes data losses. Assuming that the feedforward is computed before the feedback, it would be possible to send the new feedforward with the old feedback to the actuator. When the deadline of the control tasks, which compute the feedback control, are met, the data will be overwritten by the new feedforward combined with the new feedback. If the deadline is missed, the IO board still has a control signal (using the precomputed feedforward value), which gives better performance than using merely the old control signal. In addition, because data losses hardly affect the feedback controller, it may be beneficial for ASML to do research in multirate control in which the feedback controller has a lower sampling frequency as opposed to the feedforward controller. This reduces the number of computations, which have to be carried out by the real-time platform, which would lead to less deadline misses and a cost reduction since fewer processors are needed.

In this section, we showed how our proposed methods apply to an industrial case study and how valuable information could be obtained. For instance, we were able to show that data losses with respect to the feedforward controller are far more critical to obtain nanometer precision than data losses with respect to the feedback controller.

VI. Conclusion

In this paper, we have presented an analysis and design framework for real-time control systems subject to data losses. The framework consists of 1) two basic models, to capture data losses as a consequence of deadline misses in a real-time control system; and 2) analysis techniques, which are used to analytically compute the mean and the variance of responses in the time domain of systems that are subject to data losses. We have demonstrated the usefulness of our methods on an experimental case study in which we performed both simulations and experiments. Furthermore, we applied them to an industrial case study in which we have determined that data losses in the feedforward controller are more severe than in the feedback controller. In the industrial case study, we proposed two possible solutions to cope with deadline misses that are identified as highly favorable.

In the future, we want to extend our methods to suit more general models. These include models for which Assumption 1 does not necessarily hold, models with delays (see, e.g., [31]–[33]), multirate sampling [31], [34], general nonlinear systems based on fuzzy dynamic models [35], and time-varying sampling periods [36].

References

[1] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

[2] Y. Jiang, Y. Zhu, K. Yang, C. Hu, and D. Yu, "A data-driven iterative decoupling feedforward control strategy with application to an ultra-precision motion stage," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 620–627, Jan. 2015.

[3] A. Cuenca, J. Salt, A. Sala, and R. Pizá, "A delay-dependent dual-rate PID controller over an ethernet network," *IEEE Trans. Ind. Electron.*, vol. 7, no. 1, pp. 18–29, Feb. 2011.

[4] V. Casanova, J. Salt, and A. Cuenca, "Irregular actuation and sampling in a networked control system over profibus-dp," in *Proc. 32nd Annu. Meeting IEEE Ind. Electron. Soc. (IECON'06)*, Nov. 2006, pp. 4616–4621.

[5] S. Adyanthaya, Z. Zhang, M. Geilen, J. Voeten, T. Basten, and R. Schiffelers, "Robustness analysis of multiprocessor schedules," in *Proc. Int. Conf. Embedded Comput. Syst. Archit. Model. Simul. (SAMOS XIV)*, 2014, pp. 9–17.

[6] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. New York, NY, USA: Springer, 2011, vol. 24.

[7] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proc. Amer. Control Conf.*, 2001, vol. 2, pp. 1491–1496.

[8] B. Lincoln and A. Cervin, "Jitterbug: A tool for analysis of real-time control performance," in *Proc. 41st IEEE Conf. Decis. Control*, 2002, vol. 2, pp. 1319–1324.
[9] S. C. Smith and P. Seiler, "Estimation with lossy measurements: Jump estimators for jump systems," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2163–2171, Dec. 2003.
[10] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.
[11] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.
[12] L. Schenato, "To zero or to hold control inputs with lossy links?" *IEEE Trans. Autom. Control*, vol. 54, no. 5, pp. 1093–1099, May 2009.
[13] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1807–1819, Aug. 2009.
[14] D. E. Quevedo and D. Nešić, "Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses," *Automatica*, vol. 48, no. 8, pp. 1803–1811, 2012.
[15] T. M. P. Gommans, W. P. M. H. Heemels, N. W. Bauer, and N. van de Wouw, "Compensation-based control for lossy communication networks," *Int. J. Control*, vol. 86, no. 10, pp. 1880–1897, 2013.
[16] H. Dong, Z. Wang, and H. Gao, "Distributed H∞ filtering for a class of Markovian jump nonlinear time-delay systems over lossy sensor networks," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4665–4672, Oct. 2013.
[17] B. Demirel, "Performance analysis of wireless networked control systems," Ph.D. dissertation, KTH Royal Inst. Technol., Stockholm, Sweden, 2015.
[18] R. Perrone, R. Macêdo, G. Lima, and V. Lima, "Estimating execution time probability distributions in component-based real-time systems," in *Proc. 10th Brazilian Workshop Real-Time Embedded Syst.*, 2008.
[19] D. Antunes and W.P.M.H. Heemels, "Frequency-domain analysis of control loops with intermittent data losses," *IEEE Trans. Autom. Control*, to be published.
[20] D. Antunes, W. Geelen, and W. Heemels, "Frequency-domain analysis of real-time and networked control systems with stochastic delays and drops," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 934–940.
[21] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*. New York, NY, USA: Springer, 2006.
[22] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, "How does control timing affect performance?" *IEEE Control Syst. Mag.*, vol. 23, no. 3, pp. 16–30, Jun. 2003.
[23] K. Seki, Y. Tsuchimoto, and M. Iwasaki, "Feedforward compensation by specified step settling with frequency shaping of position reference," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1552–1561, Mar. 2014.
[24] T. Gommans, "Nonlinear controller design for linear motion systems," B.S. thesis, Dept. Mech. Eng., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2008, p. 45.
[25] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," *Tech. Univ. Denmark*, 2012, p. 59 [Online]. Available: http://www.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf, accessed on Nov. 04, 2014.
[26] K. J. Aström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[27] J. Boot, M. J. G. van de Molengraft, and P. W. J. M. Nuij, "Frequency response measurement in closed loop: Brushing up our knowledge," Eindhoven Univ. Technology, Eindhoven, The Netherlands, 2003, [Online]. Available: http://www.dct.tue.nl/Pdf/FRF\s\do5(m)easurement.pdf, accessed on Nov. 4, 2014.
[28] M. Santina and A. Stubberud, "Discrete-time equivalents of continuous-time systems," in *The Control Handbook: Control Systems Fundamentals*, W. S. Levine, Ed., 2nd ed. Boca Raton, FL, USA: CRC Press, 2011, pp. 12.10, 12.23–12.24.
[29] *ASML Holding N.V.* [Online]. Available: http://www.asml.com, accessed on Nov. 5, 2014.
[30] M. Baggen, M. Heertjes, and R. Kamidi, "Data-based feed-forward control in MIMO motion systems," in *Proc. Amer. Control Conf.*, 2008, pp. 3011–3016.
[31] B. Zhang, K. Zhou, and D. Wang, "Multirate repetitive control for PWM DC/AC converters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2883–2890, Jun. 2014.
[32] D. Yashiro and T. Yakoh, "Feedback controller with low-pass-filter-based delay regulation for networked control systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3744–3752, Jul. 2014.
[33] A. Cuenca, J. Salt, P. Albertos, and V. Casanova, "Algebraic design of multi-rate control systems for environments with limited random delays," in *Proc. 32nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON'06)*, Nov. 2006, pp. 448–453.
[34] F. Liu, H. Gao, J. Qiu, S. Yin, J. Fan, and T. Chai, "Networked multirate output feedback control for setpoints compensation and its application to rougher flotation process," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 460–468, Jan. 2014.
[35] J. Qiu, G. Feng, and H. Gao, "Fuzzy-model-based piecewise h_∞ static-output-feedback controller design for networked nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 919–934, Oct. 2010.
[36] E. Kurniawan, Z. Cao, and Z. Man, "Design of robust repetitive control with time-varying sampling periods," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2834–2841, Jun. 2014.

W. Geelen was born in Weert, The Netherlands, in 1989. He received the Bachelor's degree in mechatronics from Fontys University of Applied Sciences, Venlo, The Netherlands, in 2013, and the Master's degree in systems and control from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2015.

Currently, he is a Mechatronics Consultant at Nobleo Technology, Eindhoven, The Netherlands, and working at ASML, Veldhoven, The Netherlands, within the Waferstage Control Group. His research interests include real-time digital control systems, the combination of embedded systems, and control theory.

D. Antunes (M'13) was born in Viseu, Portugal, in 1982. He received the Licenciatura degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2005, and the Ph.D. degree in automatic control from the Institute for Systems and Robotics, IST, in 2011.

From 2011 to 2013, he held a postdoctoral position with Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands. Currently, he is an Assistant Professor with the Department of Mechanical Engineering, TU/e. His research interests include networked control systems, stochastic control, dynamic programming, and systems biology.

J. P. M. Voeten received the Masters degree in mathematics and computing science and the Ph.D. degree in electrical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1997.

He is a Senior Scientist at TNO and an Associate Professor with the Electronic Systems Group, Faculty of Electrical Engineering, Eindhoven University of Technology. His research interests include system-level design, model-driven engineering, and performance modeling for cyber physical systems.

R. R. H. Schiffelers received the M.Sc. and Ph.D. degrees in mechanical engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 2001 and 2006, respectively.

He is a Postdoctoral Researcher and he has dealt with modeling, analysis, and synthesis of supervisory controllers for Philips MRI scanners and participated in European projects dealing with the design of the compositional interchange format for hybrid systems (CIF). In 2010, he joined ASML, the world's leading provider of lithography systems for the semiconductor industry. As a software innovation architect, he initiates, defines, and supervises several (multidisciplinary) research projects. Since 2012, he has been an Assistant Professor with the Model Driven Software Engineering Group, Faculty of Mathematics and Computer Science, Eindhoven University of Technology.

W. P. M. H. Heemels (SM'10) received the M.Sc. degree in mathematics and the Ph.D. degree in control theory (both *summa cum laude*) from Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, in 1995 and 1999, respectively.

After being an Assistant Professor with the Electrical Engineering Department, TU/e, and a Research Fellow with the Embedded Systems Institute (ESI), he became a Full Professor in the Control Systems Technology Group, Mechanical Engineering Department, TU/e. He held visiting research positions at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 2001, at Océ, Venlo, The Netherlands, in 2004, and at the University of California, Santa Barbara, CA, USA, in 2008. His research interests include general system and control theory, hybrid and cyber-physical systems, networked and event-triggered control, and constrained systems including model predictive control.

Prof. Heemels is an Associate Editor for *Nonlinear Analysis: Hybrid Systems* and *Automatica*. He was the recipient of a prestigious VICI grant from the Dutch Technology Foundation (STW) and The Netherlands Organization for Scientific Research (NWO). In addition, he served as the General Chair of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS) 2012, Eindhoven, The Netherlands, and was the IPC Chair for the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS) 2013, Koblenz, Germany.