2016 European Control Conference
June 29 - July 1, 2016. Aalborg, Denmark

Performance analysis and controller improvement for linear systems with (m,k)-firm data losses

E.P. van Horssen*, A.R. Baghban Behrouzian#, D. Goswami#, D. Antunes*, T. Basten#, W.P.M.H. Heemels*

***Abstract*— This paper describes methods for the analysis and design of control applications with real-time constraints, which allow data losses in the sensing-to-actuation path governed by the property of (m,k)-firmness. An automaton consisting of open- and closed-loop dynamics and a graph representing (m,k)-firmness defines the overall system behavior as a constrained switched linear system. The worst-case quadratic cost is analyzed for a given optimal linear quadratic regulator design. A simple analytic upper bounding method is compared to a method based on solving a (computationally more complex) semidefinite program. Furthermore, control design methods for performance improvement for the worst case are presented. A known LMI-based method is compared to an iterative controller improvement scheme inspired by ideas from dynamic programming. Conservatism and computational effort of the methods are discussed. A numerical example is used for illustration.**

I. Introduction

In order to limit cost, embedded systems are often used as multi-purpose platforms running several applications. This poses new challenges in the hardware and software design, e.g. scheduling of the application tasks when sharing resources such as processor time and memory [1]. In particular, the impact of design choices on control applications are an important subject for study, since safety and performance are likely to be affected.

In this paper, we study the performance of control applications, i.e. the Quality-of-Control (QoC), arising from sharing processor time with other applications, leading to a certain Quality-of-Service (QoS) [2] for the control task. In particular, the effect of deadline misses due to application scheduling is studied. The deadline misses, identifiable with dropouts or packet loss in a networked context, are assumed to adhere to the condition of (m,k)-firmness [3], i.e. in any k consecutive executions of the controller computation task, at least m meet the deadline.

We use tools for switched systems (e.g. [4]–[6]) to analyze the worst-case performance (QoC) expressed in terms of infinite-horizon quadratic cost and to design improved controllers achieving robust performance under certain QoS expressed through (m,k)-firm deadline misses [3].

The model of (m,k)-firmness without additional constraints has, to the authors' knowledge, not been considered for performance in terms of infinite-horizon quadratic cost in a switched systems setting. [7], [8] study the design of parameters (m,k) in the context of networked control. The work [9] aims to find acceptable structured schedules for a given performance constraint, while this work addresses analysis and control design for the unstructured (m,k)-firmness property.

For linear time-invariant systems, this work considers the implementation of a sampled-data [10] state feedback control law with fixed sampling interval on an embedded platform. The occurrence of deadline misses prohibits updating the control signal at each sampling instant, resulting in a discrete-time switched linear system with two types of dynamic modes (open- and closed-loop). The deadline miss model of (m,k)-firmness is shown to arise when verifying the scheduling of applications on a multi-purpose embedded platform. We use that sequences of deadline meets/misses under (m,k)-firmness can be captured by a graph, leading to a constrained switched linear system (cSLS) [5], [6], [9]. First, worst-case performance, in terms of infinite-horizon quadratic cost, is analyzed under deadline misses for an optimal controller [11] designed for the case without dropouts. Second, we give LMI-based and iterative procedures inspired by ideas from (approximate) dynamic programming [11]–[13] to synthesize controllers that guarantee improvement of the worst-case performance at the cost of optimality for the best case. The conservatism and the computational complexity of the methods are discussed.

In Section II, the embedded platform is detailed for which (m,k)-firmness is verified. The definition of the control system in Section III and the graph representing (m,k)-firmness in Section IV leads to the definition of the cSLS. The methods for performance analysis and improved controller design are presented and discussed in Section V and Section VI, respectively. Section VII concludes the work.

II. Multi-purpose platform

In this section, we give a motivating set-up of a multi-purpose hardware platform leading to the property of (m,k)-firmness on deadline misses for control tasks.

A. Resource allocation

We consider a processing unit which runs a Time Division Multiple Access (TDMA) scheduling policy (see e.g. [2]),

The authors are with the Control System Technology group(*), Dept. of Mechanical Engineering, and the Electronic Systems group(#), Dept. of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, e-mail: {e.p.v.horssen}@tue.nl

This research is supported by the ARTEMIS joint undertaking under the ALMARVI project (621439), by the Dutch Technology Foundation (STW) under grant number 12697 "Control based on Data-Intensive Sensing," and by the Innovational Research Incentives Scheme under the VICI grant "Wireless control systems: A new frontier in automation" (No. 11382) awarded by NWO (The Netherlands Organization for Scientific Research) and STW (Dutch Technology Foundation).

978-1-5090-2590-9 ©2016 EUCA

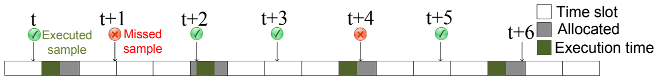

Fig. 1. Example of deadline misses for several consecutive TDMA periods where the second of four time slots is assigned to λ_C and $e_C = \frac{1}{2}T_{TS}$.

such that processing time is divided in N *time slots* (TSs) of equal duration T_{TS}. Any application running on the platform is assigned one or more TSs. Hence, the amount of resources assigned to an application $\lambda \in \Lambda$, where Λ is a set with n^Λ applications, depends on the size and number of the allocated TSs. In this work, the resource allocation is chosen a priori, leading to *static* schedule, i.e. each application has the same slots allocated each TDMA period.

B. Deadline misses

We focus here on a control application $\lambda_C \in \Lambda$ running on the platform. Sensing (reading sensors) and actuation (updating the actuation signals) tasks are performed periodically by dedicated hardware. In the processing unit, λ_C computes the new actuation signals. The execution of λ_C in the processing unit is determined by the tuple (e_C, d_C, T_C). Here, e_C is the execution time of λ_C, i.e. the duration the processor needs to be available to λ_C to compute a new control action based on the last measurement. The execution of λ_C should be completed by the deadline d_C, i.e. when the actuation signals are to be updated. The sampling interval T_C is the time between consecutive sensing moments (i.e. when sensors are read), which are indexed by $t \in \mathbb{N}$. It holds that $e_C \leq d_C$, otherwise all deadlines are missed. For simplicity, we assume that actuation signals are updated at the next sensing moment, i.e. $d_C = T_C$. (Results for $d_C < T_C$ can be derived analogously.) The resulting sensing-to-actuation delay of one sampling interval T_C is accounted for in Section III. The execution time e_C, the deadline d_C, the sampling interval T_C, and the occurrence of the sensing moments are independent of the resource allocation. Hence, it is unknown when sensing moments occur in the TDMA period.

For each interval $[t, t+1)$ with index $t \in \mathbb{N}$ and duration T_C, we say that *sufficient resources* are allocated to λ_C if the available processor time before d_C is larger than e_C, and *insufficient resources* otherwise. If the resources in the interval $[t, t+1)$ are insufficient, a *deadline miss* occurs. What happens to the actuation signals in the case of a deadline miss is discussed in Section III. Note that e_C is not restricted to be smaller than one time slot. Any sensing or actuation delay can be included in e_C, hence we can assume that these actions are instantaneous.

Figure 1 shows an example of deadline miss occurrence over multiple time wheels. The samples that arrived at time $t+1$ and $t+4$ do not have sufficient allocated resources before the next sensing moment to process and are therefore deadline missed samples (red crosses). The other samples are processed properly and meet the deadline (green check marks).

C. Verification of deadline misses

A task has the (m,k)-firmness guarantee if at least m out of k consecutive instances of the task meet their deadline.

Definition 1 ([1], [3]): A task on a processing unit is said to have an (m,k)*-firmness guarantee* with $m, k \in \mathbb{N}_{\geq 1}$ and $m \leq k$ if, for each $t \in \mathbb{N}$, at least m samples, with indices in the set $\mathbb{N}_{[t,t+k-1]}$, meet their deadlines.

In any window of k samples, the number m depends on the arrival time in the TDMA period of the first sample, which is unknown. The main verification challenge is to cover all possible arrival times of the first sample. Since the TDMA schedule repeats every N time slots, all possible arrival times of k consecutive samples can be covered by sliding the arrival time of the first sample over the entire TDMA period (i.e., from 0 to $N \times T_{TS}$). This verification problem can be addressed by standard model checking tools. The envisioned design approach lets an embedded control engineer iterate between a verification step for (m,k)-firmness of deadline misses and the methods proposed in this paper to achieve a desired control performance under a given resource restriction.

III. Feedback control problem

This section describes the control system behavior and its performance measure leading to the problem formulation. Furthermore, an example system is described.

A. Control System

We consider a continuous-time (CT) linear time-invariant (LTI) system

$$\tfrac{d\xi}{d\tau}(\tau) = A_C\xi(\tau) + B_C u(\tau), \qquad \tau \in \mathbb{R}_{\leq 0}, \tag{1}$$

where $\xi(\tau) \in \mathbb{R}^{n^\xi}$ is the state of the system and $u(\tau) \in \mathbb{R}^{n^u}$ the control input at time $\tau \in \mathbb{R}$. The control objective is to minimize an infinite-horizon quadratic cost criterion

$$J_C(\xi(0)) = \int_0^\infty \xi(\tau)^\top Q_C \xi(\tau) + u(\tau)^\top R_C u(\tau) d\tau. \tag{2}$$

We assume that (A_C, B_C) is controllable, B_C has full rank, $(A_C, Q_C^{\frac{1}{2}})$ is observable, and $R_C \succ 0$, which are standard assumptions. Note that $X \succ 0$ ($X \succeq 0$) means that X is symmetric and positive (semi-)definite.

We take a sampled-data approach (see e.g. [10]) and use a zero-order-hold control input $u(\tau) = u_{t-1}$, $\tau \in \mathbb{R}_{[tT_C,(t+1)T_C)}$, $t \in \mathbb{N}$, where u_{-1} is a given initial condition (typically $u_{-1} := 0$). The control input is updated at the next sampling instance, i.e. with a one-sample delay of T_C, as already mentioned in Section II-B. We can represent the system at the sampling instances $\tau = tT_C$, $t \in \mathbb{N}$, by a discrete-time (DT) system

$$x_{t+1} = Ax_t + Bu_t, \qquad t \in \mathbb{N}, \tag{3}$$

where $x_t = \left[\xi(tT_C)^\top, u_{t-1}^\top\right]^\top \in \mathbb{R}^{n^x}$, with $n^x = n^\xi + n^u$, consists of the sampled state $\xi(tT_C)$ of the CT system and the previous control input u_{t-1} and $u_t \in \mathbb{R}^{n^u}$ is the control input, and the initial state $x_0 = [\xi(0)^\top \; u_{-1}^\top]^\top$. Matrices A and B are obtained by exact discretization and by incorporating the one-sample delay using state augmentation. The cost (2) is exactly discretized [10, Sec. 11.1], giving

$$J(x_0) := \textstyle\sum_{t=0}^\infty x_t{}^\top Q x_t + 2x_t^\top S u_t + u_t^\top R u_t, \tag{4}$$

where $\begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \succeq 0$ is positive semi-definite. The sampling interval T_C is assumed to be non-pathological to preserve stabilizability and detectability in the discretization step.

The system is controlled by a (time-varying) state feedback of the form

$$u_t = -K_t x_t, \qquad t \in \mathbb{N}, \tag{5}$$

where $K_t \in \mathbb{R}^{n^u \times n^x}$ is a gain matrix that depends on the sampling instance.

B. Behavior under deadline misses

To incorporate deadline misses as discussed in the previous section, we choose that in case of a deadline miss $u_t = 0$. However, other approaches, such as holding the previous control value can also be taken in the same framework. As a consequence, the system can operate in two modes: The closed-loop mode $A_1(t) := A - BK_t$, and the open-loop mode $A_0(t) := A$. Hence, the dynamics can be represented by the switched system

$$x_{t+1} = A_{\sigma_t}(t) x_t, \qquad t \in \mathbb{N}, \tag{6}$$

where the active mode $\sigma_t \in \{1, 0\}$ is determined by the deadline miss sequence $\sigma : \mathbb{N} \to \{1, 0\}$ which satisfies the (m, k)-firmness condition.

The cost corresponding to each mode in (4) can be given as $Q_1(t) := Q + K_t^\top R K_t - SK_t - K_t^\top S^\top$, and $Q_0(t) := Q$, such that

$$J(x_0) = \textstyle\sum_{t=0}^{\infty} x_t^\top Q_{\sigma_t}(t) x_t, \qquad \sigma_t \in \{1, 0\} \tag{7}$$

is the cost for a particular dropout sequence σ.

C. Problem statement

The problem we consider is this paper is the following: For a control system, as detailed in Section III-A and III-B , that is affected by deadline misses with the (m, k)-firmness property, as detailed in Section II, i) quantify the (worst-case) performance loss in the presence of deadline misses for a given time-invariant control gain $K_t = K$, and ii) improve upon the (worst-case) performance by updating the control gain exploiting the (m, k)-firmness property. Parts i) and ii) of the problem are addressed in Section V and Section VI, respectively.

D. Example system

The methods in this paper are illustrated by means of an example system. We consider the unstable second-order mass-spring-damper system (1) with

$$A_C = \begin{bmatrix} 0 & 1 \\ -\frac{k_C}{m_C} & -\frac{b_C}{m_C} \end{bmatrix}, \qquad B_C = \begin{bmatrix} 0 \\ \frac{1}{m_C} \end{bmatrix}, \tag{8}$$

with force input with $m_C = 10, k_C = 1, b_C = -20$. The aim is to minimize cost (1) with

$$Q_C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_C = \begin{bmatrix} 1 \end{bmatrix}. \tag{9}$$

We use a sampling interval $T_C = 7$ms and a one-sample actuation delay resulting in a system of the form (3) with $n^x = 3$ and $n^u = 1$, and cost of the form (4). The details are omitted for brevity.

IV. (M,K)-FIRMNESS AUTOMATON MODEL

In this section, we propose a systematic way to model the (m, k)-firmness property for control purposes. We show, using standard graph theory, that all sequences of deadlines meets/misses of infinite length that satisfy the property of (m, k)-firmness can be represented by a directed graph $\mathcal{G}$. (see, e.g., [3, Fig. 2]) [14, Fig. 1-3] for simple examples). We represent a sequence σ by a binary string. Note that, under a given (m, k)-firmness guarantee, having a deadline miss gives us information about which sequences can occur, i.e. if a deadline miss occurs, then one less deadline miss can occur in the subsequent $k - 1$ computations. Let the set of subsequences of at most length k (e.g. '1', '0', '00', '01', '01101', etc.) that can occur under a given (m, k)-firmness, i.e. with at most m zeros, be denoted by Σ, with $|\Sigma| =: n^\Sigma$, where $|X|$ denotes the cardinality of a set or vector. All infinite sequences that satisfy a given (m, k)-firmness can then be generated by transitions between elements of Σ, where a transition is found by concatenating a '1' or '0' to the subsequence. We note that (in language-equivalent sense) some subsequences and transitions are the same, hence we wish to reduce the number of subsequences. For any (m, k)-firmness condition, a minimal (in language-equivalent sense) labeled directed graph $\mathcal{G} := (V, E)$ can be systematically generated by tools such as [15]. The graph $\mathcal{G}$ is defined by the set $V := \{v_1, v_2, \ldots, v_{n^V}\} \subseteq \Sigma$ of vertices $v_i \in \Sigma$ for $i = 1, 2, \ldots, n^V$, and the set $E := \{e_1, e_2, \ldots, e_{n^E}\} \subseteq \Sigma \times \Sigma$ of directed edges $e_j \in V \times V$ for $j = 1, 2, \ldots, n^E$. An edge from vertex v_j to v_j is denoted by (v_j, v_j). The vertices represent the subsequences that can occur and the edges determine which transitions (deadline miss occurrences) between subsequences are allowed. For such a minimal graph $\mathcal{G}$, n^V is typically much smaller than n^Σ. In this work, we select the vertex labeled with '1' as the initial node v_1, corresponding to the case where no dropouts have occurred in the last k instances. An automaton is generated for $(m = 5, k = 7)$ and visualized in Figure 2. Note that n^V and n^E may increase rapidly for increasing values of k.

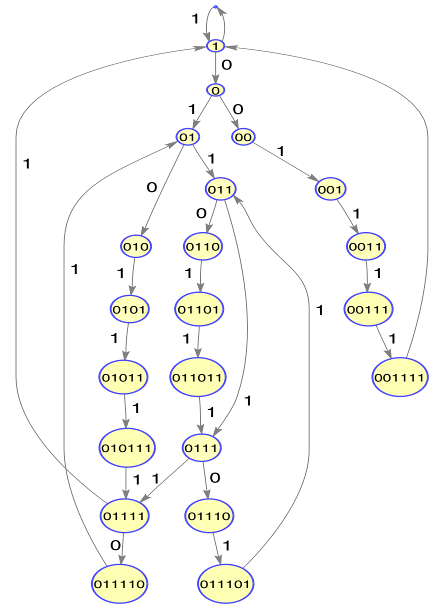

Fig. 2. Computer-generated automaton model for $(m = 5, k = 7)$ with 21 vertices and 27 edges. The markers on the vertices indicate the occurred subsequence and the markers on the edges determine the active mode.

Note that the incoming edges $(v_i, v_j) \in E$ correspond to the same dynamic mode '1' or '0' for each node $v_j \in V$. Therefore, we use the notation A_{v_j} and let each vertex $v_j \in V$ correspond to the dynamic mode of the incoming edges. Furthermore, we denote by E_1 and E_0 the edges $(v_i, v_j) \in E$ for which the dynamics on vertex v_j correspond to A_1 and A_0 in (6), respectively. The directed graph $\mathcal{G}$, together with dynamics (6) corresponding to '1' or '0' on the vertices, defines a hybrid automaton or, more specific, a constrained switched linear system (cSLS) [5], [6], [9]. This system model, for any given (m, k)-firmness property, will be used to obtain the results in the subsequent sections.

V. Performance analysis

Addressing part i) of the problem statement in Section III-C, this section describes methods to compute the performance loss, in terms of (4), when a system is affected by deadlines misses with the (m, k)-firmness property. The bounds are computed for an optimal linear quadratic regulator (LQR) design (for the case without deadline misses). An LMI-based method is compared to a closed-form analytic expression.

A. Optimal control solution

In the remainder of Section V, we take a standard design approach and assume that $K_t = K$ for all $t \in \mathbb{N}$, where K is the result of the optimal control design procedure for the linear quadratic regulator (LQR) of minimizing (4) subject to (3) without deadline misses. More specifically, $K := (R + B^\top PB)^{-1}(B^\top PA + S^\top)$ where P follows from the solution to the discrete-time algebraic Riccati equation (DARE) $P = A^\top PA + Q - (B^\top PA + S^\top)^\top (R + B^\top PB)^{-1}(B^\top PA + S^\top)$. Then, we have that $A_1(t) = A - BK$ in (6) and $Q_1(t) := Q + K^\top RK - SK - K^\top S^\top$ in (7) are time-invariant. For a given initial value x_0, a lower bound on the cost is given by $J^\star(x_0) = x_0^\top Px_0$, corresponding to the case when no deadline misses occur.

We are interested in finding an expression for the bound $\overline{J}(x_0)$ for which it holds that

$$\overline{J}(x_0) \geq \max_{\sigma^{\mathcal{G}}} \sum_{t=0}^{\infty} x_t^\top Q_{\sigma_t} x_t, \quad \text{for all } x_0 \in \mathbb{R}^{n^x}, \quad (10)$$

where $\sigma^{\mathcal{G}}$ denotes the set of infinite sequences on $\mathcal{G}$ starting from vertex '1'. With the intention of using the bound in online switching conditions [12], [13], we aim to find a compact expression for $\overline{J}(x_0)$, which can be evaluated easily.

B. Upper bound by LMI-based method

Since the (m, k)-firmness can be modeled as a cSLS (see Section IV), we can use known methods to give LMI-based conditions for stability [16], [17].

Associate a quadratic storage function $S_i(x) := x^\top P_i x$, $P_i \succ 0$ to each vertex $v_i \in V$, for $i = 1, 2, \ldots, n^V$. Constraints are imposed for all edges in E in the form of LMIs.

Theorem 1: The system (3), with cost measured by (4), and with feedback (5) with controller gain K, and with the property of (m, k)-firmness on deadline misses governed by an automaton with labeled graph $\mathcal{G} = (V, E)$ as described in Section IV, is globally exponentially stable if there exist $P_i \succ 0$, for all $v_i \in V$, such that the conditions

$$P_i \succeq Q_{v_j} + A_{v_j}^\top P_j A_{v_j}, \qquad \text{for all } (v_i, v_j) \in E, \quad (11)$$

are satisfied. Furthermore, an upper bound on the cost as in (10) is given by $\overline{J}(x_0) = x_0^\top P_{root} x_0$ where $root \in V$ corresponds to the initial node of the automaton.

The proof is standard, and is therefore omitted for brevity.

We can use Theorem 1 to compute the worst case loss w.r.t. the optimal cost without deadline misses $J^\star(x_0) = x_0^\top Px_0$.

<u>Method 1 (M1): LMI-based analysis</u>

Solve the semidefinite program (SDP)

$$\min_{P_i \succ 0, \text{ for all } v_i \in V} \alpha$$

subject to

$$P_1 \preceq \alpha P,$$

$$P_i \succeq Q_1 + A_1^\top P_j A_1, \qquad \text{for all } (v_i, v_j) \in E_1, \quad (12)$$

$$P_i \succeq Q_0 + A_0^\top P_j A_0, \qquad \text{for all } (v_i, v_j) \in E_0, \quad (13)$$

for P the solution to the DARE in Section V-A and $\alpha \in \mathbb{R}$.

M1 provides global asymptotic stability by mode-dependent Lyapunov functions whilst minimizing an upper bound on the performance. In particular, α gives a scaling of the performance compared to the optimal LQR cost.

Although M1 provides a solution $\overline{J}(x_0)$ to (10), computing it requires a significant computational effort. One way to address this is by replacing all P_i in M1 by $\alpha_i P$, limiting the number of variables at the cost of introducing conservatism. In the following section, we provide an alternative method to obtain an upper bound as in (10) in the form of an analytic expression.

C. Upper bound by analytic expression

Now, we derive an analytic expression of the form

$$\overline{J_a}(x_0) = g \cdot x_0^\top Px_0, \quad (14)$$

with constants $g \in \mathbb{R}_{\geq 0}$ and $P \in \mathbb{R}^{n^x \times n^x}$, $P \succ 0$, as an upper bound as in (10).

Theorem 2: The system (3), with cost measured by (4), and with feedback (5) with controller gain K, and with the property of (m, k)-firmness on deadline misses governed by an automaton with labeled graph $\mathcal{G} = (V, E)$ as described in Section IV, is globally exponentially stable and has an upper bound on the cost of the form (14) satisfying (10), with

$$g := r^\star \cdot \beta \cdot \left(\tfrac{1}{1-\rho}\right), \quad (15)$$

$$r^\star := \min\left\{r \in \mathbb{R}_{\geq 0} \mid Q_1 \preceq rP \text{ and } Q_0 \preceq rP\right\}, \quad (16)$$

$$\beta := \sum_{i=1}^{k-m} \rho_0^{(i-1)} + \rho_0^{(k-m)} \cdot \sum_{i=1}^{m} \rho_1^{(i-1)}, \quad (17)$$

$$\rho := \rho_0^{(k-m)} \rho_1^m, \quad (18)$$

if there exist $P \succ 0$ and $\rho_1, \rho_0 \in \mathbb{R}_{\geq 0}$, with $\rho_1 \leq \rho_0$, that satisfy

$$\rho < 1, \qquad \rho_1 P \succeq A_1^\top PA_1, \qquad \rho_0 P \succeq A_0^\top PA_0, \quad (19)$$

for some $P \succ 0$.

Proof: Since by (19) we have that

$$x_t^\top Q_{\sigma_t} x_t \leq r^\star \cdot x_t^\top Px_t, \qquad \text{for all } t \in \mathbb{N},$$

$$x_{j\cdot k}^\top Px_{j\cdot k} \leq \rho^j \cdot x_0^\top Px_0, \qquad \text{for all } j \in \mathbb{N},$$

$$\sum_{t=j\cdot k}^{j\cdot k+(k-1)} x_t^\top Px_t \leq \beta \cdot x_{j\cdot k}^\top Px_{j\cdot k}, \qquad \text{for all } j \in \mathbb{N}.$$

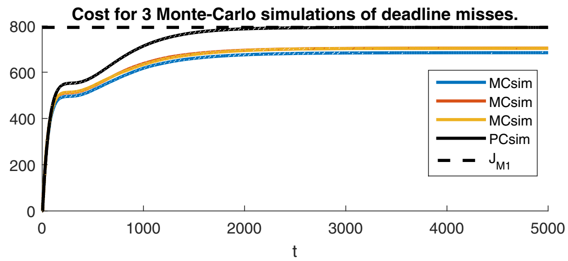

Fig. 3. Cumulative cost for the Monte-Carlo (MC) simulations (color), the periodic case (PC) 0011111 (black), and the bound given by M1.

By $\rho < 1$, the state converges exponentially. The limit of the sum over all $j \in \mathbb{N}$ can be computed by

$$\sum_{j=0}^{\infty} \rho^j = \left(\frac{1}{1-\rho}\right),$$

which is a converging geometric series.

Combining these (in-)equalities gives (14), with g as in (15) for P that satisfies (19), which satisfies (10). ■

Theorem 2 gives a bound based on the k-step convergence rate of the state, whilst bounding the cost for the underlying steps. The current result is open to improvement.

The bound is generated by several steps that introduce conservatism. We note that this is the case for other bounds in literature as well. For instance, in [4, Lemma 4], a bound was given, requiring the assumption $||x_t||_2^2 \leq b \cdot a^t \cdot ||x_0||_2^2$. Our method can allow relaxation of b and a.

We summarize the use of Theorem 2 in the following method.

Method 2 (M2): Evaluate analytic expression

Find ρ_1, ρ_0, P such that (19) holds. Use for P the solution to the DARE in Section V-A. Simple techniques can be used to find (upper bounds on) the parameters ρ_1, ρ_0 (see e.g. [18]). If the conditions for Theorem 2 hold, compute $\overline{J_a}(x_0)$ using the expression in Theorem 2.

Since M2 only requires simple computations, it is very flexible to variations in (m, k). However, the condition $\rho < 1$ in (19) might be very restrictive. Nevertheless, the result can be useful if the condition is satisfied, as it is simple to obtain.

D. Comparison of M1 and M2

To compare M1 and M2 numerically, the system in Section III-D is used. We choose $(m = 5, k = 7)$, such that the automaton in Figure 2 applies, and compute a solution to M1. We choose '1' as root node and run Monte-Carlo simulations for 50 randomly generated deadline miss sequences (using a 50% deadline miss probability) of the system for $t = 1, \ldots, 5000$ with initial condition $x_0 = [1, 1, 0]^\top$. Additionally, we run a simulation with periodic repeating deadline miss pattern 0011111 as an illustrative case. All computations in this work have been carried out in Matlab by using freely available software [19], [20]. The cumulative cost for all simulations is displayed in Figure 3.

Additionally, we compute the bound $\overline{J}(x_0) = 793.900$ by M1. As expected, there is a significant loss compared to the LQR cost $J^\star(x_0) = 470.929$. The bound is depicted in Figure 3 as well. We see, as expected, that M1 provides a good bound on the worst-case performance.

Let $\lambda_M(X)$ and $\lambda_m(X)$ denote the largest and smallest eigenvalue of a symmetric matrix X, respectively. For comparison, we use $\rho_1 = 1 - \lambda_m(Q_1)/\lambda_M(P) = 0.999983$ and $\rho_o = 1 + 2\sqrt{\lambda_M(A_0^\top A_0)} = 3.029569$ [18]. We find that $\rho = 9.1775$ and we cannot compute $\overline{J_a}(x_0)$ for $(m = 5, k = 7)$. Note that the bound does hold for very large values of m and k. Other methods such as the (constrained) joint spectral radius [6], [21] may be helpful, although these methods require quite some computational effort as well.

VI. Robust Switched Controller Design

In this section, we adapt the controller of Section V-A to improve robustness of performance under deadline misses, thereby addressing part ii) of the problem statement in Section III-C. The resulting controller is a switched controller with a gain that depends on the vertex of an automaton as described in Section IV, i.e. the restriction $K_t = K$ in Section V-A is relaxed. First, an LMI-based procedure is presented based on known results for (c)SLS. Second, an iterative procedure is presented, inspired by (approximate) dynamic programming and rollout [11]–[13]. It uses the SDP of M1 and a separate procedure to update the control gains. Advantages and disadvantages of both methods are discussed.

A. LMI-based control design

This procedure is based on the matrix inequality

$$P_i \succeq Q - SK_{ij} - K_{ij}^\top S^\top + K_{ij}^\top RK_{ij} \quad (20)$$
$$+ (A - BK_{ij})^\top P_j (A - BK_{ij}), \text{ for all } (v_i, v_j) \in E,$$

for P_i, P_j, K_{ij} unknown, which is the synthesis counterpart of the condition (11) in Theorem 1. Conditions for mode-dependent controller synthesis with improved performance guarantee are given in the following theorem.

Theorem 3: The conditions of Theorem 1 are satisfied, with $A_{v_j} = A - BK_i$, for all $(v_i, v_j) \in E_1$, and $A_{v_j} = A$, for all $(v_i, v_j) \in E_0$, if there exist $X_i \succ 0$, and $Y_i \in \mathbb{R}^{n^u \times n^x}$, for all $v_i \in V$, such that the conditions

$$\begin{bmatrix} X_i & (AX_i - BY_i)^\top & \begin{bmatrix} X_i \\ -Y_i \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \\ (AX_i - BY_i) & X_j & 0 \\ \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} X_i \\ -Y_i \end{bmatrix} & 0 & \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \end{bmatrix} \succeq 0,$$
$$\text{for all } (v_i, v_j) \in E_1, \quad (21)$$

$$\begin{bmatrix} X_i & (AX_i)^\top & X_i Q \\ (AX_i) & X_j & 0 \\ QX_i & 0 & Q \end{bmatrix} \succeq 0, \quad \text{for all } (v_i, v_j) \in E_0, \quad (22)$$

are satisfied. Hence, $u_t = -K_i x_t$ if the system is at vertex v_i and no deadline miss occurs for sample t, i.e. for transition (v_i, v_j) with v_j a closed-loop vertex. The controller gains K_i can be computed by

$$P_i = X_i^{-1}, \qquad K_i = Y_i P_i.$$

Proof: By standard matrix manipulations, i.e. Schur's complement and methods from [22] and [23], the conditions (21) and (22) are equal to (20). Details are omitted for brevity. ■

Starting at the root node, we can evaluate Theorem 3 by the following method.

Method 3 (M3): LMI-based synthesis

Solve the SDP

$$\min_{X_i \succ 0,\ Y_i,\ \text{for all } v_i \in V} -\alpha \quad \text{subject to} \quad X_1 \succ \alpha P^{-1}, (21), (22),$$

for $\alpha \in \mathbb{R}$, which guarantees $P_1 \preceq \alpha^{-1}P$, where P is the solution to the DARE in Section V-A.

Note that M3 provides a direct solution $\overline{J}(x_0) = x_0^\top X_1^{-1} x_0$ to (10), as well as a mode-dependent controller. However, the computational effort required is even larger than M1 as the number of variables is increased. As suggested for method M1, one may also replace all X_i in M1 by $\alpha_i P^{-1}$ to limit the number of variables at the cost of introducing conservatism. In the following section, we propose an iterative procedure to approach a similar result to M3, which may be computationally more attractive.

B. Iterative controller redesign

This section details a scheme for updating the controller gains of Section V-A in M1 by using the information gained when a deadline miss occurs. In particular, the procedure builds on the result of M1 in the sense that we use the mode-dependent P_i, for all $v_i \in V$ as a starting point.

The conditions (12) and (13) come from the desire to give a bound $x_t^\top P_i x_t$ on the sum of the stage cost $x_t{}^\top Q x_t + 2x_t^\top S u_t + u_t^\top R u_t$ and the upper bound on the cost-to-go $x_{t+1} P_j x_{t+1}$, where $x_{t+1} = Ax_t + Bu_t$ and P_j depends on the mode σ_t. From optimal control arguments [11], we know that the optimal solution to

$$\min_{u_t} x_t{}^\top Q x_t + 2x_t^\top S u_t + u_t^\top R u_t + x_{t+1} P_j x_{t+1}$$

is given by (5), where

$$K_t = (R + B^\top P_j B)^{-1}(B^\top P_j A + S^\top), \qquad (23)$$

if the system is in mode $v_i \in V$ at time t. Let the n^V gains corresponding to each $v_i \in V$, and computed with P_j, be denoted by K_i for all $(v_i, v_j) \in E_1$. Then, by optimality

$$P_i \succeq Q_1 + A_1^\top P_j A_1 \succeq Q_1(t) + A_1(t)^\top P_j A_1(t) \qquad (24)$$

for $A_1(t)$ and $Q_1(t)$ as in Section III-B with (23). Therefore, possibly there exists some $P_i^{new} \succ 0$ that satisfies

$$P_i \succ P_i^{new} \succeq Q_1(t) + A_1(t)^\top P_j A_1(t),$$
$$P_i \succ P_i^{new} \succeq Q_0 + A_0^\top P_j A_0.$$

Finding P_i^{new} can be formulated as a new SDP. When formulated for all vertices $v_i \in V$, it boils down to using $M1$, but now with (12) replaced by

$$P_i \succeq Q_{1,i} + A_{1,i}^\top P_j A_{1,i}, \qquad \text{for all } (v_i, v_j) \in E_1, \quad (25)$$

for $Q_{1,i}$ and $A_{1,i}$ as in Section III-B with K_i. Repeating this procedure gives a stepwise improvement of the performance bound. The procedure is summarized in an algorithm in the following method.

Method 4 (M4): Iterative controller redesign

Solve LQR to find K and initial A_1 and Q_1, for all $v_i \in V$.

Algorithm 1:

1) Use M1 to compute P_i, for all $v_i \in V$. (assuming that it is feasible initially)
2) Compute $K_i^{new} = (R + B^\top P_j B)^{-1}(B^\top P_j A + S^\top)$, for all $(v_i, v_j) \in E_1$.
3) Update $A_{1,i}$ and $Q_{1,i}$ by K_i^{new} for each i.
4) Repeat from step 1) with $A_{1,i}$ and $Q_{1,i}$, until P_1 in M1 has converged or is satisfactory.

An upper bound on the performance is given by $\overline{J}(x_0) = x_0^\top P_1 x_0$ after the last iteration.

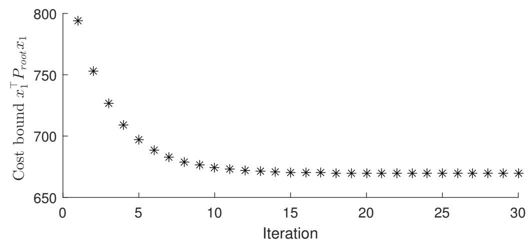

Fig. 4. Cost bound for iterations of M4.

In each iteration of M4, the solution is improved whilst stability is guaranteed.

Theorem 4: If there exists a solution to M1, then a solution to Algorithm 1 exists, and the mode-dependent controllers obtained in each step globally exponentially stabilize the system and the cost bound defined by P_1 is non-increasing.

Proof: Directly from arguments in this section. ■

Note that compared to M3, it is only needed to solve a mode-dependent version of M1, but no controller variables are involved. Hence, the computational expense in each iteration is less than that of M3. Both M4 and M3 require the separate computation of the control gains by inverse matrix multiplication. M4 has to solve the SDP in step 1) and compute all controller matrices in step 2) in each iteration, which also requires quite some computational effort. However, we see that the computation of the controller matrices can be parallelized for all $v_i \in V$, allowing for efficient implementation in multi-processor platforms. Furthermore, Algorithm 1 can be stopped after each iteration if the computation time is limited and would still yield an improved solution compared to M1. It is expected that, in general, the first iteration step provides the most benefit. M4 requires the solution to M1 to exist, which can be restrictive. However, any stabilizing solution to M1, i.e. any K, can be used as an initial condition for M4. M4 is found to achieve comparable results to M3, as is shown for an example in the next section.

C. Comparison of M3 and M4

We use the same simulation conditions as in Section V-D for comparison of M3 and M4. We run M4 with 30 iterations of Algorithm 1. In Figure 4, the convergence of the cost bound for M4 is shown. We see that in the first few iterations the performance improvement is most significant and after about 15 iterations it has almost converged.

The performance bounds are found to be $\overline{J_{M3}}(x_0) = 669.581$ for M3 and $\overline{J_{M4}}(x_0) = 669.587$ for M4, i.e. they are practically equal. Hence, a (significant) performance gain of 15.7% is achieved compared to the bound that was found for M1. The controller gains of M4 are very similar to those of M3 and converging further for increasing number of iterations of Algorithm 1, M4 seems to converge to the solution of M3.

The bound $\overline{J}(x_0)$ for M4 is depicted in Figure 5. Again, we use Monte-Carlo simulations to illustrate the results. We show the states and the cost for three randomly generated deadline miss sequences and the periodic '0011111' sequence. The cumulative cost for M4, which are similar to M3, is displayed in Figure 5. The costs satisfy the bounds

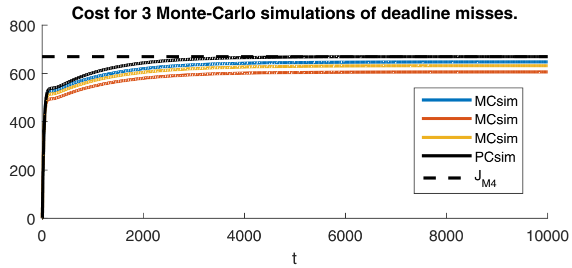

Fig. 5. Cumulative cost for the Monte-Carlo (MC) simulations (color) and the periodic case (PC) 0011111 (black) and the bound for M4 (dashed).

as expected. We observed that if the state cost is high, the M3/M4 controller is more aggressive than the LQR controller. It aims to reduce the states before a deadline miss occurs. This incurs a higher initial cost, but pays off (if a deadline miss occurs) in the stage cost later. Upon a deadline miss, the controller is relaxed, thereby reducing the cost.

We can quantify the performance loss compared to the LQR solution when no deadline misses occur (the "best case"). For this, we can solve M1 for P_{root} with $P_{root} \succeq Q_{c,root} + A_{c,root}^\top P_{root} A_{c,root}$ as the only constraint. We take '1' as root and find $x_0^\top P_1 x_0 = 476.320$ for M3 and $x_0^\top P_1 x_0 = 488.118$ for M4, corresponding to a performance decrease of 1% and 3.5% w.r.t. $J^\star(x_0)$, respectively. The difference comes from the incomplete convergence of M4. Hence, we can conclude that the trade-off between performance degradation for the best case and robustness of performance for the worst case is useful and beneficial.

While M3 is the most complete method, i.e. it guarantees exponential stability, an upper bound on the performance, and a controller synthesis method, it suffers from an increase in complexity proportional to the size of the (m, k)-firmness automaton. Method M4 achieves similar results to M3, but builds on the feasibility of M1. A trade-off in computational effort and performance improvement can be made by choosing the number of iterations of Algorithm 1.

VII. Conclusion

In this paper, we present methods that can be used as the control analysis and design counterparts of scheduling problems in embedded system design. This gives quantitative insights in the trade-offs between QoC (in terms of quadratic cost) and QoS (expressed by (m, k)-firmness). We show that control systems affected by data loss governed by (m, k)-firmness can be modeled as a constrained switched linear system. Performance loss due to the deadline misses is analyzed and compared to the optimal case (without deadline misses). A known LMI-based tool is used, which is computationally intensive. It is compared to an analytic expression which is, however, significantly more conservative, but allow possible improvement. Furthermore, LMI-based and iterative methods for controller improvement are shown to achieve similar performance for varying computational effort. Compared to the LMI-based method, the iterative method allows for a trade-off of QoC improvement against computational effort.

References

[1] M. Kauer, D. Soudbakhsh, D. Goswami, S. Chakraborty, and A. M. Annaswamy, "Fault-tolerant control synthesis and verification of distributed embedded systems," in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, 2014, pp. 1–6.

[2] B. Akesson, A. Minaeva, P. Sucha, A. Nelson, and Z. Hanzalek, "An efficient configuration methodology for time-division multiplexed single resources," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2015 IEEE*, April 2015, pp. 161–171.

[3] M. Hamdaoui and P. Ramanathan, "A dynamic priority assignment technique for streams with (m,k)-firm deadlines," *Computers, IEEE Transactions on*, vol. 44, no. 12, pp. 1443–1451, Dec 1995.

[4] W. Zhang, A. Abate, J. Hu, and M. P. Vitus, "Exponential stabilization of discrete-time switched linear systems," *Automatica*, vol. 45, no. 11, pp. 2526–2536, Nov. 2009.

[5] M. Fiacchini, A. Girard, and M. Jungers, "On the stabilizability of discrete-time switched linear systems: Novel conditions and comparisons," *Automatic Control, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[6] M. Philippe, R. Essick, G. Dullerud, and R. Jungers, "Stability of discrete-time switching systems with constrained switching sequences," 2015, arXiv preprint arXiv:1503.06984.

[7] N. Jia, Y.-Q. Song, and R.-Z. Lin, "Analysis of networked control system with packet drops governed by (m,k)-firm constraint," in *6th IFAC international conference on fieldbus systems and their applications (FeT'2005)*, Puebla, Mexico, Nov. 2005.

[8] F. Felicioni, N. Jia, Y.-Q. Song, and F. Simonot-Lion, "Impact of a (m,k)-firm Data Dropouts Policy on the Quality of Control," in *6th IEEE International Workshop on Factory Communication Systems*. Torino, Italy: IEEE, June 2006, pp. 353–359.

[9] G. Weiss and R. Alur, "Automata based interfaces for control and scheduling," in *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2007, pp. 601–613.

[10] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*. Courier Corporation, 2013.

[11] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005, 3rd Edition. Vol. 1 and Vol. 2.

[12] D. Antunes and W. Heemels, "Performance analysis of a class of linear quadratic regulators for switched linear systems," in *IEEE Conference on Decision and Control (CDC) 2014, Los Angeles, USA*, December 2014, pp. 5475–5480.

[13] ——, "Rollout event-triggered control: Beyond periodic control performance," *IEEE Transactions on Automatic Control*, vol. 59, pp. 3296–3311, August 2014.

[14] D. Liu, X. Hu, M. Lemmon, and Q. Ling, "Firm real-time system scheduling based on a novel QoS constraint," in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, Dec 2003, pp. 386–395.

[15] D. A. van Beek, W. Fokkink, D. Hendriks, A. Hofkamp, J. Markovski, J. M. van de Mortel-Fronczak, and M. A. Reniers, "CIF 3: Model-based engineering of supervisory controllers," in *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014. Proceedings*, vol. 8413. Springer, 2014, pp. 575–580.

[16] J. Daafouz, P. Riedinger, and C. Iung, "Stability analysis and control synthesis for switched systems: a switched lyapunov function approach," *Automatic Control, IEEE Transactions on*, vol. 47, no. 11, pp. 1883–1887, 2002.

[17] J. C. Geromel and P. Colaneri, "Stability and stabilization of discrete time switched systems," *International Journal of Control*, vol. 79, no. 07, pp. 719–728, 2006.

[18] A. Kundu and D. Chatterjee, "Stabilizing discrete-time switched linear systems," in *17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'14, Berlin, Germany, April 15-17, 2014*, 2014, pp. 11–20.

[19] J. Lofberg, "Yalmip: a toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, 2004, pp. 284–289.

[20] J. F. Sturm, "Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999.

[21] R. Jungers, *The joint spectral radius: theory and applications*. Springer Science & Business Media, 2009, vol. 385.

[22] J. Bernussou, P. Peres, and J. Geromel, "A linear programming oriented procedure for quadratic stabilization of uncertain systems," *Systems & Control Letters*, vol. 13, no. 1, pp. 65 – 72, 1989.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM, June 1994, vol. 15.