Symbolic Abstractions for the Scheduling of Event-Triggered Control Systems

Arman Sharifi Kolarijani, Dieky Adzkiya and Manuel Mazo, Jr.

Abstract— **In this paper, the problem of scheduling event-triggered networked control systems sharing a communication channel is addressed. Event-triggered control strategies effectively reduce the usage of resources in the implementation of control loops, in particular communication bandwidth. However, there is a lack of a well-established framework to analyze their corresponding communication load and synthesize schedulers. We focus on the case of linear-time-invariant plants and propose a procedure to build a timed automaton that captures the sampling behavior of each event-triggered controller. We show that these timed automata approximately simulate the controllers' sampling behavior. Finally, a conflict-free scheduling policy is synthesized using timed game automata to guarantee reliable communication in the network.**

I. Introduction

Networked control systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators and controllers occurs through a shared band-limited digital communication network. Such structure makes the application of traditional control strategies with periodic sampling inefficient. Alternative approaches with aperiodic sampling such as event-triggered control (ETC) [1] and self-triggered control (STC) [2] have been proposed in order to reduce the usage of resources in the implementation of control loops, in particular communication bandwidth. In these approaches, the execution time of control tasks is based on the violation of a pre-specified condition. More precisely, in ETC an intelligent sensory system is responsible to decide the execution time of control tasks, whereas in STC the execution time is determined by the controller. However, there is a lack of a well-established framework to analyze their corresponding communication load and synthesize communication schedulers.

Several codesign approaches of real-time systems can be found in the literature, e.g. feedback modification to task attributes [3], [4], [5], [6], anytime controllers [7], [8], and event-based control and scheduling [9], [10]. The aforementioned studies address simultaneously the controller and scheduler design. We decouple the event-triggered implementation and scheduling design in this study, by characterizing the sampling behavior of linear-time-invariant (LTI) systems with event-triggered implementations. Then, such characterizations can be analyzed independently for scheduling purposes, thus providing a scalable and versatile event-triggered NCSs design procedure. We generate finite-state quotient systems that approximately simulate the original infinite-state systems. The procedure consists of three steps. First, the state space is partitioned into finitely many conic regions. Then, for each conic region, we determine a time interval by using the convex embedding approach proposed in [11]. This time interval captures all inter-sample times associated to sampled states in the corresponding conic region. Finally, possible transitions between these conic regions are computed by using a reachability analysis, e.g. [12]. Furthermore, we show that the obtained quotient system can be recast as a timed automaton (TA) [13], which then can be used to synthesize conflict-free communication schedules.

The scheduling problem over TA and its variants has been studied in the literature, e.g. steel plant [14], job shop [15], task graph [16]. Furthermore, the optimal scheduling of a production w.r.t. a predefined cost for a finite time horizon has been investigated in [17], [18]. In this case, the models are TAs with weights (or costs) on both locations and edges [19], [20]. Finally the optimal scheduling for infinite time horizon is discussed in [21]. In this work, we assume that at any time instant, the shared communication network can be used by at most one control loop to update its input value. We propose a technique to synthesize a conflict-free schedule using a network of timed game automata (NTGA) [22], in which in order to avoid communication conflicts, we allow control loops to be updated earlier than the event triggering dictates.

The authors are with the Delft Center for Systems and Control, TU Delft - Delft University of Technology, The Netherlands - {a.sharifikolarijani,d.adzkiya,m.mazo}@tudelft.nl

II. Models and preliminaries

A. Timed Automata and Timed Game Automata

A timed automaton (TA) [13] is a directed graph extended with real-valued variables (called clocks) that model the logical clocks. Such automaton can be used to model real-time systems. We define C as a set of finitely many clocks. Clock constraints are used to restrict the behavior of the automaton. A clock constraint is a conjunctive formula of atomic constraints of the form $x \bowtie n$ or $x - y \bowtie n$ for $x, y \in C$, $\bowtie \in \{\leq, <, =, >, \geq\}$ and $n \in \mathbb{N}$. We use $\mathcal{B}(C)$ to denote the set of clock constraints.

Definition 1: [Timed (game) automata] A *timed automaton* TA is a sextuple $(L, \ell_0, \mathsf{Act}, C, E, \mathsf{Inv})$ where:

- L is a set of finitely many locations (or vertices);
- $\ell_0 \in L$ is the initial location;
- Act is the set of actions;
- C is a set of finitely many real-valued clocks;
- $E \subseteq L \times \mathcal{B}(C) \times \mathsf{Act} \times 2^C \times L$ is the set of edges;
- $\mathsf{Inv} : L \to \mathcal{B}(C)$ assigns invariants to locations.

The location invariants are restricted to constraints of the form: $c \leq n$ or $c < n$ where c is a clock and n is a natural number.

A *timed game automaton* TGA is a septuple $(L, \ell_0, \mathsf{Act}_c, \mathsf{Act}_u, C, E, \mathsf{Inv})$ where $(L, \ell_0, \mathsf{Act}_c \cup$

$\mathsf{Act}_u, C, E, \mathsf{Inv})$ is a timed automaton; Act_c is a set of controllable actions; Act_u is a set of uncontrollable actions; and $\mathsf{Act}_c \cap \mathsf{Act}_u = \varnothing$.

The semantics of a TA and a TGA are defined as a transition system where a state consists of the current location and the current value of clocks. There are two types of transitions between states. The automaton may either delay for some time (a delayed transition), or take an enabled edge (a discrete transition). Each edge is labeled with a guard ($\in \mathcal{B}(C)$), an action ($\in \mathsf{Act}$ or $\in \mathsf{Act}_c \cup \mathsf{Act}_u$) and a reset ($\in 2^C$). The guard is described as a clock constraint. An edge can be taken when the value of clocks satisfies the guard associated with the edge. A subset of all the clocks may be reset to zero when a discrete transition is taken. Finally each edge is associated with an action. The set of actions is used for synchronous communication between a pair of TAs or TGAs. It is done by hand-shake synchronization using input and output actions: if an edge labeled with output action $a!$ in a TA (or a TGA) is taken, an edge labeled with input action $a?$ in the other TA (or other TGA) has to be taken simultaneously. An action that can be taken independently is called internal action and will be denoted by $*$. In a TGA, controllable actions represent those that can be triggered by the controller, whereas uncontrollable actions are triggered by the environment/opponent. The composition of several TGAs is called network of timed game automata (NTGA), where the set of actions is defined as the cross product of the set of actions associated with all TGA. An action in an NTGA is controllable if all components are controllable, i.e. the opponent has priority over the controller.

B. Event Triggered LTI Control Systems

Consider a linear-time-invariant (LTI) system given by:

$$\dot{\xi}(t) = A\xi(t) + B\upsilon(t),\ \xi(t) \in \mathbb{R}^n, \upsilon(t) \in \mathbb{R}^m \quad (1)$$

with a linear state-feedback law implemented in a sample-and-hold manner:

$$\upsilon(t) = \upsilon(t_k) = K\xi(t_k),\ \forall t \in [t_k, t_{k+1}),\ k \in \mathbb{N}. \quad (2)$$

We further assume that no delay is present between reading the state and updating the actuators. The interested reader can find in [1] more details, including accounting for delays.

Denote by $\xi_x(t) \in \mathbb{R}^n$ the solution of (1)-(2) for $t \in [t_k, t_{k+1}]$ with initial condition $\xi_x(t_k) = x$. Moreover, consider $e_x(t) := x - \xi_x(t)$ as the virtual error introduced by the sampling process. The event triggering approach proposed in [1], suggests to use the following sampling law:

$$t_{k+1} = \min\{t > t_k \mid |e_x(t)|^2 \geq \alpha|\xi_x(t)|^2\},\ \alpha \in [0, 1),$$

resulting in the sampling interval associated to state x:

$$\tau(x) := \min\{t \mid |e_x(t)|^2 \geq \alpha|\xi_x(t)|^2, \xi_x(0) = x\}. \quad (3)$$

It follows that for $\sigma \in [0, t_{k+1} - t_k]$:

$$\xi_x(t_k + \sigma) = \Lambda(\sigma)x \text{ and } e_x(t_k + \sigma) = [I - \Lambda(\sigma)]x \quad (4)$$

where $\Lambda(\sigma) = [I + \int_0^\sigma e^{Ar}dr(A + BK)]$. Substituting (4) in (3), the state-dependent sampling law can be reformulated as

$$\tau(x) = \min\{\sigma > 0 \mid x^T\Phi(\sigma)x = 0\},\ \text{where} \quad (5)$$

$$\Phi(\sigma) = [I - \Lambda^T(\sigma)][I - \Lambda(\sigma)] - \alpha\Lambda^T(\sigma)\Lambda(\sigma). \quad (6)$$

C. Systems and Relations

In what follows, we briefly present some concepts from system theory, and propose a modified notion of *quotient system* (see e.g. [23] for the traditional definition).

Definition 2 (System [23]): A system is a sextuple $(X, X_0, U, \longrightarrow, Y, H)$ consisting of:

- a set of states X;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- a set of outputs Y;
- an output map $H : X \to Y$.

A system is said to be metric if the set of outputs Y is equipped with a metric, and autonomous if the cardinality of U is not larger than one.

Definition 3 (Approximate Simulation Relation [23]): Consider two metric systems $S_a = (X_a, X_{a0}, U_a, \underset{a}{\longrightarrow}, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, \underset{b}{\longrightarrow}, Y_b, H_b)$ with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$, where $\mathbb{R}_0^+$ represents the set of nonnegative real numbers. A relation $R \subseteq X_a \times X_b$ is an ε-approximate simulation relation from S_a to S_b if the following three conditions are satisfied:

1) $\forall x_{a0} \in X_{a0}, \exists x_{b0} \in X_{b0}$ such that $(x_{a0}, x_{b0}) \in R$;
2) $\forall (x_a, x_b) \in R,$ we have $d(H_a(x_a), H_b(x_b)) \leq \varepsilon$;
3) $\forall (x_a, x_b) \in R, (x_a, u_a, x'_a) \in \underset{a}{\longrightarrow}$ in S_a $\exists (x_b, u_b, x'_b) \in \underset{b}{\longrightarrow}$ in S_b satisfying $(x'_a, x'_b) \in R$.

We say that S_b ε-approximately simulates S_a, denoted by $S_a \preceq_{\mathcal{S}}^{\varepsilon} S_b$, if there exists an ε-approximate simulation relation R from S_a to S_b.

Definition 4 (Power Quotient System): Let $S = (X, X_0, \varnothing, \longrightarrow, Y, H)$ be an autonomous system and R be an equivalence relation on X. The power quotient of S by R, denoted by $S_{/R}$, is the autonomous system $(X_{/R}, X_{/R,0}, \varnothing, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$ consisting of:

- $X_{/R} = X/R$;
- $X_{/R,0} = \{x_{/R} \in X_{/R} | x_{/R} \cap X_0 \neq \varnothing\}$;
- $(x_{/R}, u, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\exists (x, u, x') \in \longrightarrow$ with $x \in x_{/R}$ and $x' \in x'_{/R}$;
- $Y_{/R} \subset 2^Y$;
- $H_{/R}(x_{/R}) = \bigcup\limits_{x \in x_{/R}} H(x)$.

Lemma 1: Let S be an autonomous metric system, R be an equivalence relation on X, and let the autonomous metric system $S_{/R}$ be the power quotient system of S by R. For any

$$\varepsilon \geq \max_{\substack{x \in x_{/R} \\ x_{/R} \in X/R}} d(H(x), H_{/R}(x_{/R})),$$

with d the Hausdorff distance over the set 2^Y, $S_{/R}$ ε-approximately simulates S, i.e. $S \preceq_{\mathcal{S}}^{\varepsilon} S_{/R}$.

Proof: Consider the candidate simulation relation: $R' \subset X \times X_{/R}$, where $(x, x_{/R}) \in R'$ if and only if $x \in x_{/R}$. From Definition 4, the conditions from Definition 3 immediately follow when one considers the Hausdorff distance and 2^Y as a common output set for S and $S_{/R}$. ■

III. Abstractions of event-triggered LTI systems

A. Problem Statement

Consider the system $S = (X, X_0, \varnothing, \longrightarrow, Y, H)$:

- $X = \mathbb{R}^n$;
- $X_0 = \mathbb{R}^n$;
- $(x, x') \in \longrightarrow$ iff $\xi_x(\tau(x)) = x'$ given by (1)-(3);
- $Y \subset \mathbb{R}^+$;
- $H : \mathbb{R}^n \to \mathbb{R}^+$ where $H(x) = \tau(x)$.

Such a system produces as output sequences all possible sequences of inter-sample intervals of the concrete system (1)-(2) with triggering condition (3).

Problem 1: Provide a construction of power quotient systems $S_{/\mathcal{P}}$ of systems S as defined above.

Based on Definition 4, we propose to construct the system $S_{/\mathcal{P}} = (X_{/\mathcal{P}}, X_{/\mathcal{P},0}, \varnothing, \underset{/\mathcal{P}}{\longrightarrow}, Y_{/\mathcal{P}}, H_{/\mathcal{P}})$ where

- $X_{/\mathcal{P}} = \mathbb{R}^n_{/\mathcal{P}} := \{\mathcal{R}_1, \ldots, \mathcal{R}_q\}$;
- $X_{/\mathcal{P},0} = \mathbb{R}^n_{/\mathcal{P}}$;
- $(x_{/\mathcal{P}}, x'_{/\mathcal{P}}) \in \underset{/\mathcal{P}}{\longrightarrow}$ if $\exists x \in x_{/\mathcal{P}}$, $\exists x' \in x'_{/\mathcal{P}}$ such that $\xi_x(H(x)) = x'$ as determined by (1)-(3);
- $Y_{/\mathcal{P}} \subset 2^Y \subset \mathbb{IR}^+$, where $\mathbb{IR}^+$ represents the set of closed intervals $[a, b]$ such that $0 < a \leq b$;
- $H_{/\mathcal{P}}(x_{/\mathcal{P}}) = [\min_{x \in x_{/\mathcal{P}}} H(x), \max_{x \in x_{/\mathcal{P}}} H(x)] := [\underline{\tau}_{x_{/\mathcal{P}}}, \bar{\tau}_{x_{/\mathcal{P}}}]$.

B. Construction of the Abstraction

In this subsection, we introduce how to: select an appropriate equivalence relation $\mathcal{P}$, compute the respective intervals $[\underline{\tau}_{x_{/\mathcal{P}}}, \bar{\tau}_{x_{/\mathcal{P}}}]$, and determine transitions among abstract states.

1) Set of States: The state set construction approach is heavily inspired by an important observation from (5):

Remark 1 ([24]): All states, excluding the origin, which lie on a line that goes through the origin, have the same inter-sample time, i.e. $\tau(x) = \tau(\lambda x)$, $\forall \lambda \neq 0$.

Remark 1 suggests a convenient approach is to partition the state space into a finite number of convex polyhedral cones (pointed at the origin) $\mathcal{R}_s$ where $s \in \{1, \ldots, q\}$ and $\bigcup_{s=1}^q \mathcal{R}_s = \mathbb{R}^n$. This state space abstraction technique is proposed by [24], dividing each of the angular spherical coordinates of $x \in \mathbb{R}^n$: $\theta_1, \ldots, \theta_{n-2} \in [0, \pi]$, $\theta_{n-1} \in [-\pi, \pi]$ into $\bar{m}$ (not necessarily equidistant) intervals resulting in $q = \bar{m}^{(n-1)}$ conic regions. Additionally, Remark 1 also suggests that it suffices to only consider half of the state space since x and $-x$ behave in the same way in (5). Therefore, one can consider half of the state space (e.g, by taking $\theta_{n-1} \in [0, \pi]$), and later map the results to the other half of the state space. We consider thus the following equivalence relation $\mathcal{P}$ to construct our abstraction:

$$(x, x') \in \mathcal{P} \Leftrightarrow \exists\, s \in \{1, \ldots, q\} \text{ s.t. } x, x' \in \mathcal{R}_s.$$

2) Output Map: The construction of the output map $H_{/\mathcal{P}}$ and the output set $Y_{/\mathcal{P}}$ are described in the sequel. A time interval $[\underline{\tau}_s, \bar{\tau}_s]$ is calculated such that $\forall x \in \mathcal{R}_s, \tau(x) \in [\underline{\tau}_s, \bar{\tau}_s]$ based on the approach proposed by [11]. This approach relies on the construction of a convex polytope (or a sequence of convex polytopes) in the space of real matrices around the matrix $\Phi(\sigma)$. This enables the reduction of the evaluation of $x^T\Phi(\sigma)x$ for infinitely many values of σ to a finite number of evaluations. Let q be the number of equivalence classes, i.e. $s \in \{1, \ldots, q\}$, $Q_s \in \mathcal{M}_2(\mathbb{R})$ and $E_s \in \mathcal{M}_{n\times p}(\mathbb{R})$ with $p \leq 2n - 2$, where $\mathcal{M}_{i\times j}(\mathbb{R})$ denotes the space of $i \times j$ real matrices. We assume that the equivalence classes of $\mathcal{P}$ are defined as cones pointed at the origin given by $\mathcal{R}_s = \{x \in \mathbb{R}^2 |\ x^T Q_s x \geq 0\}$, $Q_s = Q_s^T$ whenever $n = 2$ or $\mathcal{R}_s = \{x \in \mathbb{R}^n |\ E_s x \geq 0\}$ otherwise. Consider also a scalar $\bar{\sigma} > 0$ denoting a time instant for which the triggering mechanism (5) is enabled in the whole state space, i.e. $x^T\Phi(\bar{\sigma})x \geq 0, \forall x \in \mathbb{R}^n$. Let $N_{conv} + 1$ be the number of vertices employed to define the polytope containing $\Phi(\sigma)$ in a given time interval, and $l \geq 1$ the number of time subdivisions considered in the time interval $[0, \bar{\sigma}]$ (for the purpose of reducing the conservatism involved in the polytopic embedding of $\Phi(\sigma)$). For details of the proofs of these results we refer the interested reader to [25], [11], [24].

Lemma 2: Let $s \in \{1, \ldots, q\}$, and consider a time bound $\underline{\tau}_s \in (0, \bar{\sigma}]$. If $x^T\underline{\Phi}_{(i,j),s}x \leq 0$ holds $\forall (i, j) \in \mathcal{K}_s = (\{0, \ldots, N_{conv}\} \times \{0, \ldots, \lfloor \frac{\underline{\tau}_s l}{\bar{\sigma}} \rfloor\})$, then:

$$x^T\Phi(\sigma)x \leq 0, \quad \forall \sigma \in [0, \underline{\tau}_s]$$

with Φ defined in (6) and

$$\underline{\Phi}_{(i,j),s} = \hat{\underline{\Phi}}_{(i,j),s} + \underline{\nu} I,$$

$$\hat{\underline{\Phi}}_{(i,j),s} = \begin{cases} \sum_{k=0}^{i} L_{k,j}(\frac{\bar{\sigma}}{l})^k & \text{if } j < \lfloor \frac{\underline{\tau}_s l}{\bar{\sigma}} \rfloor, \\ \sum_{k=0}^{i} L_{k,j}(\underline{\tau}_s - \frac{j\bar{\sigma}}{l})^k & \text{if } j = \lfloor \frac{\underline{\tau}_s l}{\bar{\sigma}} \rfloor, \end{cases}$$

$$\begin{cases} L_{0,j} &= I - \Pi_{1,j} - \Pi_{1,j}^T + (1-\alpha)\Pi_{1,j}^T\Pi_{1,j}, \\ L_{1,j} &= [(1-\alpha)\Pi_{1,j}^T - I]\Pi_{2,j} \\ &\quad + \Pi_{2,j}^T[(1-\alpha)\Pi_{1,j} - I], \\ L_{k\geq 2,j} &= [(1-\alpha)\Pi_{1,j}^T - I]\frac{A^{k-1}}{k!}\Pi_{2,j} \\ &\quad + \Pi_{2,j}^T\frac{(A^{k-1})^T}{k!}[(1-\alpha)\Pi_{1,j} - I] \\ &\quad + (1-\alpha)\Pi_{2,j}^T(\sum_{i=1}^{k-1}\frac{(A^{i-1})^T}{i!}\frac{A^{k-i-1}}{(k-i)!})\Pi_{2,j}, \end{cases} \tag{7}$$

$$\begin{cases} \Pi_{1,j} = I + M_j(A + BK), & M_j = \int_0^{j\frac{\bar{\sigma}}{l}} e^{As}ds, \\ \Pi_{2,j} = N_j(A + BK), & N_j = AM_j + I, \end{cases}$$

$$\underline{\nu} \geq \max_{\substack{\sigma' \in [0, \frac{\bar{\sigma}}{l}] \\ r \in \{0, \ldots, l-1\}}} \lambda_{\max}(\Phi(\sigma' + r\tfrac{\bar{\sigma}}{l}) - \tilde{\Phi}_{N_{conv},r}(\sigma')), \tag{8}$$

$$\tilde{\Phi}_{N_{conv},r}(\sigma) = \sum_{k=0}^{N_{conv}} L_{k,r}\sigma^k. \tag{9}$$

Proof: This is a particular case of Lemma 1 in [11] when considering time intervals beginning at zero. ∎

Theorem 1 (Regional Lower Bound Approximation): Consider a scalar $\underline{\tau}_s \in (0, \bar{\sigma}]$ and matrices $\underline{\Phi}_{\kappa,s}$, $\kappa = (i, j) \in \mathcal{K}_s$, defined as in Lemma 2. If there exist scalars $\underline{\varepsilon}_{\kappa,s} \geq 0$ (for $n = 2$) or symmetric matrices $\underline{U}_{\kappa,s}$ with nonnegative entries (for $n \geq 3$) such that for all $\kappa \in \mathcal{K}_s$ the following LMIs hold:

$$\begin{cases} \underline{\Phi}_{\kappa,s} + \underline{\varepsilon}_{\kappa,s}Q_s \preceq 0 & \text{if } n = 2 \\ \underline{\Phi}_{\kappa,s} + E_s^T\underline{U}_{\kappa,s}E_s \preceq 0 & \text{if } n \geq 3 \end{cases},$$

the inter-sample time (3) of the system (1)-(2) is regionally bounded from below by $\underline{\tau}_s$, $\forall x \in \mathcal{R}_s$.

Proof: This result is a direct consequence of the application of the S-procedure (lossless for the case of $n = 2$ and lossy otherwise) along with Lemma 2. ∎

Lemma 3: Let $s \in \{1, \ldots, q\}$, and consider a time bound $\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$. If $x^T \bar{\Phi}_{(i,j),s} x \geq 0$ holds $\forall (i,j) \in \mathcal{K}_s = (\{0, \ldots, N_{conv}\} \times \{\lfloor \frac{\bar{\tau}_s l}{\bar{\sigma}} \rfloor, \ldots, l-1\})$, then:

$$x^T \Phi(\sigma) x \geq 0, \quad \forall \sigma \in [\bar{\tau}_s, \bar{\sigma}]$$

with Φ defined in (6) and

$$\bar{\Phi}_{(i,j),s} = \bar{\bar{\Phi}}_{(i,j),s} + \bar{\nu} I,$$

$$\bar{\bar{\Phi}}_{(i,j),s} = \begin{cases} \sum_{k=0}^{i} L_{k,j} (\frac{(j+1)\bar{\sigma}}{l} - \bar{\tau}_s)^k & \text{if } j = \lfloor \frac{\bar{\tau}_s l}{\bar{\sigma}} \rfloor, \\ \sum_{k=0}^{i} L_{k,j} (\frac{\bar{\sigma}}{l})^k & \text{if } j > \lfloor \frac{\bar{\tau}_s l}{\bar{\sigma}} \rfloor, \end{cases}$$

$$\bar{\nu} \leq \max_{\substack{\sigma' \in [0, \frac{\bar{\sigma}}{l}] \\ r \in \{0, \ldots, l-1\}}} \lambda_{\min}(\Phi(\sigma' + r\frac{\bar{\sigma}}{l}) - \tilde{\Phi}_{N_{conv}, r}(\sigma')), \tag{10}$$

where $L_{k,j}$ and $\tilde{\Phi}_{N_{conv},r}$ are given by (7) and (9), resp.

Proof: The proof of this lemma is analogous to that of Lemma 2, see [25] for details. ■

Theorem 2 (Regional Upper Bound Approximation): Consider a scalar $\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$ and matrices $\bar{\Phi}_{\kappa,s}$, $\kappa = (i,j) \in \mathcal{K}_s$, defined as in Lemma 3. If there exist scalars $\bar{\varepsilon}_{\kappa,s} \geq 0$ (for $n = 2$) or symmetric matrices $\bar{U}_{\kappa,s}$ with nonnegative entries (for $n \geq 3$) such that for all $\kappa \in \mathcal{K}_s$ the following LMIs hold:

$$\begin{cases} \bar{\Phi}_{\kappa,s} - \bar{\varepsilon}_{\kappa,s} Q_s \succeq 0 & \text{if } n = 2 \\ \bar{\Phi}_{\kappa,s} - E_s^T \bar{U}_{\kappa,s} E_s \succeq 0 & \text{if } n \geq 3 \end{cases},$$

the inter-sample time (3) of the system (1)-(2) is regionally bounded from above by $\bar{\tau}_s$, $\forall x \in \mathcal{R}_s$.

Proof: This result is a direct consequence of the application of the S-procedure (lossless for the case of $n = 2$ and lossy otherwise) along with Lemma 3. ■

Remark 2: To the best of our knowledge, there is no formal approach to compute $\bar{\sigma}$. However, one can perform a line search to find $\bar{\sigma}$ as the first value at which $\Phi(\bar{\sigma}) \succeq 0$. Alternatively, one can select $\bar{\sigma}$ as the largest inter-sample time allowed by practical motivations.

In order to compute $\underline{\tau}_s$ and $\bar{\tau}_s$ employing the previous results the following procedure can be followed. To determine $\underline{\tau}_s$, first use (8) to compute $\underline{\nu}$. Next, employ Theorem 1 to derive $\underline{\tau}_s$, for each $s \in \{1, \ldots, q\}$ by means of a line search on $\underline{\tau}_s$ (where $\underline{\tau}_s \in (0, \bar{\sigma}]$) combined with LMI feasibility problems on $\underline{\varepsilon}_{\kappa,s}$ or $\underline{U}_{\kappa,s}$ in each step of the line search. Similarly, to compute $\bar{\tau}_s$, one can use (10) to compute $\bar{\nu}$ and Theorem 2. Again that entails a combination of a line search on $\bar{\tau}_s$ (where $\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$) and LMI feasibility problems on $\bar{\varepsilon}_{\kappa,s}$ or $\bar{U}_{\kappa,s}$.

3) Transition Relation: Deriving all the transitions in $S_{/\mathcal{P}}$ boils down to computing the reachable set of each set $\mathcal{R}_s$ on the time interval $[\underline{\tau}_s, \bar{\tau}_s]$. For this reachability analysis, it suffices to consider only subsets $X_{0,s} \subset \mathcal{R}_s$ being convex polytopes with one vertex placed on each of the extreme rays of $\mathcal{R}_s$ (excluding the origin). Such a choice is justified by the following facts: i) all the states that lie on a line going through the origin have the same inter-sample time (see Remark 1); ii) the states of the system placed on a line going through the origin are mapped to another line that passes through the origin (since the state evolution (4) is a linear map on x); and iii) the image of a convex polytope under a linear map remains a convex polytope. Thus, computing (an over-approximation of) the reachable set of $X_{0,s}$ automatically provides (an over-approximation of) the reachable set of $\mathcal{R}_s$. Denote by $\mathcal{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s})$ the reachable set of $X_{0,s}$ during the time interval $[\underline{\tau}_s, \bar{\tau}_s]$, i.e.:

$$\mathcal{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s}) := \{x' \in \mathbb{R}^n \mid \exists x \in X_{0,s}, \exists \tau \in [\underline{\tau}_s, \bar{\tau}_s], x' = \xi_x(\tau)\}$$

Computing an over-approximation of the reachable set of a polytope under linear time invariants can be effectively computed following e.g. [12].

In order to derive the transitions in $S_{/\mathcal{P}}$, one needs to compute the intersection between the over-approximation of $\mathcal{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s})$, denoted by $\hat{\mathcal{X}}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s})$, and all conic regions $\mathcal{R}_t$. This can be done by solving a simple feasibility problem for each pair of conic regions $(\mathcal{R}_s, \mathcal{R}_t)$:

$$\text{Feas} \quad \begin{array}{l} H_s \bar{x} \leq d_s^\star \\ E_t \bar{x} \geq 0 \end{array} \tag{11}$$

where $\{\bar{x} \in \mathbb{R}^n \mid H_s \bar{x} \leq d_s^\star\}$ represents the polytopic over approximation $\hat{\mathcal{X}}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s})$, and $\{\bar{x} \in \mathbb{R}^n \mid E_t \bar{x} \geq 0\}$ characterizes t-th conic region where $t \in \{1, \ldots, q\}$. The feasibility of (11) indicates that there is a transition from abstract state $\mathcal{R}_s$ to $\mathcal{R}_t$ in $S_{/\mathcal{P}}$.

Remark 3: Due to the conservatism involved in the derivation of $\underline{\tau}_s$, $\bar{\tau}_s$, and the analysis of Section III-B, $\mathcal{S}_{/\mathcal{P}}$ is in fact an over approximation of the minimal power quotient system of $\mathcal{S}$, i.e. there exist spurious transitions $(x_{/R}, u, x'_{/R}) \in \underset{/R}{\longrightarrow}$ such that $\nexists (x, u, x') \in \longrightarrow$ with $x \in x_{/R}$ and $x' \in x'_{/R}$. This has no effect for the satisfaction of $S \preceq_{\mathcal{S}}^{\varepsilon} S_{/R}$ as Definition 4 does not preclude the presence of such spurious transitions.

Remark 4: The precision of the constructed abstraction can always be improved by subdividing each considered region into new regions, at the cost of constructing larger abstractions.

C. Timed Automata Representation

Finally, based on the underlying relation between an abstract state $x_{/\mathcal{P}} \in X_{/\mathcal{P}}$ and its corresponding output $y_{/\mathcal{P}} \in Y_{/\mathcal{P}}$, we show that $S_{/\mathcal{P}}$ is semantically equivalent to a TA. The system $S_{/\mathcal{P}}$:

1) remains at $x_{/\mathcal{P}}$ during the time interval $[0, \underline{\tau}_{x_{/\mathcal{P}}})$,
2) possibly leaves $x_{/\mathcal{P}}$ during the time interval $[\underline{\tau}_{x_{/\mathcal{P}}}, \bar{\tau}_{x_{/\mathcal{P}}})$, and
3) is forced to leave $x_{/\mathcal{P}}$ at $\bar{\tau}_{x_{/\mathcal{P}}}$.

Therefore, $S_{/\mathcal{P}}$ is equivalent to a timed automaton $\mathsf{TA} = (L, \ell_0, \mathsf{Act}, C, E, \mathsf{Inv})$ where:

- $L = X_{/\mathcal{P}}$;
- $\ell_0 := \mathcal{R}_s$ such that $\xi(0) \in \mathcal{R}_s$;
- $\mathsf{Act} = \{*\}$ is an arbitrary symbol;
- $C = \{c\}$;
- E is given by all tuples $(\mathcal{R}_s, g, a, r, \mathcal{R}_t)$ such that $(\mathcal{R}_s, \mathcal{R}_t) \in \underset{/\mathcal{P}}{\longrightarrow}$, $g = \{c \mid c \in [\underline{\tau}_s, \bar{\tau}_s]\}$, $a = *$, and r is given by $c := 0$;
- $\mathsf{Inv}(\mathcal{R}_s) := \{c \mid c \in [0, \bar{\tau}_s]\}, \forall s \in \{1, \ldots, q\}$.

IV. Scheduling networked control systems

Consider now a set of networked control systems (NCSs) sharing a common communication channel. Each control loop consists of a sensor, a plant, an actuator, and a controller, interconnected through the shared communication network. Assume that the network can be used by at most one control loop at any time instant. If several control loops request updates at the same time when the network is available a conflict arises, and only one control loop will be chosen nondeterministically to access the network. While in time-triggered control systems these type of problems can be prevented by appropriate scheduling, when one or several control-loops are event-triggered a-priori scheduling is not possible, because of the unknown event-times. In this section we propose an approach based on NTGA to avoid such conflicts. We consider schedulers that after each transmission of measurements decide whether the next update time of each control loop should be based on the triggering mechanism or forced to be at an earlier pre-defined time.

Definition 5: A timed game automaton generated by a communication network with minimum channel occupancy time Δ is given by $\mathsf{TGA}_{net} = (L, \ell_0, \mathsf{Act}_c, \mathsf{Act}_u, C, E, \mathsf{Inv})$ where

- $L = \{Idle, InUse, Bad\}$;
- $\ell_0 = Idle$;
- $\mathsf{Act}_c = \{early?, done!\}$;
- $\mathsf{Act}_u = \{start?\}$;
- $C = \{c\}$;
- $E = \{(Idle, c \geq 0, start?, \{c\}, InUse),$
  $(Idle, c \geq 0, early?, \{c\}, InUse),$
  $(InUse, c = \Delta, done!, \varnothing, Idle),$
  $(InUse, c \geq 0, early?, \varnothing, Bad),$
  $(InUse, c \geq 0, start?, \varnothing, Bad),$
  $(Bad, c \geq 0, early?, \varnothing, Bad),$
  $(Bad, c \geq 0, start?, \varnothing, Bad)\}$;
- $\mathsf{Inv}(Idle) = \{c | \ c \geq 0\}, \mathsf{Inv}(InUse) = \{c | \ c \leq \Delta\}$.

The location $Idle$ represents the network being available, $InUse$ represents the network is being used by a control loop, and Bad represents a conflict has happened. The active location changes from $Idle$ to $InUse$ when the control tasks are instantaneously executed. Notice that the clock variable c is reset during this discrete transition. We consider that a minimum channel occupancy time Δ needs to elapse before the network is free again to service the control tasks. During this time, the active location is $InUse$. When there is an update request from a control loop, a conflict happens and the location changes to Bad. Notice that Bad is an absorbing location, i.e. once we enter this location we cannot leave it. When the value of c equals Δ, the active location changes to $Idle$ and the network is available again.

Given a control loop, we construct now a TGA allowing a supervisor (scheduler) to force earlier controller updates than those dictated by the event-triggering mechanism.

Definition 6: Given a TA $(L, \ell_0, \mathsf{Act}, C, E, \mathsf{Inv})$ generated from an event-triggered control loop and a set of earliest update time parameters $\{d_1, \ldots, d_q\}$, the TGA with an option for earlier update is given by $(\widetilde{L}, \widetilde{\ell}_0, \widetilde{\mathsf{Act}}_c, \widetilde{\mathsf{Act}}_u, \widetilde{C}, \widetilde{E}, \widetilde{\mathsf{Inv}})$ where

- $\widetilde{L} = L \cup \bigcup_{s=1}^{q} \{Use_s, Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]}\}$;
- $\widetilde{\ell}_0 = \mathcal{R}_s$ s.t. $\xi(0) \in \mathcal{R}_s$;
- $\widetilde{\mathsf{Act}}_c = \{early!\}$;
- $\widetilde{\mathsf{Act}}_u = \{done?, start!\}$;
- $\widetilde{C} = C$;
- $\widetilde{E} = \bigcup_{s=1}^{q} \{(\mathcal{R}_s, \underline{\tau}_s \leq c \leq \bar{\tau}_s, start!, \varnothing, Use_s),$
  $(\mathcal{R}_s, \underline{\tau}_s - d_s \leq c \leq \underline{\tau}_s, early!, \varnothing, Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]})\} \cup$
  $\bigcup_{s=1}^{q} \bigcup_{\{t | \mathcal{R}_s \to \mathcal{R}_t \in E\}} \{(Use_s, c \geq 0, done?, \{c\}, \mathcal{R}_t)\} \cup$
  $\bigcup_{s=1}^{q} \bigcup_{t \in \mathcal{E}_s} \{(Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]}, c \geq 0, done?, \{c\}, \mathcal{R}_t)\}$;
- $\widetilde{\mathsf{Inv}}(\mathcal{R}_s) = \{0 \leq c \leq \bar{\tau}_s\}, \widetilde{\mathsf{Inv}}(Use_s) = \{c \geq 0\} = \widetilde{\mathsf{Inv}}(Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]})$, for all $s \in \{1, \ldots, q\}$;

with $\mathcal{E}_s = \{t | \hat{\mathcal{X}}_{[\underline{\tau}_s - d_s, \underline{\tau}_s]} \cap \mathcal{R}_t \neq \varnothing\}$.

The new locations Use_s and $Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]}$ for $s \in \{1, \ldots, q\}$ represent the control loop is using the network and the sampled state is in $\mathcal{R}_s$. Location Use_s is active when the update time is determined by the triggering mechanism (3). In this case, the scheduler does not choose the update time. Thus, edges from $\mathcal{R}_s$ to Use_s are labeled with the uncontrollable action $start!$. Location $Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]}$ is active when the update time is forced to be earlier. In this case, the scheduler is able to choose the update time. Thus, the edges from $\mathcal{R}_s$ to $Use_s^{[\underline{\tau}_s - d_s, \underline{\tau}_s]}$ are labeled with the controllable action $early!$.

Composing TGA_{net} and all the TGAs associated with the control loops, an NTGA abstracting the overall system of controllers sharing a network is obtained. Conflicts in this abstraction are associated to states for which the location of TGA_{net} is Bad. Thus, one can synthesize schedulers by solving safety problems on this NTGA aimed at avoiding these conflict states. This type of problems can be solved by existing tools like UPPAAL-Tiga [26].

V. Case study

We showcase the results in an example. with two NCSs sharing the same communication network. The first control loop is given by [1, p. 1683]

$$\dot{\xi} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \upsilon, \qquad \upsilon = \begin{bmatrix} 1 & -4 \end{bmatrix} \xi. \tag{12}$$

The second control loop is given by [27, p. 1699]

$$\dot{\xi} = \begin{bmatrix} -0.5 & 0 \\ 0 & 3.5 \end{bmatrix} \xi + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \upsilon, \ \upsilon = \begin{bmatrix} 1.02 & -5.62 \end{bmatrix} \xi. \tag{13}$$

We apply the procedure in Section III-B to (13) with the following design parameters: $\alpha = 0.05$, $N_{conv} = 5$, $l = 100$, $\bar{m} = 10$, and $\bar{\sigma} = 1$ time unit. The resulting number of conic regions is $q = 2 \times \bar{m}^{(n-1)} = 2 \times 10^{(2-1)} = 20$. Figure 1 (left) depicts the conic regions s and the associated $\underline{\tau}_s$ and $\bar{\tau}_s$. The set of edges is schematically shown in Fig. 1 (right). The outcome of the procedure for (12) is not reported here because of space limitations.

Next we synthesize a conflict-free communication scheduling policy for the two NCSs described above by using UPPAAL-Tiga [26]. In both control loops, the set of initial conditions is $\mathcal{R}_1$. We define the minimum channel occupancy time $\Delta = 0.005$ time units. For simplicity, the parameter for earlier triggering time is chosen to be the same for all regions, i.e. $d_1 = \cdots = d_{20}$, and denoted by d. Two scenarios are considered: In the first scenario,

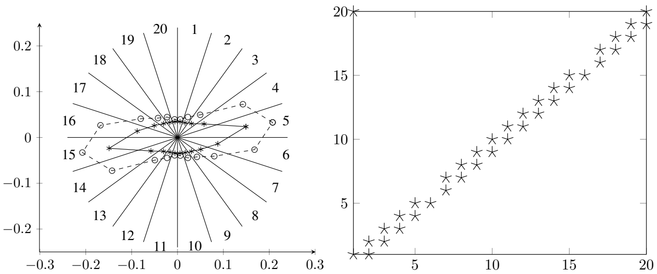

Fig. 1. (Left plot) Lower and upper bounds approximation of regional inter-sample times in (13) are depicted by black solid and black dashed lines, respectively. The radius of each asterisk and each circle denotes the value of lower and upper bounds in the corresponding region. The number in each conic region represents the index of that region. (Right plot) Schematic representation of set of edges in the timed automaton generated by (13). A star in coordinate (i, j) denotes the edge from location i to location j.

we try to synthesize a schedule for the following values of d: $0.001, 0.002, \dots, 0.010$, and we find that a conflict-free communication schedule can be synthesized if $d \geq 0.004$. In the second scenario, we try to determine the largest value of Δ for the following values of d: $0.004, 0.005, \dots, 0.010$ such that a conflict-free communication schedule can be synthesized. The procedure is as follows: for each value of d, initially we define $\Delta = 0.005$ and then increase Δ with step size of 0.001 until a schedule cannot be synthesized. We find that for all values of d, the largest value of Δ equals $d + 0.001$.

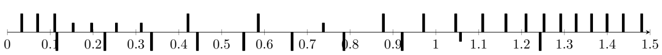

Fig. 2. Status of the shared communication network up to 1.5 time units. The bars on the top and on the bottom of the x axis represent the network is being used by (13) and (12), respectively. Long and short bars represent event-triggered and earlier mechanisms, respectively.

VI. Conclusions and Future Work

We have provided an algorithm to construct a TA that characterizes the sequence of sampling times in an event-triggered LTI system, and shown how one may use these TA's to synthesize conflict-free scheduling policies for sets of NCSs. The scheduling policy we propose is centralized, i.e. the scheduler needs to know the last transmitted state of all control loops. While this may be useful in systems in which a network coordinator is present, e.g. in wirelessHART, it is worth investigating extensions to fully decentralized scheduling. The strategy we present to resolve conflicts relies on earlier triggering of updates. Current research is focusing on strategies in which the scheduler may instead force control loops to employ event-triggering conditions that reduce the control performance to avoid communication conflicts. It would be interesting to extend this idea leveraging results from priced timed game automata, which extend timed game automata with costs on both locations and transitions, to optimize the performance that the scheduler delivers: i.e. minimize transmissions while maximizing convergence rates of the control loops. Finally, we are working on a public toolbox to construct the presented abstractions and schedulers, and on a real NCS implementation to evaluate the practical feasibility of the proposed techniques.

References

[1] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sept. 2007.

[2] M. Velasco, J. Fuertes, and P. Marti, “The self triggered task model for real-time control systems,” in *Proc. 24th IEEE Real-Time Systems Symposium (Work in Progress)*, 2003, pp. 67–70.

[3] G. Buttazzo, G. Lipari, and L. Abeni, “Elastic task model for adaptive rate control,” in *Proc. 19th IEEE Real-Time Systems Symposium (RTSS'98)*, Dec. 1998, pp. 286–295.

[4] M. Caccamo, G. Buttazzo, and L. Sha, “Elastic feedback control,” in *Proc. 12th Euromicro Conference on Real-Time Systems (ECRTS'00)*, 2000, pp. 121–128.

[5] C. Lu, J. Stankovic, S. Son, and G. Tao, “Feedback control real-time scheduling: Framework, modeling, and algorithms,” *Real-Time Systems*, vol. 23, no. 1-2, pp. 85–126, 2002.

[6] A. Cervin and J. Eker, “Control-scheduling codesign of real-time systems: The control server approach,” *Journal of Embedded Computing*, vol. 1, no. 2, pp. 209–224, 2004.

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

[8] D. Fontanelli, L. Greco, and A. Bicchi, “Anytime control algorithms for embedded real-time systems,” in *Hybrid Systems: Computation and Control (HSCC'08)*, 2008, pp. 158–171.

[9] S. Al-Areqi, D. Gorges, S. Reimann, and S. Liu, “Event-based control and scheduling codesign of networked embedded control systems,” in *Proc. 32nd Amer. Control Conf. (ACC'13)*, June 2013, pp. 5299–5304.

[10] S. Al-Areqi, D. Gorges, and S. Liu, “Stochastic event-based control and scheduling of large-scale networked control systems,” in *Proc. European Control Conf. (ECC'14)*, June 2014, pp. 2316–2321.

[11] L. Hetel, J. Daafouz, and C. Lung, “LMI control design for a class of exponential uncertain systems with application to network controlled switched systems,” in *Proc. American Control Conference (ACC'07)*, July 2007, pp. 1401–1406.

[12] A. Chutinan and B. Krogh, “Computing polyhedral approximations to flow pipes for dynamic systems,” in *Proc. 37th IEEE Conference on Decision and Control (CDC'98)*, vol. 2, Dec. 1998, pp. 2089–2094.

[13] R. Alur and D. Dill, “A theory of timed automata,” *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[14] A. Fehnker, “Scheduling a steel plant with timed automata,” in *Proc. 6th Int. Conf. Real-Time Computing Systems and Applications (RTCSA'99)*, 1999, pp. 280–286.

[15] Y. Abdeddaïm and O. Maler, “Job-shop scheduling using timed automata?” in *Computer Aided Verification (CAV'01)*, 2001, pp. 478–492.

[16] Y. Abdeddaïm, A. Kerbaa, and O. Maler, “Task graph scheduling using timed automata,” in *Proc. Int. Parallel and Distributed Processing Symposium (IPDPS'03)*, Apr. 2003, pp. 8 pp.–.

[17] G. Behrmann, E. Brinksma, M. Hendriks, and A. Mader, “Scheduling lacquer production by reachability analysis – a case study,” in *Proc. 16th IFAC World Congress*. Elsevier, 2005.

[18] ——, “Production scheduling by reachability analysis - a case study,” in *Proc. 19th IEEE Int. Parallel and Distributed Processing Symposium (IPDPS'05)*, Apr. 2005, pp. 140a–140a.

[19] G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager, “Minimum-cost reachability for priced time automata,” in *Hybrid Systems: Computation and Control (HSCC'01)*, 2001, pp. 147–161.

[20] R. Alur, S. La Torre, and G. Pappas, “Optimal paths in weighted timed automata,” in *Hybrid Systems: Computation and Control (HSCC'01)*, 2001, pp. 49–62.

[21] P. Bouyer, E. Brinksma, and K. Larsen, “Optimal infinite scheduling for multi-priced timed automata,” *Formal Methods in System Design*, vol. 32, no. 1, pp. 3–23, 2008.

[22] O. Maler, A. Pnueli, and J. Sifakis, “On the synthesis of discrete controllers for timed systems,” in *Proc. 12th Symp. on Theoretical Aspects of Computer Science (STACS'95)*, 1995, pp. 229–242.

[23] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer London, Limited, 2009.

[24] C. Fiter, L. Hetel, W. Perruquetti, and J.-P. Richard, “A state dependent sampling for linear state feedback,” *Automatica*, vol. 48, no. 8, pp. 1860–1867, 2012.

[25] A. Sharifi Kolarijani and M. Mazo Jr., “A formal traffic characterization of LTI event-triggered control systems,” *CoRR*, vol. abs/1503.05816, 2015.

[26] F. Cassez, A. David, E. Fleury, K. Larsen, and D. Lime, “Efficient on-the-fly algorithms for the analysis of timed games,” in *Concurrency Theory (CONCUR'05)*, 2005, pp. 66–80.

[27] L. Hetel, A. Kruszewski, W. Perruquetti, and J.-P. Richard, "Discrete and intersample analysis of systems with aperiodic sampling," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1696–1701, July 2011.