Event-triggered Control With Self-triggered Sampling for Discrete-time Uncertain Systems

Masako Kishida, *Member, IEEE*

***Abstract*—This paper proposes a resource-aware control approach for discrete-time linear uncertain systems. The approach uses a self-trigger condition to determine the sampling time to guarantee that the system is uniformly ultimately bounded and an event-trigger condition to determine control updates. The self-trigger condition is constructed using the skewed structured singular value to treat uncertainties in the prediction, and the event-trigger condition is constructed by considering the costs of sampling and control updates. A numerical example is provided to illustrate the approach.**

***Index Terms*—Event-triggered control, self-triggered control, uncertain systems.**

I. Introduction

Many recent control systems are connected via networks. Using a network for communication among controllers, sensors, and actuators reduces the deployment cost and increases flexibility compared with conventional control systems. This allows control systems to grow in size and complexity, and increases the ability to perform a wide variety of tasks. Thus, networked control systems are now widespread in our daily lives from power systems [1] to robotics [2] and health care systems [3].

Networked control systems are often required to perform control tasks with limited resources. This is may be because the communication band is shared with other tasks, or possibly because the devices use batteries. In such situations, using conventional periodic sampling [4] is not ideal because it is not designed to minimize resource usage, and thus performs nonessential communications and/or consumes energy unnecessarily. Motivated by this fact, aperiodic resource-aware control approaches, such as event- and self-triggered controls, have attracted many researchers (see [5]–[7], and references therein).

An event-triggered control samples the plant state continuously and updates the control signal when specified conditions are met [8], [9]. By contrast, a self-triggered control determines the next execution time of both the sampling and control update online using predictions based on the plant model and previously received sampled data. Recent studies have proposed several improvements to these event- and self-triggered controls. For example, the use of periodic sampling for event-triggered control [10], [11] and continuous sampling only after a specific minimum time since the last event [12] has been proposed to mitigate the continuous monitoring of the event-triggered control. The use of different asynchronized trigger conditions with different subsystems [13] and two event-trigger conditions for sampling and control input computation [9], [14] has also been considered. Self- and event-trigger conditions are both used in a model to schedule data transmissions and control action updates [15].

The main contribution of this paper is to present a resource-aware control approach that minimizes the “cost density” (see Definition 9) by using self-triggered sampling for event-triggered control in discrete-time linear systems subject to parametric uncertainties and disturbances. This idea of combining self- and event-triggered controls was initially introduced by the author and colleagues for continuous-time systems [16], [17]. This paper develops a discrete-time resource-aware control with the idea of [16] and [17]. For this purpose, as in [18], where self-triggered control for discrete-time linear uncertain systems is considered, we use the skewed structured singular value for predictions [19] to determine the sampling time and linear programming [20]–[22] to determine the control inputs. This allows us not only to optimize the overall control cost, but also to reduce the conservativeness because of uncertainties in the frequency of control updates. Note that unlike [15], this paper applies the event-trigger condition to the sampled data, instead of the output of the continuous-time plant model.

The remainder of this paper is organized as follows. Section II provides mathematical notations as well as definitions and lemmas that are used throughout this paper. Section III presents the mathematical model and problem setup. Section IV prepares mathematical tools for the proposed algorithms in the later sections. Section V presents a self-trigger algorithm to determine the sampling time instances. Section VI presents an event-trigger algorithm to determine the instances of the control input updates on top of the self-trigger sampling. Finally, a small numerical example given in Section VII is followed by the conclusion in Section VIII.

II. Preliminaries

A. Notation

For a vector $x \in \mathbb{R}^n$, $\|x\|_1$ is the l_1 norm and $\|x\|_\infty$ is the maximum norm of x. For two vectors $x, y \in \mathbb{R}^n$, the equality and inequalities are used to indicate the following:

1) $x = y$, iff $x_i = y_i$ for all $i = 1, \dots, n$;
2) $x < y$, iff $x_i < y_i$ for all $i = 1, \dots, n$; and
3) $x \leq y$, iff $x_i \leq y_i$ for all $i = 1, \dots, n$.

For a matrix $M \in \mathbb{R}^{n \times m}$, $\|M\|_2$ is the spectral norm and $\|M\|_\infty = \max_i \sum_{j=1}^{n} |M_{ij}|$ is the maximum absolute row sum norm of M, where M_{ij} indicates the ijth element of the matrix M. For a natural number n, I_n is the identity matrix of size n, e_i is the ith column of I_n, and $\mathbf{1}_n$ is the vector of ones whose length is n. The subscripts n are dropped when the size is clear. For matrices M and N, $M \otimes N$ is the Kronecker product.

B. Treatment of Uncertainty

Uncertainties in this paper are assumed to be unknown but bounded and structured. They are described using the following notation.

Manuscript received February 27, 2018; accepted June 4, 2018. Date of publication June 8, 2018; date of current version February 26, 2019. This work was supported by JSPS KAKENHI under Grant 16H07412. Recommended by Associate Editor Q.-S. Jia.

The author is with the Principles of Informatics Research Division, National Institute of Informatics 2-1-2 Hitotsubashi, Tokyo 101-8430, Japan (e-mail: kishida@nii.ac.jp).

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/TAC.2018.2845693

Let three nonnegative integers m_r, m_c, and m_C specify the number of independent uncertainties, and $k^i \in \mathbb{N}^{m_i}$, $i = r, c, C$. Then, a matrix block structure is defined by

$$\mathcal{K}(k^r; k^c; k^C) := (k_1^r, \ldots, k_{m_r}^r; k_1^c, \ldots, k_{m_c}^c; k_1^C, \ldots, k_{m_C}^C)$$

where $k^i = \varnothing$ if $m_i = 0$. The sets of structured uncertainties for a given $\mathcal{K}(k^r; k^c; k^C)$ are defined by

$$\begin{aligned}\boldsymbol{\Delta}_{\mathcal{K}} := \Big\{ &\mathrm{diag}[\delta_1^r I_{k_1^r}, \ldots, \delta_{m_r}^r I_{k_{m_r}^r}, \delta_1^c I_{k_1^c}, \ldots, \delta_{m_c}^c I_{k_{m_c}^c},\\ &\Delta_1, \ldots, \Delta_{m_C}] : \delta_i^r \in \mathbb{R},\ \delta_i^c \in \mathbb{C},\ \Delta_i \in \mathbb{C}^{k_i^C \times k_i^C} \Big\}\\ \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}} &:= \{\Delta \in \boldsymbol{\Delta}_{\mathcal{K}} : \|\Delta\|_2 \le 1\}\\ k\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}} &:= \{\Delta \in \boldsymbol{\Delta}_{\mathcal{K}} : \|\Delta\|_2 \le k\}.\end{aligned}$$

With the aforementioned notation, the skewed structured singular value is defined as follows.

Definition 1 (Skewed structured singular value ν [23]): For a matrix $M \in \mathbb{R}^{m \times m}$, and two block structures $\mathcal{K}_1$ and $\mathcal{K}_2$, the skewed structured singular value ν is defined by

$$\nu_{\mathcal{K}_1,\mathcal{K}_2}(M) := \frac{1}{\min \left\{ \begin{array}{c} k \ge 0 : \Delta_1 \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_1}, \Delta_2 \in k\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_2}, \\ \Delta = \mathrm{diag}[\Delta_1, \Delta_2], \text{ s.t.} \det(I - M\Delta) = 0 \end{array} \right\}}$$

unless no k exists for $\Delta_2 \in k\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_2}$ that makes $I - M\Delta$ singular, in which case $\nu_{\mathcal{K}_1,\mathcal{K}_2}(M) = 0$.

A scalar, vector, or matrix that is a rational function (i.e., a function that is a ratio of polynomials) of uncertain parameters can be expressed using the following linear fractional transform (LFT).

Definition 2 (LFT [24]): For matrices

$$M = \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right]$$

and Δ of compatible dimensions, an LFT is defined by

$$F_u(M, \Delta) := M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}.$$

If an LFT is scalar and $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$ is uncertain, its maximum absolute value can be expressed using the skewed structured singular value as follows.

Lemma 1 (Bound on the absolute value of a scalar LFT [19]): For a scalar LFT $F_u(M, \Delta) \in \mathbb{R}$ with $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$, it holds that

$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} |F_u(M, \Delta)| = \nu_{\mathcal{K},\mathcal{K}'}(M),\ \mathcal{K}' = (\varnothing; \varnothing; 1).$$

This $\mathcal{K}' = (\varnothing; \varnothing; 1)$ will be used throughout this paper.

Lemma 1 allows us to find both the maximum and minimum.

Corollary 1 (Bound on a scalar LFT [25]): For a scalar LFT $F_u(M, \Delta) \in \mathbb{R}$ with $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$, it holds that

$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} F_u(M, \Delta) = \nu_{\mathcal{K},\mathcal{K}'}(\bar{M}) - c$$

$$\min_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} F_u(M, \Delta) = -\nu_{\mathcal{K},\mathcal{K}'}(\underline{M}) + c$$

where $c > \nu_{\mathcal{K},\mathcal{K}'}(M)$, $\bar{M}$ is the same as M except that the M_{22} block is replaced by $M_{22} + c$, and $\underline{M}$ is the same as M except that the M_{22} block is replaced by $M_{22} - c$.

Here, note that M_{22} is scalar because $F_u(M, \Delta)$ is scalar.

The two-norm of an LFT can also be expressed using the skewed structured singular value as follows.

Lemma 2 (Bound on the two-norm of an LFT [19]): For an LFT $F_u(M, \Delta) \in \mathbb{R}^{n \times m}$ with $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$, it holds that

$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} \|F_u(M, \Delta)\|_2 = \nu_{\mathcal{K},\mathcal{K}''}(M),\ \mathcal{K}'' = (\varnothing; \varnothing; n).$$

III. Problem Setup

This section provides an overview of the problem setup by presenting the system model along with definitions used in the proposed algorithm.

We consider the uncertain linear discrete-time system of the form

$$x[t+1] = F_u(A, \Delta)x[t] + Bu[t] + Ew[t] \qquad (1)$$

where $x[t] \in \mathbb{R}^n$ is the system state, $u[t] \in \mathbb{R}^{n_u}$ is the control input, and $w[t] \in \mathcal{W} := \{w \in \mathbb{R}^{n_w} : \|w\|_\infty \le 1\}$ is the unknown but bounded disturbance for each time instance t, and $F_u(A, \Delta)$ is a constant but uncertain LFT for matrices A (known) and $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$ (unknown), $\mathcal{K} = \mathcal{K}(k^r; \varnothing; \varnothing)$, of compatible dimensions, $B \in \mathbb{R}^{n \times n_u}$ is a constant matrix, and $E \in \mathbb{R}^{n \times n_w}$ is a constant matrix.

Remark 1: Extending the proposed approach to a case in which the matrix E is also uncertain and in the form of an LFT is straightforward. In fact, if we let $E = F_u(\bar{E}, \Delta_E)$, then we can express $Ew[t] = F_u(\bar{W}, \Delta_w)$. Replace $\|E\|_\infty$ by $\max_{\Delta_E} \|F_u(\bar{E}, \Delta_E)\|_\infty$, and similarly for $\|Ew[t]\|_\infty$ and $\|e_l^T E\|_\infty$ throughout the paper.

Because the system (1) contains uncertainties, one natural control problem formulation is to consider a performance specification that requires the state to enter and remain in a set around the origin. This motivates us to use the following definitions.

Definition 3 (C-set [22]): A set $\mathcal{S}$ is said to be a C-set if it is a convex and compact set containing the origin in its interior.

Definition 4 (Uniform ultimate boundedness [20]): The system (1) is said to be uniformly ultimately bounded (UUB) in the C-set $\mathcal{S}$ if for every initial condition $x(0) = x_0 \in \mathbb{R}^n$, $T(x_0)$ exists such that for $t \ge T(x_0)$, $x(t) \in \mathcal{S}$.

Remark 2: $\mathcal{W}$ is a C-set.

We construct an approach to guaranteeing UUB for the system (1) in a box C-set

$$\mathcal{S}_p := \left\{ x : \left(I_n \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) x \le \mathbf{1}_{2n} \right\} \qquad (2)$$

while reducing the frequency of sampling and control updates, using the following structure [26]:

$$\begin{cases} t_{k+1} = t_k + T(x_k), \\ u[t] = u_k \in \mathcal{U}(x_k),\ t = t_k, t_k + 1, \ldots, t_{k+1} - 1 \end{cases} \qquad (3)$$

where t_k is the sampling time, x_k is the sampled state at time t_k, and u_k is the control input at time t_k. We assume that $t_0 = 0$. $T : \mathbb{R}^n \to \{1, \ldots, \bar{T}\}, \bar{T} \in \mathbb{N}$, and $T(x)$ denotes the time between two samplings. The integer $\bar{T}$ is a predefined upper bound on the intersampling time, which can be taken arbitrarily large as long as it is finite.

Definition 5 (Event-triggered control update): At each sampling time t_k, an algorithm of event-triggered control update performs the following:

1) finds a candidate for the next control input u_{new};
2) determines if the control input is held the same as the previous control input $u_k = u_{k-1}$ or is updated to the new control input $u_k = u_{\text{new}}$; and
3) updates the control input if $u_k = u_{\text{new}}$.

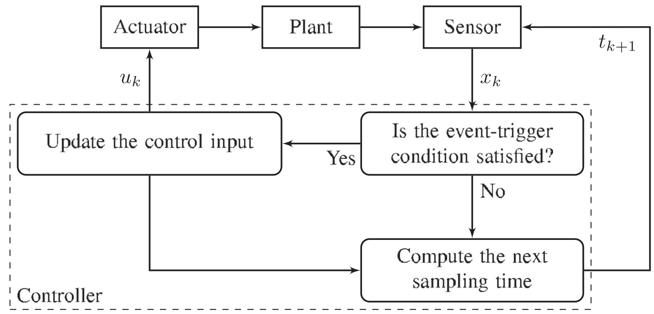


Fig. 1. Structure of the considered approach.

Definition 6 (Self-triggered sampling): At each sampling time t_k, an algorithm of self-triggered sampling determines the following.

1) The next sampling time t_{k+1} using the sampled data x_k and the control input u_k along with (1).

Therefore, the problem we consider is defined as follows.

Problem 1: Design an event-triggered control update algorithm with a self-triggered sampling algorithm that guarantees that the system (1) is UUB in the target C-set $\mathcal{S}_p$ defined in (2).

The overall approach is summarized in Fig. 1.

IV. Preparations for Algorithm Construction

This section presents a collection of building blocks for constructing solution approaches to the problem described in the previous section.

Let us first introduce the following definitions [20].

Definition 7 (Gauge function): A function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be a gauge function, if the following conditions hold

1) $\Psi(x) \geq 0$ and $\Psi(x) = 0 \Leftrightarrow x = 0$;
2) $\Psi(x+y) \leq \Psi(x) + \Psi(y)$ for all $x, y \in \mathbb{R}^n$; and
3) $\Psi(\lambda x) = \lambda \Psi(x)$ for $\lambda > 0$.

Definition 8 (Minkowski functional of $\mathcal{S}$): A gauge function $\Psi_{\mathcal{S}}(x)$ induced by a C-set $\mathcal{S}$ is said to be Minkowski functional of $\mathcal{S}$ and given by $\Psi_{\mathcal{S}}(x) = \inf\{\lambda > 0 : x \in \lambda \mathcal{S}\}$.

Thus, Definition 8 implies that $x \in \mathcal{S}$ if and only if $\Psi_{\mathcal{S}}(x) \leq 1$.

Note that for the target set $\mathcal{S}_p$ in (2), the corresponding Minkowski functional is [22]

$$\Psi_{\mathcal{S}_p}(x) = \|x\|_\infty. \tag{4}$$

A. Conditions for the Existence of Control Input to Guarantee UUB

We use the Minkowski functional for $\mathcal{S}_p$, $\Psi_{\mathcal{S}_p}(x)$, as a control Lyapunov function outside $\mathcal{S}_p$.

Theorem 1: Consider the system (1) with a target set $\mathcal{S}_p$ (2). The system is UUB if the following are satisfied.

1) For any $x[t] \in \mathcal{S}_p$
$$\Psi_{\mathcal{S}_p}(x[t+1]) \leq 1. \tag{5}$$
2) For any $x[t] \notin \mathcal{S}_p$, there exist $\lambda \in [0,1)$ and $\tau(x[t]) \in [1, \bar{T}]$, such that
$$\Psi_{\mathcal{S}_p}(x[t+\tau]) \leq \max\{\lambda^\tau \Psi_{\mathcal{S}_p}(x[t]), 1\}. \tag{6}$$
Here, the integer $\bar{T}$ is independent of $x[t]$ and can be taken arbitrarily large (this is the same $\bar{T}$ for (3)).

Proof: Directly from Definition 4. ■

To guarantee the existence of a control input that achieves UUB, we impose the following conditions.

Assumption 1: The system (1) satisfies all of the following conditions.

a) There exists $\varepsilon_1 \in [0,1)$ such that
$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, \|\delta\|_\infty \leq 1} \|A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}\delta\|_\infty \leq \varepsilon_1, \ \delta \in \mathbb{R}^n.$$
b) There exists $\varepsilon_2 \in [0, 1-\varepsilon_1)$ such that $\|E\|_\infty \leq \varepsilon_2$.
c) There exists $\varepsilon_0 \in (0, 1-\varepsilon_1-\varepsilon_2)$ such that the following linear program is feasible with respect to the variable $\left[\, \hat{k}^T \ \hat{\lambda}^T \,\right]^T$ for $\varepsilon_3 = 1 - \varepsilon_0 - \varepsilon_1 - \varepsilon_2$:
$$\left[\begin{array}{ccc|c} \hat{E}_1 b & \cdots & \hat{E}_L b & -H \\ -\hat{E}_1 b & \cdots & -\hat{E}_L b & -H \\ 0 & \cdots & 0 & -I_n \\ 0 & \cdots & 0 & \mathbf{1}_n^T \end{array}\right] \begin{bmatrix} \hat{k} \\ \hat{\lambda} \end{bmatrix} \leq \begin{bmatrix} -a \\ a \\ 0 \\ \varepsilon_3 \end{bmatrix} \tag{7}$$
where
$$a = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix}, \ b = \begin{bmatrix} b_1^T \\ \vdots \\ b_n^T \end{bmatrix}, \ H = \mathbf{1}_n \otimes I_n$$
and a_i is the ith row of matrix A_{22}, b_i is the ith row of matrix B, and $\hat{E}_l \in \mathbb{R}^{n \times n_u}$, $l = 1, \ldots, L = n \cdot n_u$, are $n \times n_u$ matrices that form a basis for $n \times n_u$ vector space. Thus, $\hat{k} \in \mathbb{R}^{n \cdot n_u}$ and $\hat{\lambda} \in \mathbb{R}^n$.

Assumption 1 provides restrictions on the allowable uncertainties in the state matrix A and the allowable effect of disturbances through the disturbance coefficient matrix E, relative to the elements of A_{22} and B that define the nominal system.

Remark 3: Assumption 1-a) is equivalent to
$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, \|\delta\|_\infty \leq 1} \left| F_u\left(\left[\begin{array}{cc|c} A_{11} & A_{12} & 0 \\ 0 & 0 & \mathbf{1} \\ \hline e_l^T A_{21} & 0 & 0 \end{array}\right], \mathrm{diag}[\Delta, \delta_1, \ldots, \delta_n] \right) \right| \leq \varepsilon_1$$
for all $l = 1, \ldots, n$. Satisfaction of this constraint can be checked using Lemma 1.

Remark 4: Assumption 1-a) can be replaced by the following.

a') There exists $\varepsilon_1 \in [0,1)$ such that
$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} \|A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}\|_2 \leq \frac{1}{\sqrt{n}}\varepsilon_1.$$

Satisfaction of this assumption can be checked using Lemma 2, which is easier to check. However, this could be more conservative than Assumption 1-a).

Theorem 2: For the system (1) with a target set $\mathcal{S}_p$ (2), Assumption 1 guarantees the existence of a control input $u[t] = u^*$ with some $\lambda^* \in [0,1)$ such that if $x[t] \in \mathcal{S}_p$, then
$$\Psi_{\mathcal{S}_p}(x[t+1]) \leq \lambda^* \tag{8}$$
for all $\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}$, $w[t] \in \mathcal{W}$. Furthermore,
$$u^* = Kx[t] = \begin{bmatrix} \hat{k}_1 & \cdots & \hat{k}_n \\ \vdots & \vdots & \vdots \\ \hat{k}_{L-n+1} & \cdots & \hat{k}_L \end{bmatrix} x[t]$$
$$\lambda^* = 1 - \varepsilon_0 < 1$$
where $\hat{k}$ (and $\hat{\lambda}$) are feasible solutions to (7).

Proof: The inequality (7) is equivalent to

$$\begin{cases} -H\hat{\lambda} \le a + \sum_{l=1}^{L} k_l \hat{E}_l b \le H\hat{\lambda} \\ 0 \le \hat{\lambda} \\ \mathbf{1}_n^T \hat{\lambda} \le \varepsilon_3 \end{cases}$$

$$\Leftrightarrow \begin{cases} -\hat{\lambda} \le a_i^T + K^T b_i^T \le \hat{\lambda},\ \forall i = 1, \dots, n \\ 0 \le \hat{\lambda} \\ \mathbf{1}_n^T \hat{\lambda} \le \varepsilon_3 \end{cases}.$$

This implies that

$$\left\|a_i^T + K^T b_i^T\right\|_1 \le \mathbf{1}_n^T \hat{\lambda} \le \varepsilon_3,\ \forall i = 1, \dots, n.$$

Thus

$$\|A_{22} + BK\|_\infty \le \mathbf{1}_n^T \hat{\lambda} \le \varepsilon_3.$$

Therefore

$$\begin{aligned} &\|F_u(A, \Delta)x[t] + Bu[t] + Ew[t]\|_\infty \\ &\le \|A_{22}x[t] + A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}x[t] + BKx[t]\|_\infty \\ &\quad + \|Ew[t]\|_\infty \\ &\le \|(A_{22} + BK)x[t]\|_\infty + \|A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}x[t]\|_\infty \\ &\quad + \|Ew[t]\|_\infty \\ &\le \|A_{22} + BK\|_\infty \|x[t]\|_\infty \\ &\quad + \|A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}\delta\|_\infty + \|E\|_\infty \|w[t]\|_\infty \\ &\le \|A_{22} + BK\|_\infty + \|A_{21}\Delta(I - A_{11}\Delta)^{-1}A_{12}\delta\|_\infty + \|E\|_\infty \\ &\le \varepsilon_3 + \varepsilon_1 + \varepsilon_2 = 1 - \varepsilon_0 = \lambda^* \in [0, 1). \end{aligned}$$ ■

Corollary 2: Under Assumption 1, the existence of a control input $u = u^*$ that satisfies Condition 1) in Theorem 1 is guaranteed.

Proof:

$$\|F_u(A, \Delta)x[t] + Bu^* + Ew[t]\|_\infty = \|x[t + 1]\|_\infty \le \lambda^* < 1$$

which implies (5). ■

Corollary 3: Under Assumption 1, the existence of a control input $u = u^*$ that satisfies Condition 2) with $\tau = 1$ in Theorem 1 is guaranteed.

Proof:

$$\Psi_{\mathcal{S}_p}(x[t + 1]) \le \lambda \quad \forall x[t] \in \mathcal{S}_p$$

for some $\lambda \in (0, 1]$ implies that [22]

$$\Psi_{\mathcal{S}_p}(x[t + 1]) \le \lambda \Psi_{\mathcal{S}_p}(x[t]) \quad \forall x[t] \notin \mathcal{S}_p.$$ ■

Thus, Corollaries 2 and 3 provide a sufficient condition to guarantee the existence of a control input that maintains the system (1) to be UUB.

B. Cost Optimization

Later in Section VI, we construct an event-trigger condition using "cost density," which is the cost associated with the sampling and control update per unit time, to minimize the overall cost for control.

Definition 9 (Cost density [16]): The cost density from time t_k to t_{k+1} is defined by

$$\rho_{k+1} := \frac{\text{cost used during } t_k \text{ to } t_{k+1}}{t_{k+1} - t_k}.$$

Here, the "cost used during t_k to t_{k+1}" is the sum of the cost of sampling and control update. The cost for sampling is the cost for one sampling multiplied by the number of samplings in $[t_k, t_{k+1})$, and the cost for control update is the cost for one control update multiplied by the number of control updates in $[t_k, t_{k+1})$.

In particular, if the ratio r of the cost for one control update and for one sampling is defined by

$$r := \frac{\text{cost for one control update}}{\text{cost for one sampling}}$$

then the normalized cost density (cost density normalized by sampling) is given by

$$\bar{\rho}_{k+1} = \begin{cases} \dfrac{r + 1}{t_{k+1} - t_k}, & \text{if there is exactly one control update} \\ & \text{in } [t_k, t_{k+1}) \\ \dfrac{1}{t_{k+1} - t_k}, & \text{if there is no control update} \\ & \text{in } [t_k, t_{k+1}). \end{cases} \tag{9}$$

Remark 5: Extending the presented results to the case in which the cost for control update and/or for sampling is time-varying is straightforward.

In practice, the cost for one sampling includes the sensing cost (power consumed by sensors to take one sampling) and the communication cost (cost to send the sampled data to the controller). The cost for one control update includes the communication cost (cost to send the actuation command to the actuator) and the actuator power cost (power consumed by actuators to actuate the plant, which may depend on the actuation signal). The use of cost density would considerably reduce the overall cost if the cost for sampling and for control update is very different. For example, if the control objective is to move heavy machinery in construction then, we would prefer not to move a machine very often because the cost for actuation would be much greater than the sensing cost.

V. Self-Trigger Condition for Sampling

Using the building blocks described in the previous Section IV, this section presents a theorem that summarizes a resource-aware sampling method and discusses the implementation of the theorem in control algorithms.

At every sampling time t_k, we determine the next sampling time t_{k+1} using Theorem 1 along with the sampled data x_k and control input u_k that is determined using the event-trigger condition (discussed in the next section).

Theorem 3: The following sampling strategy is sufficient to design control inputs that guarantee that the system (1) will be UUB with a target set $\mathcal{S}_p$ (2).

At every sampling time t_k, choose the next sampling time $t = t_{k+1} = t_k + T(x_k)$ such that the following conditions hold:

1) if $x_k \in \mathcal{S}_p$, then $T(x_k)$ is the largest $T \in [1, \bar{T}]$ that satisfies

$$\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k], \dots, w[t_k + i - 1] \in \mathcal{W}} \Psi_{\mathcal{S}_p}(x[t_k + i]) \le 1 \tag{10}$$

for all $i = 1, \dots, T$; and

2) if $x_k \notin \mathcal{S}_p$, then $T(x_k)$ is the largest $T \in [1, \bar{T}]$ that satisfies

$$\begin{aligned} &\max_{\Delta \in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k], \dots, w[t_k + i - 1] \in \mathcal{W}} \Psi_{\mathcal{S}_p}(x[t_k + i]) \\ &\le \max\{\lambda^i \Psi_{\mathcal{S}_p}(x_k), 1\} \end{aligned} \tag{11}$$

but not necessarily for all $i = 1, \dots, T - 1$.

Proof: Clearly, (10) guarantees (5), and (11) guarantees (6). ■

In both cases, the values of the right-hand sides of (10) and (11) are clear for each i for a given x_k.

However, finding the values of the left-hand sides of (10) and (11) is a nontrivial task.

Let $\bar{x}[t_k+1]$ and $\underline{x}[t_k+1]$ be vectors whose elements are defined as follows.

$$\begin{aligned}
\bar{x}_l[t_k+1] &:= \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k]\in\mathcal{W}} e_l^T\left(F_u(A,\Delta)x_k + Bu_k + Ew[t_k]\right)\\
&= \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_l^T F_u(A,\Delta)x_k + e_l^T Bu_k + \|e_l^T E\|_\infty\\
\underline{x}_l[t_k+1] &:= \min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k]\in\mathcal{W}} e_l^T\left(F_u(A,\Delta)x_k + Bu_k + Ew[t_k]\right)\\
&= \min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_l^T F_u(A,\Delta)x_k + e_l^T Bu_k - \|e_l^T E\|_\infty.
\end{aligned}$$

Then,

$$\begin{aligned}
&\max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k]\in\mathcal{W}} \Psi_{\mathcal{S}_p}(x[t_k+1])\\
&= \max_l\left(\max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k]\in\mathcal{W}} |e_l^T x[t_k+1]|\right)\\
&= \left\|\begin{bmatrix}\bar{x}[t_k+1]\\ \underline{x}[t_k+1]\end{bmatrix}\right\|_\infty.
\end{aligned}$$

For $i>1$, three approaches to computing bounds on $x[t_k+i]$ were discussed in [19]: a cheap but conservative approach, a less conservative but expensive approach, and a balanced approach somewhere between the two. In this paper, we adopt the cheapest approach [18].

For each time instance t_k+i, $i=1,2,\ldots$, let define the nominal vector x_{nom} and weight matrix W by

$$\begin{aligned}
x_{\text{nom}}[t_k+i] &:= \frac{1}{2}\left(\bar{x}[t_k+i] + \underline{x}[t_k+i]\right)\\
W[t_k+i] &:= \frac{1}{2}\text{diag}\left[\bar{x}[t_k+i] - \underline{x}[t_k+i]\right]
\end{aligned}$$

then the state x can be expressed in the form of LFT by

$$x[t_k+i] = F_u\left(A_{t_k+i}, \Delta_x\right)$$

where

$$\begin{aligned}
A_{t_k+i} &= \left[\begin{array}{c|c} 0 & \mathbf{1}\\ \hline W[t_k+i] & x_{\text{nom}}[t_k+i]\end{array}\right]\\
\Delta_x &\in \mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_x},\ \mathcal{K}_x = (\underbrace{1,\ldots,1}_{n};\varnothing;\varnothing).
\end{aligned}$$

Similar to $i=1$, for $i=2,3,\ldots$, let $\bar{x}[t_k+i]$ and $\underline{x}[t_k+i]$ be vectors whose elements are defined by

$$\begin{aligned}
\bar{x}_l[t_k+i] &:= \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k],\ldots,w[t_k+i-1]\in\mathcal{W}} e_l^T\left(F_u(A,\Delta)x[t_k+i-1]\right.\\
&\quad\left. + Bu_k + Ew[t_k+i-1]\right)\\
&= \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_{\text{aug}}}} F_u(\hat{A}_{t_k+i-1}, \Delta_{\text{aug}}) + e_l^T Bu_k + \|e_l^T E\|_\infty\\
\underline{x}_l[t_k+i] &:= \min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k],\ldots,w[t_k+i-1]\in\mathcal{W}} e_l^T\left(F_u(A,\Delta)x[t_k+i-1]\right.\\
&\quad\left. + Bu_k + Ew[t_k+i-1]\right)\\
&= \min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}_{\text{aug}}}} F_u(\hat{A}_{t_k+i-1}, \Delta_{\text{aug}}) + e_l^T Bu_k - \|e_l^T E\|_\infty
\end{aligned}$$

where

$$\begin{aligned}
\hat{A}_{t_k+i} &= \left[\begin{array}{cc|c} A_{11} & A_{12}W[t_k+i] & A_{12}x_{\text{nom}}[t_k+i]\\ 0 & 0 & \mathbf{1}\\ \hline e_l^T A_{21} & e_l^T A_{22}W[t_k+i] & e_l^T A_{22}x_{\text{nom}}[t_k+i]\end{array}\right]\\
\mathcal{K}_{\text{aug}} &= (k^r, \underbrace{1,\ldots,1}_{n};\varnothing;\varnothing)
\end{aligned}$$

for $l=1,\ldots,n$. Therefore

$$\max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t_k],\ldots,w[t_k+i-1]\in\mathcal{W}} \Psi_{\mathcal{S}_p}(x[t_k+i]) = \left\|\begin{bmatrix}\bar{x}[t_k+i]\\ \underline{x}[t_k+i]\end{bmatrix}\right\|_\infty. \tag{12}$$

Because (12) holds whether $x_k\in\mathcal{S}_p$ or $x_k\notin\mathcal{S}_p$, we can replace the left-hand sides of (10) and (11) by (12).

VI. Event-Trigger Condition for Control Update

In conjunction with self-triggered sampling described in Section V, this section presents approaches to determining control inputs and control updates using the results given in Section IV.

A. Choosing a Candidate for the Control Input

At every sampling time t_k, a candidate for the control input u_{new} is chosen by attempting to have the maximum contraction index $\lambda=\hat{\lambda}$ for the state be at the next time instance.

Thus, u_{new} is determined by solving linear programming as in [18], [20]–[22]

$$\begin{aligned}
&\min_u \hat{\lambda}\\
&\text{s.t.}\ \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}, w[t]\in\mathcal{W}} \mathcal{S}_p(x[t+1]) \le \hat{\lambda},\ \hat{\lambda}\ge 0.
\end{aligned}$$

Equivalently, the above linear programming can be expressed by

$$\begin{aligned}
&\min_{u,\hat{\lambda}} \begin{bmatrix}0\\1\end{bmatrix}^T\begin{bmatrix}u\\ \hat{\lambda}\end{bmatrix}\\
&\text{s.t.}\ \begin{bmatrix}B & -\mathbf{1}_n\\ -B & -\mathbf{1}_n\\ 0 & -1\end{bmatrix}\begin{bmatrix}u\\ \hat{\lambda}\end{bmatrix} \le \begin{bmatrix}-A_{22}x_k - b_{\max}\\ A_{22}x_k + b_{\min}\\ 0\end{bmatrix}
\end{aligned} \tag{13}$$

where

$$\begin{aligned}
b_{\max} &= \begin{bmatrix}\max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_1^T F_u(A_0,\Delta) - \|e_1^T E\|_\infty\\ \vdots\\ \max_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_n^T F_u(A_0,\Delta) - \|e_n^T E\|_\infty\end{bmatrix}\\
b_{\min} &= \begin{bmatrix}\min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_1^T F_u(A_0,\Delta) - \|e_1^T E\|_\infty\\ \vdots\\ \min_{\Delta\in\mathbf{B}\boldsymbol{\Delta}_{\mathcal{K}}} e_n^T F_u(A_0,\Delta) - \|e_n^T E\|_\infty\end{bmatrix}\\
A_0 &= \begin{bmatrix}A_{11} & A_{12}x_k\\ A_{21} & 0\end{bmatrix}.
\end{aligned}$$

Remark 6: This u_{new} is optimized for the time at t_k+1. Thus, it may happen that $u[t_{k-1}]$ yields a longer T in (10) and/or (11).

B. Determining if the Control is to be Updated or Not

To minimize the overall cost, we use the normalized cost density (9) to design an event-trigger condition for control updates. This is a greedy approach that minimizes each "block" of duration between samplings, and there is no guarantee to minimize the cost over long time horizons.

For this purpose, we compare the costs for two scenarios:

a) the control is updated at t_k; and

b) the control is not updated at t_k.

For the case of a), first, the new control input u_{new} is computed using (13). With u_{new} and the sampled data x_k, we determine the duration until the next sampling $T^{\text{update}}(x_k)$ according to (10) and (11) as described in the previous section. Then, the normalized cost density (9) is given by

$$\bar{\rho}_{k+1}^{\text{update}} = \frac{r+1}{T^{\text{update}}(x_k)}.$$

For the case of b), the control input is kept the same as u_{k-1}. As previously, with u_{k-1} and the sampled data x_k, we determine the duration until the next sampling $T^{\text{hold}}(x_k)$ according to (10) and (11). Then, the normalized cost density (9) is given by

$$\bar{\rho}_{k+1}^{\text{hold}} = \frac{1}{T^{\text{hold}}(x_k)}.$$

Thus, we let the event-trigger condition

$$\bar{\rho}_{k+1}^{\text{update}} < \bar{\rho}_{k+1}^{\text{hold}} \tag{14}$$

and update the control input if (14) is satisfied.

Remark 7: When implementing the proposed algorithm, we first check the event-trigger condition and determine the next control input u_k. We do not need to compute $T(x_k)$ again to find the next sampling time but can use $T^{\text{update}}(x_k)$ and $T^{\text{hold}}(x_k)$ that were already computed

$$t_{k+1} = \begin{cases} t_k + T^{\text{update}}(x_k) \\ \qquad \text{if the event-trigger condition is satisfied} \\ t_k + T^{\text{hold}}(x_k) \\ \qquad \text{if the event-trigger condition is not satisfied.} \end{cases}$$

VII. Numerical Example

To illustrate the proposed approach, numerical simulations were conducted for the system model (1).

Consider the uncertain state matrix

$$F_u(A, \Delta) = \begin{bmatrix} 0 & p_1 \\ p_2 & (p_1 - 1)p_2 \end{bmatrix}$$

which can be stable or unstable.

As a stable example,

1) $p_1 = 1.2 \pm 0.03$, $p_2 = 0.5 \pm 0.02$, and the actual system (unknown)

$$A_{\text{true}} = \begin{bmatrix} 0 & 1.1920 \\ 0.5270 & 0.1012 \end{bmatrix}$$

are used, and as an unstable example,

2) $p_1 = 1.3 \pm 0.03$, $p_2 = 0.6 \pm 0.02$, and the actual system (unknown)

$$A_{\text{true}} = \begin{bmatrix} 0 & 1.319 \\ 0.6294 & 0.2009 \end{bmatrix}$$

are used.

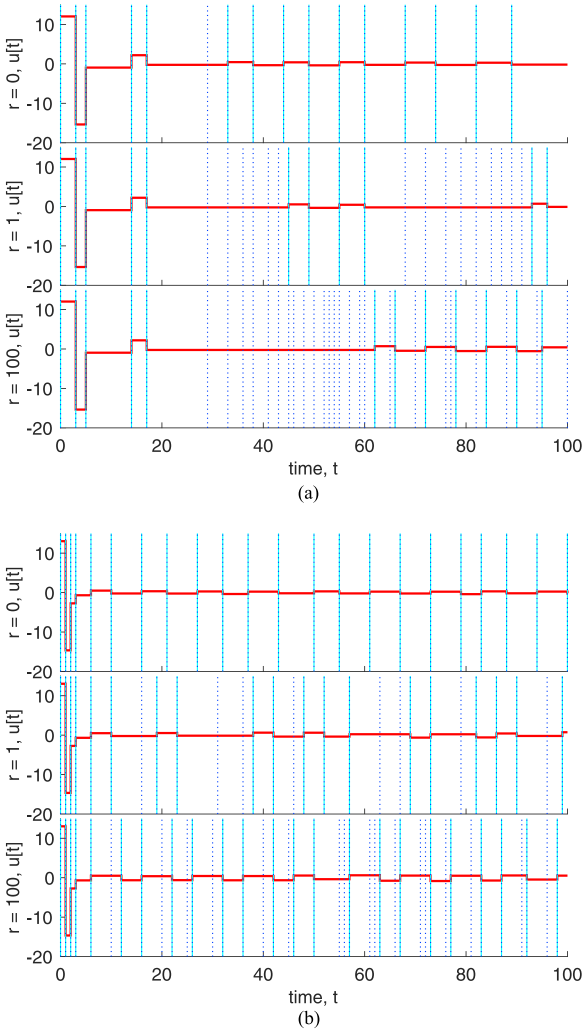
For other parameters, the matrices

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \quad E = \begin{bmatrix} 0.07 & 0 \\ 0 & -0.06 \end{bmatrix}$$

are used along with the initial condition

$$x[0] = x[t_0] = [12, \ -10]^T$$

and uniformly distributed disturbance $w[t]$ such that $\|w[t]\|_\infty \leq 1$.



(a)

(b)

Fig. 2. Control inputs for three different values of r (red: control input, blue vertical dots: time instances of sampling, cyan vertical solid line: time instances of control updates). (a) Stable system. (b) Unstable system.

Fig. 2 compares the control inputs along with sampling time and control update time for three values of r for both stable and unstable systems. For an extreme case of $r = 0$ (i.e., the control update does not incur any cost), the control input is updated nearly every time when sampled in both stable and unstable systems. This is because updating the control generally increases the duration until the next sampling. However, it can be observed that near time 30, the control input was not updated at the time of sampling in the stable system. This is explained in Remark 6. We can also observe that as the values of r increase, the control updates occur less frequently and the sampling occurs more frequently in the stable system. This is because as r increases, the cost of the control update increases and the controller tries to avoid control updates. As a result, the state can approach the boundary of the performance condition it must satisfy, thus requiring frequent sampling. However, as could have been imagined, an unstable system requires frequent control updates for any value of r. Nevertheless, the frequency of the update reduces compared to the frequency of sampling as r increases.

Fig. 3 presents typical state trajectories as a function of time for $r = 1$ for the stable system. As the state approaches the boundary (± 1), the control is updated.

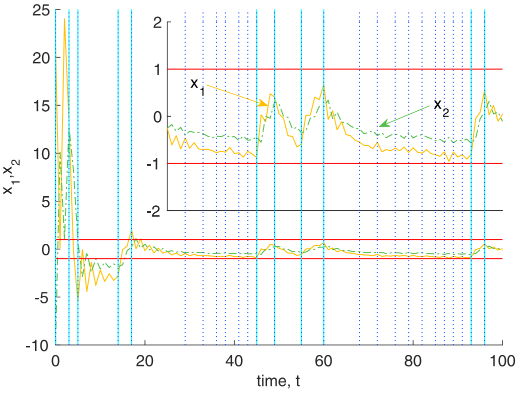


Fig. 3. State trajectories ($r = 1$) (orange: x_1, green dashed–dot: x_2, blue vertical dots: time instances of sampling, cyan vertical solid line: time instances of control updates, red horizontal lines at ± 1: the desired bounds).

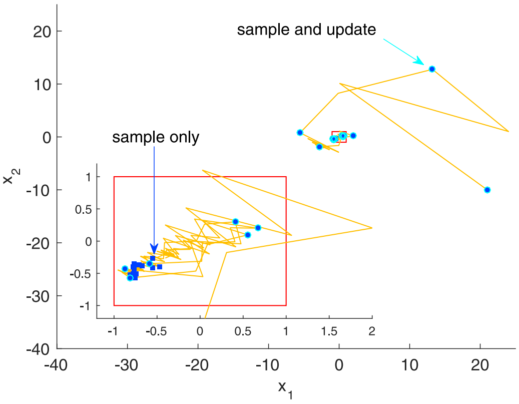


Fig. 4. Phase plane plot ($r = 1$) (red box: target set $\mathcal{S}_p$, blue squares: time instances of sampling, cyan circles: time instances of control updates).

Fig. 4 shows a typical state trajectory for $r = 1$ for the stable system. It can be observed that the state goes into the target set $\mathcal{S}_p$ and remains there. Note that the trajectory enters $\mathcal{S}_p$ and leaves $\mathcal{S}_p$ once before staying in $\mathcal{S}_p$. This is allowed because no sampling occurs in $\mathcal{S}_p$ before it leaves $\mathcal{S}_p$.

Little difference was observed for the unstable system corresponding to Figs. 3 and 4, and omitted.

To compute the values of ν, the SMAC Toolbox [27] was used.

VIII. Conclusion

With a focus on reducing the number of sampling and control updates, this paper proposed a resource-aware control approach to achieve UUB for an uncertain linear system expressed in the form of an LFT. This is accomplished by combining self-trigger sampling with event-triggered control updates. The results of the numerical example indicate satisfactory performance of the proposed approach.

References

[1] A. Teixeira, H. Sandberg, and K. H. Johansson, “Networked control systems under cyber attacks with applications to power networks,” in *Proc. Amer. Control Conf.*, 2010, pp. 3690–3696.

[2] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks* (ser. Applied Mathematics Series). Princeton, NJ, USA: Princeton Univ. Press, 2009.

[3] M. Franceschetti and P. Minero, “Elements of Information Theory for Networked Control Systems,” in *Information and Control in Networks*. New York, NY, USA: Springer-Verlag, 2014, pp. 3–37.

[4] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2011.

[5] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *Proc. IEEE Conf. Decision Control*, 2012, pp. 3270–3285.

[6] M. Mazo and P. Tabuada, “On event-triggered and self-triggered control over sensor/actuator networks,” in *Proc. IEEE Conf. Decision Control*, 2008, pp. 435–440.

[7] R. Postoyan, P. Tabuada, D. Nešić, and A. Anta, “Event-triggered and self-triggered stabilization of distributed networked control systems,” in *Proc. IEEE Conf. Decision Control Eur. Control Conf.*, 2011, pp. 2565–2570.

[8] M. Lemmon, “Event-Triggered Feedback in Control, Estimation, and Optimization,” in *Networked Control Systems*. New York, NY, USA: Springer-Verlag, 2010, pp. 293–358.

[9] P. Tallapragada and N. Chopra, “Event-triggered dynamic output feedback control for LTI systems,” in *Proc. IEEE Conf. Decision Control*, 2012, pp. 6597–6602.

[10] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, “Periodic event-triggered control for linear systems,” *IEEE Trans. Automat. Control*, vol. 58, no. 4, pp. 847–861, Apr. 2013.

[11] C. Peng and Q. L. Han, “A novel event-triggered transmission scheme and L_2 control co-design for sampled-data control systems,” *IEEE Trans. Automat. Control*, vol. 58, no. 10, pp. 2620–2626, Oct. 2013.

[12] F. Forni, S. Galeani, D. Nešić, and L. Zaccarian, “Event-triggered transmission for linear control over communication channels,” *Automatica*, vol. 50, no. 2, pp. 490–498, 2014.

[13] L. Lu, “Asynchronous separable self-triggered model predictive control based on relaxed dynamic programming,” *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 948–953, 2015.

[14] S. Hu and D. Yue, “$\mathcal{L}_2$-gain analysis of event-triggered networked control systems: A discontinuous Lyapunov functional approach,” *Int. J. Robust Nonlinear Control*, vol. 23, no. 11, pp. 1277–1300, 2013.

[15] H. Yu, E. Garcia, and P. J. Antsaklis, “Model-based scheduling for networked control systems,” in *Proc. Amer. Control Conf.*, 2013, pp. 2350–2355.

[16] M. Kishida, M. Kögel, and R. Findeisen, “Event-triggered actuator signal update using self-triggered sampled data for uncertain linear systems,” in *Proc. Amer. Control Conf.*, 2017, pp. 3035–3041.

[17] M. Kishida, M. Kgel, and R. Findeisen, “A combined event- and self-triggered control approach with guaranteed finite-gain l_2 stability for uncertain linear systems,” *IET Control Theory Appl.*, vol. 11, no. 11, pp. 1674–1683, Jul. 2017.

[18] M. Kishida, “Self-triggered control for uniform ultimate boundedness using skewed structured singular values,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 313–15 318, 2017.

[19] M. Kishida, P. Rumschinski, R. Findeisen, and R. D. Braatz, “Efficient polynomial-time outer bounds on state trajectories for uncertain polynomial systems using skewed structured singular values,” *IEEE Trans. Automat. Control*, vol. 59, no. 11, pp. 3063–3068, Nov. 2014.

[20] F. Blanchini, “Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions,” *IEEE Trans. Automat. Control*, vol. 39, no. 2, pp. 428–433, Feb. 1994.

[21] F. Blanchini, “Nonquadratic Lyapunov functions for robust control,” *Automatica*, vol. 31, no. 3, pp. 451–461, 1995.

[22] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Basel, Switzerland: Birkhäuser, 2015.

[23] R. S. R. Smith”, “Model validation for uncertain systems,” Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1990.

[24] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

[25] M. Kishida, P. Rumschinski, R. Findeisen, and R. D. Braatz, “Efficient polynomial-time outer bounds on state trajectories for uncertain polynomial systems using skewed structured singular values,” in *Proc. IEEE Int. Symp. Comput.-Aided Control Syst. Design*, 2011, pp. 216–221.

[26] T. Gommans, D. Antunes, T. Donkers, P. Tabuada, and M. Heemels, “Self-triggered linear quadratic control,” *Automatica*, vol. 50, no. 4, pp. 1279–1287, 2014.

[27] C. Roos, “Systems modeling, analysis and control (SMAC) toolbox: An insight into the robustness analysis library,” in *Proc. IEEE Multiconf. Syst. Control*, Hyderabad, India, 2013, pp. 176–181.