

Aperiodic Sampled-Data Control via Explicit Transmission Mapping: A Set Invariance Approach

Kazumune Hashimoto, *Student Member, IEEE*, Shuichi Adachi, *Member, IEEE*,
and Dimos V. Dimarogonas, *Member, IEEE*

Abstract—Event-triggered and self-triggered control have been proposed in recent years as promising control strategies to reduce communication resources in Networked Control Systems (NCSs). Based on the notion of set-invariance theory, this note presents new self-triggered control strategies for linear discrete-time systems subject to input and state constraints. The proposed schemes not only achieve communication reduction for NCSs, but also ensure both asymptotic stability of the origin and constraint satisfactions. A numerical simulation example validates the effectiveness of the proposed approaches.

Index Terms—Event-triggered and self-triggered control, Constrained control, Set-invariance theory.

I. INTRODUCTION

Efficient network utilization and energy-aware communication protocols between sensors, actuators and controllers have been recent challenges in the community of Networked Control Systems (NCSs). To tackle such challenges, event and self-triggered control schemes have been proposed as alternative approaches to the typical time-triggered controllers, see e.g., [1]–[3]. In contrast to the time-triggered case where the control signals are executed periodically, event and self-triggered strategies trigger the executions based on the violation of prescribed control performances, such as Input-to-State Stability (ISS) [1] and \mathcal{L}_∞ gain stability [2].

In particular, we are interested in designing self-triggered strategies for *constrained* control systems, where certain constraints such as physical limitations and actuator saturations need to be explicitly taken into account. One of the most popular control schemes to deal with such constraints is Model Predictive Control (MPC) [4]. In the MPC strategy, the current control action is determined by solving a constrained optimal control problem online, based on the knowledge of current state information and dynamics of the plant. Moreover, applications of the event and self-triggered control to MPC have been recently proposed to reduce the frequency of solving optimal control problems, see e.g., [5]–[10].

The main contribution of this note is to provide novel self-triggered strategies for constrained systems from an alternative perspective to the afore-cited papers, namely, a perspective from *set-invariance theory* [11]. Set invariance theory has

been extensively studied for the past two decades [12]–[14], and it provides a fundamental tool to design controllers for constrained control systems. Two established concepts are those of a *controlled invariant set* and λ -*contractive set*. While a controlled invariant set implies that the state stays inside the set for all time, a λ -contractive set guarantees the more restrictive condition that the state is asymptotically stabilized to the origin.

In this note, two different types of set-invariance based self-triggered strategies are presented. In the first approach, we formulate an optimal control problem such that the controller obtains stabilizing control inputs under *multiple candidates* of transmission time intervals. Among the multiple solutions, the controller selects a suitable one such that both control performance and communication load are taken into account. Asymptotic stability of the origin is ensured by using Lyapunov techniques, where the Lyapunov function is induced by a λ -contractive set obtained offline. Although the first approach guarantees asymptotic stability, it may lead to a high computation load as it requires to solve multiple optimization problems online. Therefore, we secondly propose an alternative strategy that aims to overcome the computational drawback of the first proposal. Similarly to the concept of explicit MPC [15], we provide an *offline*, explicit mapping that sends the state information to the desired transmission time interval. As we will see in later sections, the state-space is decomposed into a finite number of subsets, to which appropriate transmission time intervals are assigned.

The rest of the paper is organized as follows. In Section II, the system description and some preliminaries of invariant set theory are given. In Section III, we propose the first approach of the self-triggered strategy. In Section IV, the second approach of the self-triggered strategy is presented. In Section V, a illustrative simulation example is given. We finally conclude in Section VI.

(Nomenclature): Let \mathbb{R} , \mathbb{R}_+ , \mathbb{N} , \mathbb{N}_+ be the *non-negative reals*, *positive reals*, *non-negative* and *positive integers*, respectively. The *interior* of the set $\mathcal{S} \subset \mathbb{R}^n$ is denoted as $\text{int}\{\mathcal{S}\}$. A set $\mathcal{S} \subset \mathbb{R}^n$ is called *\mathcal{C} -set* if it is compact, convex, and $0 \in \text{int}\{\mathcal{S}\}$. For vectors v_1, \dots, v_N , $\text{co}\{v_1, \dots, v_N\}$ denotes their *convex hull*. A set of vectors $\{v_1, \dots, v_N\}$ whose convex hull gives a set \mathcal{P} (i.e., $\mathcal{P} = \text{co}\{v_1, \dots, v_N\}$), and each v_n , $n \in \{1, 2, \dots, N\}$ is not contained in the convex hull of $v_1, \dots, v_{n-1}, v_{n+1}, \dots, v_N$ is called a set of *vertices* of \mathcal{P} . Given a \mathcal{C} -set $\mathcal{S} \subset \mathbb{R}^n$, denote by $\partial\mathcal{S} \subset \mathbb{R}^n$ the boundary of \mathcal{S} . For a given $\lambda \in \mathbb{R}$ and a \mathcal{C} -set $\mathcal{S} \subset \mathbb{R}^n$, denote $\lambda\mathcal{S}$ as

Kazumune Hashimoto and Shuichi Adachi are with Department of Applied Physics and Physico-Informatics, Keio University, Japan. Hashimoto's work is supported by Grant-in-Aid for JSPS Research Fellow (Grant No. 17J05743).

Dimos V. Dimarogonas is with the department of Electrical Engineering, KTH Royal Institute of Technology, Sweden. His work was supported by the Swedish Research Council (VR), the Swedish Foundation for Strategic Research (SSF), ERC Starting Grant BUCOPHYSYS, and Knut och Alice Wallenberg foundation (KAW).

$\lambda\mathcal{S} = \{\lambda x \in \mathbb{R}^n : x \in \mathcal{S}\}$. Given a set $\mathcal{S} \subset \mathbb{R}^n$, the function $\Psi_{\mathcal{S}} : \mathbb{R}^n \rightarrow \mathbb{R}_+$ with $\Psi_{\mathcal{S}}(x) = \inf\{\mu : x \in \mu\mathcal{S}, \mu \geq 0\}$ is called a *gauge function*. For given two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^n$, define $\mathcal{S}_1 \setminus \mathcal{S}_2$ as $\mathcal{S}_1 \setminus \mathcal{S}_2 = \{x \in \mathbb{R}^n : x \in \mathcal{S}_1, x \notin \mathcal{S}_2\}$.

II. PROBLEM FORMULATION AND SOME PRELIMINARIES

In this section, the system description and some established results of set-invariance theory are provided.

A. System description and control strategy

Consider a networked control system illustrated in Fig. 1. We assume that the dynamics of the plant are given by

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$

for $k \in \mathbb{N}$, where $x(k) \in \mathbb{R}^n$ is the state and $u(k) \in \mathbb{R}^m$ is the control variable. The state and control input are assumed to be constrained as $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$, $\forall k \in \mathbb{N}$, where $\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{U} \subset \mathbb{R}^m$ are both polyhedral \mathcal{C} -sets described as

$$\begin{aligned} \mathcal{X} &= \{x \in \mathbb{R}^n : H_x x \leq h_x\}, \\ \mathcal{U} &= \{u \in \mathbb{R}^m : H_u u \leq h_u\}, \end{aligned} \quad (2)$$

where $H_x \in \mathbb{R}^{n_x \times n}$, $H_u \in \mathbb{R}^{n_u \times m}$ and h_x, h_u are appropriately sized vectors having positive components. The control objective is to steer the state to the origin, i.e., $x(k) \rightarrow 0$ as $k \rightarrow \infty$. Let k_m , $m \in \mathbb{N}$ with $k_0 = 0$ be the transmission time instants when the plant transmits the state information $x(k_m)$ to the controller and updates the control input. In the self-triggered strategy, the transmission times are determined as

$$k_{m+1} = k_m + \Gamma(x(k_m)), \quad m \in \mathbb{N}, \quad (3)$$

where $\Gamma : \mathcal{X} \rightarrow \{1, 2, \dots, j_{\max}\}$ denotes a mapping that sends the state information to the corresponding transmission time interval. Here, a maximal transmission time interval $j_{\max} \in \mathbb{N}_+$ is set a priori in order to formulate the self-triggered strategy. Due to the limited nature of communication bandwidth, we assume that only one control sample (not a sequence of control samples) is allowed to be transmitted at each transmission time. Namely, the control input is constant between two consecutive inter-transmission times, i.e.,

$$u(k) = \kappa(x(k_m)) \in \mathcal{U}, \quad k \in [k_m, k_{m+1}), \quad (4)$$

where $\kappa : \mathcal{X} \rightarrow \mathcal{U}$ denotes the state-feedback control law. The following assumptions are made throughout the paper (see e.g., [12]):

Assumption 1. The pair (A, B) is controllable.

Assumption 2. The matrix B has full column rank.

B. Set-invariance theory

In the following, we define the standard notions of *controlled invariant set* and *λ -contractive set* [12], which are important concepts to characterize invariance and convergence properties for constrained control systems.

Definition 1 (Controlled invariant set). For a given \mathcal{C} -set $\mathcal{S} \subseteq \mathcal{X}$, \mathcal{S} is said to be a *controlled invariant set* in \mathcal{X} , if and only if

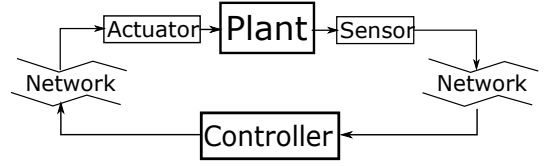


Fig. 1. Networked Control System

there exists a control law $g(x) \in \mathcal{U}$ such that $Ax + Bg(x) \in \mathcal{S}$ for all $x \in \mathcal{S}$.

Definition 2 (λ -contractive set). For a given \mathcal{C} -set $\mathcal{S} \subseteq \mathcal{X}$, \mathcal{S} is said to be a *λ -contractive set* in \mathcal{X} for $\lambda \in [0, 1]$, if and only if there exists a control law $g(x) \in \mathcal{U}$ such that $Ax + Bg(x) \in \lambda\mathcal{S}$ for all $x \in \mathcal{S}$.

Roughly speaking, a set \mathcal{S} is called *λ -contractive set* if all states in \mathcal{S} can be driven into a tighter (or equivalent) region $\lambda\mathcal{S}$ by applying a one-step control input. From the definition, a controlled invariant set implies a *λ -contractive set* with $\lambda = 1$. We review several established results for obtaining a contractive set and the corresponding properties. For given $\lambda \in [0, 1)$ and \mathcal{C} -set $\mathcal{X} \subset \mathbb{R}^n$, there are several ways to efficiently construct a *λ -contractive set* in \mathcal{X} . For a given \mathcal{C} -set $\mathcal{D} \subset \mathbb{R}^n$, let $\mathcal{Q}_\lambda : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be the mapping

$$\mathcal{Q}_\lambda(\mathcal{D}) = \{x \in \mathcal{X} : \exists u \in \mathcal{U}, Ax + Bu \in \lambda\mathcal{D}\}. \quad (5)$$

A simple algorithm to obtain a *λ -contractive set* in \mathcal{X} is to compute $\Omega_j \subset \mathbb{R}^n$, $j \in \mathbb{N}$ as

$$\Omega_0 = \mathcal{X}, \quad \Omega_{j+1} = \mathcal{Q}_\lambda(\Omega_j) \cap \mathcal{X}, \quad (6)$$

and then it holds that the set $\mathcal{S} = \lim_{j \rightarrow \infty} \Omega_j$ is *λ -contractive*, see e.g., [12]. If $\Omega_{j+1} = \Omega_j$ for some j , the *λ -contractive set* is obtained as $\mathcal{S} = \Omega_j$, which requires only a finite number of iterations. Although such condition does not hold in general, it is still shown, under Assumptions 1 and 2, that the algorithm converges in the sense that for every $\lambda < \bar{\lambda} < 1$, there exists a finite $j \in \mathbb{N}_+$ such that the set Ω_j is *$\bar{\lambda}$ -contractive* (see *Theorem 3.2* in [12]). Several other algorithms have been recently proposed, see e.g., [16], [17] and see also [18] for a detailed convergence analysis. The following lemma illustrates the existence of a (non-quadratic) Lyapunov function in a given *λ -contractive set*:

Lemma 1. [12]: Let $\mathcal{S} \subset \mathcal{X}$ be a *λ -contractive \mathcal{C} -set* with $\lambda \in [0, 1]$ and the associated gauge function $\Psi_{\mathcal{S}} : \mathcal{S} \rightarrow \mathbb{R}_+$. Then, there exists a control law $g : \mathcal{X} \rightarrow \mathcal{U}$ such that

$$\Psi_{\mathcal{S}}(Ax + Bg(x)) \leq \lambda\Psi_{\mathcal{S}}(x), \quad (7)$$

for all $x \in \mathcal{S}$.

Lemma 1 follows immediately from Definition 2. If $\lambda < 1$, (7) implies the existence of a stabilizing controller in \mathcal{S} in the sense that the output of the gauge function $\Psi_{\mathcal{S}}(\cdot)$ is guaranteed to decrease. The gauge function $\Psi_{\mathcal{S}}(\cdot)$ defined in \mathcal{S} is known as *set-induced Lyapunov function* in the literature; for a detailed discussion, see e.g., [12].

III. SELF-TRIGGERED STRATEGY

As described in the introduction, we propose two different types of self-triggered controllers; in this section, the first approach is presented.

A. Designing a stabilizing controller

For a given $\lambda \in [0, 1)$, let us first construct a λ -contractive set \mathcal{S} in \mathcal{X} . Note that since \mathcal{X} is a polyhedral \mathcal{C} -set, one can efficiently compute the λ -contractive set through polyhedral operations according to (6)¹. The obtained λ -contractive set \mathcal{S} can be denoted as

$$\mathcal{S} = \text{co}\{v_1, v_2, \dots, v_N\} \subseteq \mathcal{X}, \quad (8)$$

where $v_n, n \in \{1, 2, \dots, N\}$ represent the vertices of \mathcal{S} , and N represents the number of them.

Assumption 3. *The initial state is inside \mathcal{S} , i.e., $x(k_0) \in \mathcal{S}$.*

Based on Assumption 3, we will design the self-triggered strategy such that the state remains in \mathcal{S} for each transmission time instant. Suppose that at a certain transmission time k_m , $m \in \mathbb{N}$, the plant transmits the state information $x(k_m)$ to the controller. Based on $x(k_m)$, the controller needs to compute both a suitable controller to be applied and a transmission time interval, such that the state is stabilized to the origin. To this end, we first propose an approach to obtain stabilizing controllers under *multiple candidates* of transmission time intervals. More specifically, we obtain different control actions under different transmission time intervals, and the controller selects a suitable one among them. To obtain the stabilizing controllers, we formulate the following optimal control problem for each $j \in \{1, \dots, j_{\max}\}$:

Problem 1 (Optimal Control Problem for j). For given $x(k_m)$, $j \in \{1, \dots, j_{\max}\}$ and the λ -contractive set \mathcal{S} , find $u \in \mathcal{U}$ and $\varepsilon \in \mathbb{R}$ by solving the following problem:

$$\min_{u \in \mathcal{U}} \varepsilon, \quad (9)$$

subject to $\varepsilon \in [0, \lambda]$, and

$$(C.1) \quad A^{j'} x(k_m) + \sum_{i=1}^{j'} A^{i-1} B u \in \mathcal{X}, \quad \forall j' \in \{1, \dots, j\};$$

$$(C.2) \quad A^j x(k_m) + \sum_{i=1}^j A^{i-1} B u \in \varepsilon \varepsilon_x \mathcal{S};$$

where $\varepsilon_x = \Psi_{\mathcal{S}}(x(k_m))$. \square

In (C.2), $A^j x(k_m) + \sum_{i=1}^j A^{i-1} B u$ represents a state by applying the control input $u \in \mathcal{U}$ constantly for j time steps. Moreover, from the definition of the gauge function $\Psi_{\mathcal{S}}(\cdot)$ we have $x(k_m) \in \varepsilon_x \mathcal{S}$. Thus, Problem 1 aims to find the smallest possible scaled set $\varepsilon \varepsilon_x \mathcal{S}$, such that the state enters $\varepsilon \varepsilon_x \mathcal{S}$ (from $\varepsilon_x \mathcal{S}$) by applying a j -step constant control input. This means that a stabilizing controller is found under the transmission time interval j . The constraint in (C.1) implies that the state must remain inside \mathcal{X} while applying a j -step constant controller, which is imposed to guarantee the constraint satisfaction. Note that Problem 1 is a linear program,

¹If the iterative procedure in (6) does not converge in finite time, one can stop the procedure to obtain a $\bar{\lambda}$ -contractive set ($\lambda < \bar{\lambda} < 1$) in a finite number of iterations. In such case, we use $\bar{\lambda}$ (instead of λ) as the parameter to design the self-triggered strategies provided throughout the paper.

since all constraints imposed in (C.1), (C.2), as well as the cost in (9) are all linear.

For given $x(k_m)$ and j , let (u_j^*, ε_j^*) be a pair of optimal solutions obtained by solving Problem 1. From (C.2), the state enters $\varepsilon_j^* \varepsilon_x \mathcal{S}$ if u_j^* is applied constantly for j steps, i.e., $A^j x(k_m) + \sum_{i=1}^j A^{i-1} B u_j^* \in \varepsilon_j^* \varepsilon_x \mathcal{S}$, which means that we have $\Psi_{\mathcal{S}}(A^j x(k_m) + \sum_{i=1}^j A^{i-1} B u_j^*) \leq \varepsilon_j^* \Psi_{\mathcal{S}}(x(k_m))$, or

$$\Psi_{\mathcal{S}}(A^j x(k_m) + \sum_{i=1}^j A^{i-1} B u_j^*) - \Psi_{\mathcal{S}}(x(k_m)) \leq -(1 - \varepsilon_j^*) \Psi_{\mathcal{S}}(x(k_m)) \quad (10)$$

with $0 \leq \varepsilon_j^* \leq \lambda < 1$. Thus, $1 - \varepsilon_j^*$ represents how much the output of the gauge function (as a Lyapunov function candidate) decreases by applying the optimal controller u_j^* constantly for j steps. That is, if $1 - \varepsilon_j^*$ becomes larger (i.e., ε_j^* becomes smaller), then the state will be closer to the origin and a better control performance is achieved.

Now, consider solving Problem 1 for all $j \in \{1, \dots, j_{\max}\}$, which provides different solutions under different transmission time intervals. In the following, let $\mathcal{J}(x(k_m))$ be the set of indices (transmission time intervals) where Problem 1 provides a feasible solution. That is,

$$\mathcal{J}(x(k_m)) = \{j \in \{1, \dots, j_{\max}\} : \text{Problem 1 is feasible for } j\}. \quad (11)$$

Remark 1 (On the non-emptiness of $\mathcal{J}(x(k_m))$). If $x(k_m) \in \mathcal{S}$, there always exists $u \in \mathcal{U}$ such that $Ax(k_m) + Bu \in \lambda \varepsilon_x \mathcal{S} \subseteq \mathcal{X}$ holds from the properties of the λ -contractive set (see [12]). Thus, Problem 1 has a solution with $j = 1$ for any $x(k_m) \in \mathcal{S}$, and hence, $\mathcal{J}(x(k_m))$ is non-empty for any $x(k_m) \in \mathcal{S}$. \square

B. An overall algorithm

In this subsection an overall self-triggered algorithm is presented. After solving Problem 1 for all $j \in \{1, \dots, j_{\max}\}$, which provides the optimal (feasible) sets of solutions (u_j^*, ε_j^*) for all $j \in \mathcal{J}(x(k_m))$, the controller selects a suitable transmission time interval among them. The transmission time interval is selected such that both control performance and the communication load are taken into account. A more specific way to achieve this is given in the following overall strategy:

Algorithm 1 (Self-triggered strategy): For any transmission time k_m , $m \in \mathbb{N}$, do the following:

- 1) The plant transmits the current state information $x(k_m)$ to the controller.
- 2) Based on $x(k_m)$, the controller solves Problem 1 for all $j \in \{1, \dots, j_{\max}\}$, which provides the optimal (feasible) solutions (ε_j^*, u_j^*) for all $j \in \mathcal{J}(x(k_m))$.
- 3) The controller picks up an optimal index $j_m \in \mathcal{J}(x(k_m))$ by solving the following problem:

$$j_m = \underset{j \in \mathcal{J}(x(k_m))}{\text{argmax}} w_1(1 - \varepsilon_j^*)/j + w_2 j, \quad (12)$$

where $w_1, w_2 \geq 0$ represent given tuning weight parameters. Then, set $k_{m+1} = k_m + j_m$ and $u^*(k_m) = u_{j_m}^*$,

and the controller transmits $u^*(k_m)$ and k_{m+1} to the plant.

- 4) The plant applies $u^*(k_m)$ for all $k \in [k_m, k_{m+1})$. Set $m \leftarrow m + 1$, and then go back to step (1). \square

As shown in Algorithm 1, for each k_m we select the transmission time interval j_m according to (12). As described in the previous subsection, the term $(1 - \varepsilon_j^*)$ represents how much the output of the gauge function decreases by applying the optimal controller u_j^* constantly for j steps. Thus, the first term $(1 - \varepsilon_j^*)/j$ represents a *reward* due to the rate of decrease of the gauge function per one time step, and a better control performance can be achieved when this term becomes larger. On the other hand, from a self-triggered control viewpoint, less control updates will be obtained when control inputs can be applied constantly longer (i.e., when j becomes larger). Thus, the second part in (12) involves j to represent some reward for alleviating the communication load; as j gets larger, then we obtain less communication load and a larger reward is obtained.

Some remarks are in order regarding Algorithm 1:

Remark 2 (Relation to move-blocking MPC). The proposed algorithm is related to *move-blocking MPC* [19], in the sense that the optimal control inputs are restricted to be constant for some time period. Note that move-blocking MPC aims at reducing the computational complexity by decreasing the degrees of freedom of the optimal control problem [19]; the proposed approach, on the other hand, aims at reducing the *communication load* through the move-blocking technique, and the reduction of computation load is not a primary objective here. \square

Remark 3 (On the selection of j_{\max}). In Algorithm 1, the controller solves Problem 1 for all $j \in \{1, \dots, j_{\max}\}$ for each transmission time instant. While we can potentially achieve longer transmission intervals if j_{\max} is selected larger, the computation load of solving Problem 1 becomes heavier. Thus, in practical implementation, the user may carefully select a suitable j_{\max} by considering the trade-off between the communication load and the calculation time of solving the optimal control problem. \square

Theorem 1 (Stability). *Suppose that Assumption 3 holds, and Algorithm 1 is implemented. Then, it holds that $x(k) \rightarrow 0$ as $k \rightarrow \infty$.* \square

Proof. We first show that (12) is always feasible (i.e., we can always pick up a transmission time interval according to (12)), by proving that $\mathcal{J}(x(k_m))$ is non-empty for all $m \in \mathbb{N}$. By Assumption 3 we obtain $x(k_0) \in \mathcal{S}$ and thus $\mathcal{J}(x(k_0))$ is non-empty (see Remark 1). Since j_0 is obtained from (12), we have $j_0 \in \mathcal{J}(x(k_0))$ which means that Problem 1 has a feasible solution for $j = j_0$. Thus, from the constraint (C.2) in Problem 1, we obtain $x(k_1) = A^{j_0}x(k_0) + \sum_{i=1}^{j_0} A^{i-1}Bu^*(k_0) \in \mathcal{S}$, which means that $\mathcal{J}(x(k_1))$ is non-empty. By recursively following this argument, it is shown that $x(k_m) \in \mathcal{S}$ for all $m \in \mathbb{N}$, which follows that $\mathcal{J}(x(k_m))$ is non-empty for all $m \in \mathbb{N}$.

Now, it is shown that $x(k) \rightarrow 0$ as $k \rightarrow \infty$. Since $j_m \in$

$\mathcal{J}(x_m)$, $\forall m \in \mathbb{N}$, it holds from (C.2) in Problem 1 that:

$$\begin{aligned} \Psi_{\mathcal{S}}(x(k_{m+1})) &\leq \varepsilon_{j_m}^* \Psi_{\mathcal{S}}(x(k_m)) \\ &\leq \lambda \Psi_{\mathcal{S}}(x(k_m)). \end{aligned} \quad (13)$$

with $\lambda < 1$. Therefore, by regarding $\Psi_{\mathcal{S}}(\cdot)$ as a set-induced Lyapunov function candidate (see Lemma 1), the Lyapunov function is strictly decreasing and the state trajectory is asymptotically stabilized to the origin. This completes the proof. \square

Remark 4 (On achieving exponential stability). Although Theorem 1 states only asymptotic stability of the origin, exponential stability can be achieved by imposing an additional constraint when evaluating the reward function in (12). Specifically, let j_ℓ be chosen according to (12), subject to the constraint $\varepsilon_{j_\ell}^* \leq \lambda^j$. Indeed, imposing this constraint yields that $\Psi_{\mathcal{S}}(x(k_{m+1})) \leq \lambda^{j_m} \Psi_{\mathcal{S}}(x(k_m))$, $\forall m \in \mathbb{N}$ (instead of $\Psi_{\mathcal{S}}(x(k_{m+1})) \leq \lambda \Psi_{\mathcal{S}}(x(k_m))$) as in (13). Thus, we obtain

$$\Psi_{\mathcal{S}}(x(k_m)) \leq \lambda^{j_{m-1}} \Psi_{\mathcal{S}}(x(k_{m-1})) \leq \dots \leq \lambda^{k_m} \Psi_{\mathcal{S}}(x(0)),$$

which implies that exponential stability is guaranteed (see e.g., [12]). \square

IV. SELF-TRIGGERED CONTROL VIA EXPLICIT MAPPING Γ

In the previous section, the self-triggered strategy has been presented by solving Problem 1 for all $j \in \{1, \dots, j_{\max}\}$. However, solving Problem 1 for all candidates of transmission time intervals may lead to a high computation load, which may induce computational delays to transmit control samples to the plant. A more preferred approach may be that the transmission mapping $\Gamma : \mathcal{X} \rightarrow \{1, \dots, j_{\max}\}$ given in (3), which sends the state to the desired transmission time interval, is obtained *offline*. That is, with the mapping Γ provided explicitly offline, the next transmission time can be directly determined from the (current) state information, without having to solve Problem 1 for all $j \in \{1, \dots, j_{\max}\}$. The approach presented in this section is related to explicit MPC framework [15], in which an offline characterization of the control strategy (but here, the transmission time intervals) is given via state-space decomposition. A more specific formulation is given below.

A. Construction of Γ via state-space decomposition

In order to create the explicit mapping of Γ , we first *decompose* the contractive set \mathcal{S} into a finite number of disjoint subsets $\mathcal{S}_1, \dots, \mathcal{S}_L \subset \mathcal{S}$, i.e.,

$$\mathcal{S} = \bigcup_{\ell=1}^L \mathcal{S}_\ell, \quad (14)$$

where it holds that $\mathcal{S}_\ell \cap \mathcal{S}_{\ell'} = \emptyset$ for all $(\ell, \ell') \in \{1, \dots, L\} \times \{1, \dots, L\}$ ($\ell \neq \ell'$). Based on the decomposition, we will then assign a specific transmission time interval to each \mathcal{S}_ℓ , $\ell \in \{1, \dots, L\}$, so that the controller directly determines the next transmission time. Intuitively, if the state is located far from the origin we would like to assign a short transmission time interval to achieve stability of the origin (or achieve good control performance). In particular, in the case of unstable systems, applying a constant control signal may lead

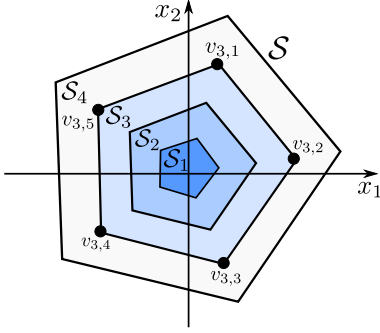


Fig. 2. Illustration of the decomposed subsets \mathcal{S}_ℓ , $\ell \in \{1, \dots, L\}$ (for the case $L = 4$) according to (16). In the figure, the pentagon represents the contractive set \mathcal{S} and the subsets $\mathcal{S}_1, \dots, \mathcal{S}_4$ are illustrated with different shades of blue.

to a divergence of states, especially if the state is far from the origin. On the other hand, if the state is close to the origin, a small control effort may be sufficient to stabilize the system. That is, assigning a long transmission time interval may be allowable to achieve both stability and communication reduction.

Motivated by the above intuition, we decompose the contractive set as follows. First, for a given $L \in \mathbb{N}_+$, define a set of scalars $\rho_1, \dots, \rho_L \in (0, 1]$, with

$$0 < \rho_1 < \rho_2 < \dots < \rho_{L-1} < \rho_L = 1. \quad (15)$$

Then, consider the following sequence of L sets $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_L \subset \mathcal{S}$:

$$\begin{aligned} \mathcal{S}_1 &= \rho_1 \mathcal{S}, \\ \mathcal{S}_\ell &= \rho_\ell \mathcal{S} \setminus \mathcal{S}_{\ell-1}, \quad \forall \ell \in \{2, \dots, L\}. \end{aligned} \quad (16)$$

The illustration of the sequence of sets is depicted in Fig. 2. It can be easily shown that the set sequence defined in (16) yields a decomposition as in (14), which satisfies the disjoint property as described above. Now, \mathcal{S} has been decomposed into a finite number of L sets $\mathcal{S}_1, \dots, \mathcal{S}_L$, to which we next assign suitable transmission time intervals. To this end, we formulate the following optimal control problem for each pair $(\ell, j) \in \{1, \dots, L\} \times \{1, \dots, j_{\max}\}$:

Problem 2 (Optimal Control Problem for (ℓ, j)). For a given pair $(\ell, j) \in \{1, \dots, L\} \times \{1, \dots, j_{\max}\}$, find $u_1, \dots, u_N \in \mathcal{U}$ and $\varepsilon \in \mathbb{R}$ by solving the following problem:

$$\min_{u_1, \dots, u_N \in \mathcal{U}} \varepsilon, \quad (17)$$

subject to $\varepsilon \in [0, \lambda]$, and

(C.1) For all $n \in \{1, \dots, N\}$, $j' \in \{1, \dots, j\}$,

$$A^{j'} v_{\ell, n} + \sum_{i=1}^{j'} A^{i-1} B u_n \in \mathcal{X}, \quad (18)$$

where $v_{\ell, n} = \rho_\ell v_n$, $n \in \{1, \dots, N\}$, $j' \in \{1, \dots, j\}$.

(C.2) For all $n \in \{1, \dots, N\}$,

$$A^j v_{\ell, n} + \sum_{i=1}^j A^{i-1} B u_n \in \varepsilon \rho_\ell \mathcal{S}. \quad (19)$$

Recall that v_n , $n \in \{1, \dots, N\}$ represent the vertices of \mathcal{S} (see (8)). Thus, $v_{\ell, n} = \rho_\ell v_n$, $n \in \{1, \dots, N\}$ represent the extreme points on the outer boundary of \mathcal{S}_ℓ (see the illustration in Fig. 2). Problem 2 for (ℓ, j) aims at finding a set of controllers u_1, \dots, u_N and a scalar ε , such that all the extreme points $v_{\ell, n}$, $n \in \{1, \dots, N\}$ can be driven into $\varepsilon \rho_\ell \mathcal{S}$ under the j -step constant control inputs. Note that Problem 2 is solved offline for all $\ell \in \{1, \dots, L\}$, $j \in \{1, \dots, j_{\max}\}$, since it can be solved by evaluating the extreme points $v_{\ell, n}$, $n \in \{1, \dots, N\}$ that are given offline.

Now, suppose that Problem 2 has a solution for (ℓ, j) , which provides optimal control inputs and a scalar denoted as $u_{\ell, j}^*(1), u_{\ell, j}^*(2), \dots, u_{\ell, j}^*(N) \in \mathcal{U}$, $\varepsilon_{\ell, j}^* \in [0, \lambda]$, respectively. The following lemma describes that the feasibility of Problem 2 for (ℓ, j) implies the existence of a stabilizing controller for all $x \in \mathcal{S}_\ell$:

Lemma 2. Suppose that Problem 2 finds a solution for (ℓ, j) . Then, for every $x \in \mathcal{S}_\ell$, there exists $u \in \mathcal{U}$ such that: (i) $A^{j'} x + \sum_{i=1}^{j'} A^{i-1} B u \in \mathcal{X}$ for all $j' \in \{1, \dots, j\}$; (ii) $A^j x + \sum_{i=1}^j A^{i-1} B u \in \varepsilon_{\ell, j}^* \varepsilon_x \mathcal{S}$ with $\varepsilon_x = \Psi_{\mathcal{S}}(x)$. \square

Proof. Since Problem 2 has a solution for (ℓ, j) , from (18) and (19) we obtain $A^{j'} v_{\ell, n} + \sum_{i=1}^{j'} A^{i-1} B u_{\ell, j}^*(n) \in \mathcal{X}$ for all $n \in \{1, \dots, N\}$, $j' \in \{1, \dots, j\}$, and $A^j v_{\ell, n} + \sum_{i=1}^j A^{i-1} B u_{\ell, j}^*(n) \in \varepsilon_{\ell, j}^* \rho_\ell \mathcal{S}$ for all $n \in \{1, \dots, N\}$.

Suppose $x \in \mathcal{S}_\ell$ and let $\varepsilon_x = \Psi_{\mathcal{S}}(x) \in [0, 1]$. Since $x \in \mathcal{S}_\ell \subseteq \rho_\ell \mathcal{S}$, we have $\varepsilon_x \leq \rho_\ell$. Moreover, since $x \in \varepsilon_x \mathcal{S}$, there exist $\lambda_n \in [0, 1]$, $n \in \{1, \dots, N\}$ such that $x = \varepsilon_x \sum_{n=1}^N \lambda_n v_n = (\varepsilon_x / \rho_\ell) \sum_{n=1}^N \lambda_n v_{\ell, n}$, $\sum_{n=1}^N \lambda_n = 1$, where we have used $v_{\ell, n} = \rho_\ell v_n$. Let $u \in \mathbb{R}^m$ be given by

$$u = \frac{\varepsilon_x}{\rho_\ell} \sum_{n=1}^N \lambda_n u_{\ell, j}^*(n) \in \mathcal{U}. \quad (20)$$

Then, for all $j' \in \{1, \dots, j\}$, we obtain

$$\begin{aligned} & A^{j'} x + \sum_{i=1}^{j'} A^{i-1} B u \\ &= \frac{\varepsilon_x}{\rho_\ell} \sum_{n=1}^N \lambda_n (A^{j'} v_{\ell, n} + \sum_{i=1}^{j'} A^{i-1} B u_{\ell, j}^*(n)) \in \frac{\varepsilon_x}{\rho_\ell} \mathcal{X} \subseteq \mathcal{X}, \end{aligned}$$

where the first inclusion holds since $A^{j'} v_{\ell, n} + \sum_{i=1}^{j'} A^{i-1} B u_{\ell, j}^*(n) \in \mathcal{X}$ for all $n \in \{1, \dots, N\}$, $j' \in \{1, \dots, j\}$, and the last inclusion holds since $\varepsilon_x \leq \rho_\ell$. Moreover, we have

$$\begin{aligned} & A^j x + \sum_{i=1}^j A^{i-1} B u \\ &= \frac{\varepsilon_x}{\rho_\ell} \sum_{n=1}^N \lambda_n (A^j v_{\ell, n} + \sum_{i=1}^j A^{i-1} B u_{\ell, j}^*(n)) \\ & \in \frac{\varepsilon_x}{\rho_\ell} \rho_\ell \varepsilon_{\ell, j}^* \mathcal{S} = \varepsilon_{\ell, j}^* \varepsilon_x \mathcal{S}, \end{aligned}$$

where the inclusion holds since $A^j v_{\ell, n} + \sum_{i=1}^j A^{i-1} B u_{\ell, j}^*(n) \in \rho_\ell \varepsilon_{\ell, j}^* \mathcal{S}$ for all $n \in \{1, \dots, N\}$.

Hence, we obtain $A^j x + \sum_{i=1}^j A^{i-1} B u \in \varepsilon_{\ell,j}^* \varepsilon_x \mathcal{S}$. This completes the proof. \square

Lemma 2 implies that for every $x \in \mathcal{S}_\ell$ there exists $u \in \mathcal{U}$ such that $\Psi_{\mathcal{S}}(A^j x + \sum_{i=1}^j A^{i-1} B u) \leq \varepsilon_{\ell,j}^* \Psi_{\mathcal{S}}(x)$ holds. Thus, this means that if $x \in \mathcal{S}_\ell$, then there exists a j -step stabilizing controller such that the output of the gauge function decreases. As mentioned previously in Section III-A, $(1 - \varepsilon_{\ell,j}^*)$ represents the decreasing rate of the gauge function. Thus, we can evaluate the control performance by $\varepsilon_{\ell,j}^*$ similarly to the self-triggered strategy presented in the previous section.

Now, suppose that for each ℓ we solve Problem 2 for all $j \in \{1, \dots, j_{\max}\}$. Let \mathcal{J}_ℓ be a set of indices (transmission time intervals) where Problem 2 has a feasible solution for ℓ , i.e.,

$$\mathcal{J}_\ell = \{j \in \{1, \dots, j_{\max}\} : \text{Problem 2 is feasible for } (\ell, j)\}. \quad (21)$$

Regarding the feasible set \mathcal{J}_ℓ , we obtain the following:

Lemma 3. \mathcal{J}_ℓ is non-empty for all $\ell \in \{1, \dots, L\}$.

The proof immediately follows from Definition 2 and is given in the Appendix. By evaluating the feasible solutions obtained above, we now assign to each \mathcal{S}_ℓ a suitable transmission time interval. In Algorithm 1, we presented the self-triggered strategy by determining the transmission interval according to the reward function in (12). Motivated by this, we similarly now consider the following assignment of the transmission time interval to \mathcal{S}_ℓ , by taking both control performance and communication load into account:

$$j_\ell^* = \operatorname{argmax}_{j \in \mathcal{J}_\ell} w_1(1 - \varepsilon_{\ell,j}^*)/j + w_2 j, \quad (22)$$

where $w_1, w_2 \geq 0$ denote the given tuning weights associated to each part of the reward similarly to (12). Note that in contrast to the previous self-triggered strategy where a suitable transmission time interval is obtained online, (22) is now given in an offline fashion. Suppose that we compute j_ℓ^* according to (22) for all $\ell \in \{1, \dots, L\}$. Then, each j_ℓ^* is assigned to \mathcal{S}_ℓ as the suitable transmission time interval. That is, if $x(k_m) \in \mathcal{S}_\ell$ for a certain transmission time k_m , the controller directly sets the next transmission time as $k_{m+1} = k_m + j_\ell^*$. Let $\mathcal{T} : \mathcal{S} \rightarrow \{1, \dots, j_{\max}\}$ be a mapping from \mathcal{S}_ℓ to the assigned transmission time interval, i.e., $j_\ell^* = \mathcal{T}(\mathcal{S}_\ell)$. Moreover, let $\mathcal{R} : \mathcal{X} \rightarrow \mathcal{S}$ be a mapping from x to the corresponding subset that x belongs to, i.e., $\mathcal{R}(x) = \mathcal{S}_\ell$, iff $x \in \mathcal{S}_\ell$, $\ell \in \{1, \dots, L\}$. Then, the overall transmission mapping $\Gamma : \mathcal{X} \rightarrow \{1, \dots, j_{\max}\}$ is given by

$$\Gamma(x) = (\mathcal{T} \circ \mathcal{R})(x). \quad (23)$$

B. An overall algorithm

Given the explicit transmission mapping obtained in (23), the overall self-triggered algorithm is now provided below:

Algorithm 2 (Self-triggered strategy via explicit mapping Γ): Given the explicit mapping Γ obtained by (23) and for any transmission time k_m , $m \in \mathbb{N}$, do the following:

- 1) The plant transmits the current state information $x(k_m)$ to the controller.
- 2) Based on $x(k_m)$, the controller sets the transmission time interval as $j_m = \Gamma(x(k_m))$. Then, set the next transmission time as $k_{m+1} = k_m + j_m$.
- 3) Suppose that $x(k_m) \in \mathcal{S}_{\ell_m}$ for some $\ell_m \in \{1, \dots, L\}$. For a given j_m obtained in step 2), the controller sets $u^*(k_m) = (\varepsilon_x / \rho_{\ell_m}) \sum_{n=1}^N \lambda_n u_{\ell_m, j_m}^*(n) \in \mathcal{U}$, where $\varepsilon_x = \Psi_{\mathcal{S}}(x(k_m))$ and $u_{\ell_m, j_m}^*(1), \dots, u_{\ell_m, j_m}^*(N) \in \mathcal{U}$ are the solution to Problem 2 for (ℓ_m, j_m) . Then, the controller transmits $u^*(k_m)$ and k_{m+1} to the plant.
- 4) The plant applies $u^*(k_m)$ for all $k \in [k_m, k_{m+1})$. Set $m \leftarrow m + 1$ and then go back to step 1). \square

As shown in Algorithm 2, in contrast to the first approach the controller only needs to compute the control input for a given transmission time interval from the explicit mapping Γ .

Remark 5 (The point location problem). For each transmission time k_m , the controller needs to find a suitable subset \mathcal{S}_ℓ such that $x(k_m) \in \mathcal{S}_\ell$ holds to determine the assigned transmission time interval according to (23). This problem, which we call *the point location problem*, can be easily solved by using the following property: we have $x(k_m) \in \mathcal{S}_1 \Leftrightarrow x(k_m) \in \rho_1 \mathcal{S}$, and for all $\ell \in \{2, \dots, L\}$,

$$x(k_m) \in \mathcal{S}_\ell \Leftrightarrow x(k_m) \notin \rho_{\ell-1} \mathcal{S}, x(k_m) \in \rho_\ell \mathcal{S}. \quad (24)$$

Hence, the point location problem can be solved by checking if $x(k_m) \in \rho_\ell \mathcal{S}$, $\ell \in \{1, \dots, L\}$ sequentially in that order, and takes the first index ℓ such that $x(k_m) \in \rho_\ell \mathcal{S}$ holds. \square

Theorem 2 (Stability). *Suppose that Assumptions 3 holds, and Algorithm 2 is implemented. Then, it holds that $x(k) \rightarrow 0$ as $k \rightarrow \infty$.* \square

Proof. We first show that selecting j_m as $j_m = \Gamma(x(k_m))$ is always feasible by proving that $x(k_m) \in \mathcal{S}$ for all $m \in \mathbb{N}$ (if $x(k_m) \notin \mathcal{S}$, the controller cannot determine j_m since the mapping is not defined). By Assumption 3, we obtain $x(k_0) \in \mathcal{S}$. To prove by induction, assume $x(k_m) \in \mathcal{S}$ for some $m \in \mathbb{N}_+$, and we will show $x(k_{m+1}) \in \mathcal{S}$. Suppose that $x(k_m) \in \mathcal{S}_{\ell_m} \subseteq \mathcal{S}$ for some $\ell_m \in \{1, \dots, L\}$, which means from (22) that $j_m = j_\ell^* = \Gamma(x(k_m))$ with $\ell = \ell_m$. Since $j_m = j_\ell^* \in \mathcal{J}_\ell$ with $\ell = \ell_m$, Problem 2 has a solution for the pair (ℓ_m, j_m) . Let $\varepsilon_{\ell_m, j_m}^* \in [0, \lambda]$ be the optimal ε as a solution to Problem 2 for (ℓ_m, j_m) . Then, from the proof of Lemma 2, setting $u^*(k_m) = (\varepsilon_x / \rho_{\ell_m}) \sum_{n=1}^N \lambda_n u_{\ell_m, j_m}^*(n)$ yields that $x(k_{m+1}) = A^{j_m} x(k_m) + \sum_{i=1}^{j_m} A^{i-1} B u^*(k_m) \in \varepsilon_x \varepsilon_{\ell_m, j_m}^* \mathcal{S} \subseteq \mathcal{S}$. Therefore, we have $x(k_m) \in \mathcal{S}$ for all $m \in \mathbb{N}$.

Now, it is shown that $x(k) \rightarrow 0$ as $k \rightarrow \infty$. Since $x(k_m) \in \mathcal{S}$ for all $m \in \mathbb{N}$, we obtain from Lemma 2 that

$$\begin{aligned} \Psi_{\mathcal{S}}(x(k_{m+1})) &\leq \varepsilon_{\ell_m, j_m}^* \Psi_{\mathcal{S}}(x(k_m)) \\ &\leq \lambda \Psi_{\mathcal{S}}(x(k_m)), \end{aligned} \quad (25)$$

with $\lambda < 1$. Therefore, by considering $\Psi_{\mathcal{S}}(\cdot)$ as a state-induced Lyapunov function candidate, the state trajectory is asymptotically stabilized to the origin. This completes the proof. \square

C. Comparisons between first and second approach

In this subsection we discuss both advantages and drawbacks of the second approach (Algorithm 2), by making some comparisons with the first one (Algorithm 1). As stated previously, the second approach is advantageous over the first one in terms of the computation load, since the transmission mapping Γ is given offline according to the procedure presented in the previous subsection. Note, however, that in the second approach, each j_ℓ^* is computed by solving Problem 2 that evaluates *the extreme points* of \mathcal{S}_ℓ (i.e., $v_{\ell,n}$, $n \in \{1, \dots, N\}$). This means that, while $x(k_m)$ is in the *interior* of \mathcal{S}_ℓ , which is not on some extreme point of \mathcal{S}_ℓ , there may exist some $j \in \mathcal{J}_\ell$ ($j \neq j_\ell^*$), such that applying $u^*(k_m) = (\varepsilon_x/\rho_\ell) \sum_{n=1}^N \lambda_n u_{\ell,j}^*(n) \in \mathcal{U}$ could yield a *larger* reward in (22) than the one obtained with $\varepsilon_{\ell,j_\ell^*}$. In this sense, the second approach yields a suboptimal (or conservative) solution compared with Algorithm 1 on the selection of transmission time intervals. This observation is also illustrated in the simulation example, where it is shown that Algorithm 1 achieves less communication load than Algorithm 2 for the case $w_1 = 0$ (for details, see Section V).

V. SIMULATION RESULTS

In this section we provide an illustrative example to validate our control schemes. The simulation was conducted on Matlab 2016a under Windows 10, Intel(R) Core(TM) 2.40 GHz, 8 GB RAM, using Multi-Parametric Toolbox (MPT3) to compute the λ -contractive set. We consider a control problem of a *batch reactor system*, which is often utilized as a benchmark in the NCSs community (see, e.g., [20]). The linearized model is given in the continuous-time domain as $\dot{x}(t) = A_c x(t) + B_c u(t)$, where A_c, B_c are given by

$$A_c = \begin{bmatrix} 1.380 & -0.208 & 6.715 & -5.676 \\ -0.581 & -4.290 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}.$$

The system is unstable having unstable poles 1.9911, 0.0633. We assume $\mathcal{X} = \{x \in \mathbb{R}^4 : \|x\|_\infty \leq 2\}$, $\mathcal{U} = \{u \in \mathbb{R}^2 : \|u\|_\infty \leq 5\}$ and $j_{\max} = 30$. We obtain the corresponding discrete-time system under a zero-order-hold controller with a sampling time interval 0.1, and the λ -contractive set \mathcal{S} is obtained with $\lambda = 0.99$ according to the procedure presented in Section II. Fig. 3 illustrates the resulting state trajectories and the corresponding control inputs by implementing Algorithm 1, starting from the initial state $x(k_0) = [1; 2; 2; 0.5]$ and the weights $(w_1, w_2) = (50, 1)$. The figure shows that the resulting state trajectories are asymptotically stabilized to the origin, and control inputs are updated only when necessary.

To analyse the effect of weights, we again simulate Algorithm 1 with $x(k_0) = [1; 2; 2; 0.5]$ under different selection of weights $(w_1, w_2) = (0, 1), (50, 1), (100, 1)$. We then compute the convergence time steps when the state enters the small region around the origin (the region satisfying $\|x\| \leq 0.001$),

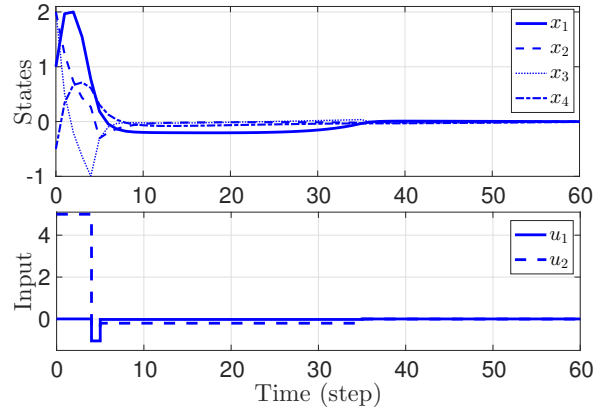


Fig. 3. Simulation results of state trajectories (upper) and the control inputs (lower) by applying Algorithm 1.

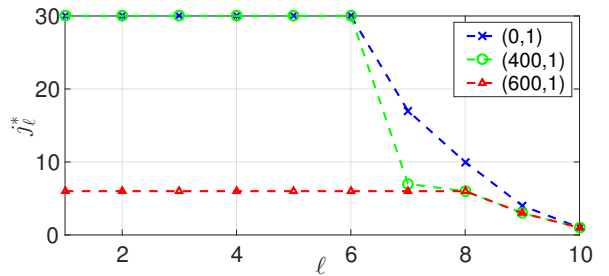


Fig. 4. j_ℓ^* as a function of ℓ with $(w_1, w_2) = (0, 1), (400, 1), (600, 1)$.

and the total number of transmission instances during the time period $k \in [0, 100]$. The results are shown in Table I. From the table, $(w_1, w_2) = (100, 1)$ achieves the fastest speed of convergence. This is due to the fact that by selecting w_1 larger, the reward for the control performance (i.e., the first term in (12)) is emphasized to be obtained. On the other hand, the number of transmission instances is the smallest for the case $(w_1, w_2) = (0, 1)$, which means that the smallest communication load is obtained. Therefore, it is shown that there exists a trade-off between control performance and communication load, and such trade-off can be regulated by tuning the weights (w_1, w_2) .

To implement the second proposal, we decompose \mathcal{S} into $L = 10$ subsets with $\rho_\ell = 0.1\ell$, $\ell \in \{1, \dots, 10\}$ and the mapping Γ is constructed according to Section IV. The selected transmission time intervals j_ℓ^* as a function of ℓ are illustrated in Fig. 4 under different selections of the weights $(w_1, w_2) = (0, 1), (400, 1), (600, 1)$. The figure shows that the transmission time interval tends to be smaller as the weight w_1 increases. This means that attaining control performance is em-

TABLE I
CONVERGENCE TIME AND NUMBER OF TRANSMISSION INSTANCES

(w_1, w_2)	(0,1)	(50, 1)	(100, 1)
Convergence (steps)	141	93	69
Transmission instances	5	6	10

TABLE II
NUMBER OF TRANSMISSION INSTANCES AND AVERAGE CALCULATION
TIME WITH $(w_1, w_2) = (0, 1)$.

	Algorithm 1	Algorithm 2
Transmission instances	5	7
Calculation time (sec)	2.98	0.83

phasized more than attaining communication reduction, if the weight for achieving the control performance w_1 is selected larger. In Fig. 4, we also illustrate the selected transmission time intervals for $(w_1, w_2) = (0, 1)$ (i.e., the blue dashed line). Since $w_1 = 0$, each j_ℓ^* corresponds to the *largest* transmission time interval such that Problem 2 becomes feasible for ℓ (i.e., the maximal index in the feasible set \mathcal{J}_ℓ). As shown in the figure, the feasible transmission time interval gets smaller as ℓ increases. Intuitively, this is due to that for unstable systems, applying a constant control input leads to a divergence of states, especially if the extreme points to solve Problem 2 are located far from the origin.

To illustrate the calculation time and the optimality of Algorithm 1 and 2 as discussed in Section IV-C, we again simulate the two algorithms with $(w_1, w_2) = (0, 1)$ and the initial state $x(k_0) = [1; 2; 2; 0.5]$. As previously described, setting $w_1 = 0$ corresponds to selecting the largest index in \mathcal{J}_ℓ . Table II illustrates the total number of transmission instances and the average calculation time to compute the control input for each transmission instance (i.e., the calculation time from step 2) to step 3) in Algorithms 1, 2). Here, both the total number of transmission instances and the average calculation time are computed over the time period $k \in [0, 100]$. From the table, Algorithm 1 achieves less communication load than Algorithm 2. As already discussed in Section IV-C, this is because of the sub-optimality of Algorithm 2; while Algorithm 1 solves Problem 1 based on the current state information online, Problem 2 is solved offline by evaluating the extreme points of the subsets. On the other hand, Algorithm 2 achieves less calculation time than Algorithm 1, as the transmission mapping Γ is explicitly given offline.

VI. CONCLUSION

In this note, we present two different types of self-triggered strategies based on the notion of set-invariance theory. In the first approach, we formulate an optimal control problem such that suitable transmission time intervals are selected by evaluating both the control performance and the communication load. The second approach aims to overcome the computation drawback of the first one by providing an offline characterization of the mapping Γ . In this approach, the state space is decomposed into a finite number of subsets, to which suitable transmission time intervals are assigned. Finally, the proposed self-triggered strategies are illustrated through a numerical example of controlling a batch reactor system.

REFERENCES

[1] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete time systems," in *Proceedings of American Control Conference (ACC)*, 2010, pp. 4719–4724.

- [2] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed \mathcal{L}_∞ gain and decentralized event-triggering," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1362–1376, 2011.
- [3] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030–2042, 2010.
- [4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Sokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.
- [5] E. Henriksson, D. E. Quevedo, E. G. W. Peters, H. Sandberg, and K. H. Johansson, "Multiple-loop self-triggered model predictive control for network scheduling and control," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2167–2181, 2015.
- [6] K. Kobayashi and K. Hiraishi, "Self-triggered model predictive control with delay compensation for networked control systems," in *Proceedings of the 38th Annual Conference of the IEEE Industrial Electronics Society*, 2012, pp. 3182–3187.
- [7] F. D. Brunner, W. P. M. H. Heemels, and F. Allgöwer, "Robust self-triggered mpc for constrained linear systems: A tube-based approach," *Automatica*, vol. 72, pp. 73–83, 2016.
- [8] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Distributed aperiodic model predictive control for multi-agent systems," *IET Control Theory and Applications*, vol. 9, no. 1, pp. 11–20, 2015.
- [9] —, "Self-triggered model predictive control for nonlinear input-affine dynamical systems via adaptive control samples selection," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 177–189, 2017.
- [10] —, "Event-triggered intermittent sampling for nonlinear model predictive control," *Automatica*, vol. 81, pp. 148–155, 2017.
- [11] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
- [12] —, "Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 428–433, 1994.
- [13] G. Bitsoris, "Positively invariant polyhedral sets of discrete-time linear systems," *International Journal of Control*, vol. 47, no. 6, pp. 1713–1726, 1988.
- [14] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic Control*, vol. 36, no. 9, pp. 1008–1020, 1991.
- [15] A. Bemporad, M. Morari, V. Dua and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [16] M. Hovd, S. Oлару, and G. Bitsoris, "Low complexity constraint control using contractive sets," in *The IFAC World Congress*, 2014, pp. 2933–2938.
- [17] S. Munir, M. Hovd, G. Sandou, and S. Oлару, "Controlled contractive sets for low-complexity constrained control," in *Proceedings of 2016 IEEE Conference on Computer Aided Control System Design (Part of 2016 IEEE Multi-Conference on Systems and Control)*, 2016, pp. 856–861.
- [18] M. S. Darup and M. Cannon, "On the computation of lambda-contractive sets for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1498–1504, 2017.
- [19] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.
- [20] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proceedings of the 51st IEEE Conference on Decision and Control (IEEE CDC)*, 2012, pp. 3270–3285.

APPENDIX

(Proof of Lemma 3) : We show that \mathcal{J}_ℓ is non-empty for all $\ell \in \{1, \dots, L\}$, by proving that Problem 2 is feasible for $(\ell, 1)$. Since $v_{\ell,1}, \dots, v_{\ell,N} \in \rho_\ell \mathcal{S}$, it holds that there exist a set of controllers $\tilde{u}_{\ell,1}, \dots, \tilde{u}_{\ell,N} \in \mathcal{U}$, such that $A v_{\ell,n} + B \tilde{u}_{\ell,n} \in \lambda \rho_\ell \mathcal{S} \subseteq \mathcal{X}$, $\forall n \in \{1, \dots, N\}$ from the properties of the λ -contractive set. Thus, this directly means from (18), (19) that Problem 2 has a feasible solution for $(\ell, 1)$, with $\varepsilon = \lambda$ and $u_{\ell,n} = \tilde{u}_{\ell,n}$, $\forall n \in \{1, \dots, N\}$. This completes the proof. \square