

# Event-Triggered Controller Synthesis for Dynamical Systems with Temporal Logic Constraints

Dipankar Maity and John S. Baras

**Abstract**—In this work, we propose an event-triggered control framework for dynamical systems with temporal logical constraints. Event-triggered control methodologies have proven to be very efficient in reducing sensing, communication and computation costs. When a continuous feedback control is replaced with an event-triggered strategy, the corresponding state trajectories also differ. In a system with logical constraints, such small deviation in the trajectory might lead to unsatisfiability of the logical constraints. In this work, we develop an approach where we ensure that the event-triggered state trajectory is confined within an  $\epsilon$  tube of the ideal trajectory associated with the continuous state feedback. At the same time, we will ensure satisfiability of the logical constraints as well.

## I. INTRODUCTION

Present control systems are typically a large network of heterogeneous components sharing some common resources and information, and with coordinated cooperation, they aim to achieve desired performance. These kinds of highly complex systems are ubiquitous in cyber-physical-systems (CPS), and also referred as networked-CPS. In many CPS, the controller synthesis is subjected to many logical constraints that arise due to presence of logical variables and reasoning among the subsystems. Recent studies on controller synthesis with linear temporal logic (LTL) have paved a way to design controllers for large complex systems with safety, synchronisation, and other logical constraints [1]–[3].

Novel formulations and efficient computational approaches have been proposed to mathematically formulate specifications such as trajectory sequencing, synchronization etc. Temporal logics such as linear temporal logic (LTL), computational tree logic (CTL), developed for model checking, have been widely accepted by the robotics community for the purpose of motion planning [4], [5]. Development of sophisticated model checking tools such as SPIN [6] and NuSMV [7] made it easier to synthesize controllers for such systems. As an alternative approach controller synthesis has been done using mixed integer linear programming [1].

Another challenge for the large connected CPS is that computation of the control law requires continuous sensing (often times distributed) and transmitting the sensed signals to the controllers. Consequently, the performance of such systems is generally determined by the availability of sensing power, bandwidth for continuous transmission and resources for fast computation. Therefore it will be beneficial if the same (with little tolerance) performance can be achieved with lesser intensive sensing and computing tasks.

The authors are with the Department of Electrical and Computer Engineering and The Institute for Systems Research, University of Maryland, College Park, USA. Email: dmaity@umd.edu, baras@umd.edu

To circumvent the problem of limited communication bandwidth or computing resources or sensing capability, researchers have developed techniques that require intermittent communications only at certain discrete time instances to perform the same task with minor performance degradation. These control methodologies are known in many forms e.g. event-triggered [8], self-triggered [9], or periodic control [10]. These control strategies do not require the state information  $x(t)$  for all time  $t$ , rather they sample  $x(t)$  intermittently depending on the systems' performance criterion [11], [12]. These techniques have been proven to be efficient for large scale inter-connected systems to reduce communication and sensing operations.

In this work, we study the temporal logic based controller synthesis problem in an event-triggered framework. We consider a controller synthesis problem for a given control affine nonlinear system and the objective is to design an event triggered controller for that system with logical constraints. We assume the logical constraints can be represented using temporal logic and its propositional calculus.

Existing literature results show that the trajectory of an event triggered system deviates from the nominal system as a consequence of limited communication [11], [12]. Although, the continuous feedback system satisfies the logical constraints, now with an event triggered controller we have no guarantee that the logical constraint over the event-triggered trajectory will be satisfied as well.

We show that suitably modifying the given logical constraints, and creating stricter constraints will make the event-triggered trajectory satisfy the original logical constraint provided we can synthesize a continuous controller to satisfy the stricter constraint. The stricter logical constraint is often times known as robust logical constraint [13] since any perturbed trajectory (within some  $\epsilon$  bound) will still satisfy the constraint. We adopt this notion of robustness in this work. In the next stage, we design an event-triggered controller ensuring that the event-triggered controller confines the trajectory within an  $\epsilon$ -tube around the actual trajectory. Further, we show the effects of delay (in transmitting the measurement to the controller) on the performance. The analysis shows that if the delay is bounded by a certain quantity, which depends on the physical parameters of the plant and the controller, then the delayed system will be able to perform similar to the delay-free system without further modification in design.

In Section II, we formally describe the problem and our two-step approach towards the problem. Section III provides preliminary background on the temporal logic and

construction of  $\epsilon$ -robust logic formulae. We design an event triggered controller for this problem in Section IV and study the effects of delay on such an event triggered controller. Finally, we illustrate the application of our framework using two examples in Section V.

## II. PROBLEM FORMULATION

Let us consider the input-affine nonlinear state space model as given in (1).

$$\begin{aligned} \dot{x} &= f_0(t, x) + \sum_{i=1}^m f_i(t, x) \cdot u_i \\ x(t_0) &= x_0. \end{aligned} \quad (1)$$

where  $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ ,  $u_i(t) \in \mathbb{R}$  is the  $i$ -th control input. The trajectory of the dynamical system starting at  $t_0$  under application of some control  $u = [u^1, \dots, u^m]$  is denoted as  $x^{u, x_0}[t_0]$ . Similarly the trajectory starting from any arbitrary point  $(t, x)$  is represented as  $x^{u, x}[t]$ . The objective of this work is to design an event-triggered controller  $u(\cdot)$  that ensures satisfiability of a temporal logical constraints ( $\varphi$ ).

By  $x^{u, x_0}[t_0] \models \varphi$  we denote that the trajectory of the dynamics (1) under input  $u$  satisfies the logical constraint  $\varphi$ . Similarly,  $x^{u, x_0}[t_0] \not\models \varphi$  denotes that the trajectory does not satisfy the logical constraint. In this work we focus on the real time linear temporal logics [4], [14] which have been proven to be very effective for expressing logical constraints in dynamical systems [1]. In the following, we formally pose the problem that we aim to solve in this work.

**Problem 2.1:** Given an input-affine dynamics (1) and a logical constraint ( $\varphi$ ) on the trajectory of the system, design an event-triggered framework to generate the control  $u(t)$  such that the event-triggered trajectory satisfies  $\varphi$ , i.e.

$$\begin{aligned} \text{find } & u \\ \text{subject to } & u \in \mathcal{U}^e \\ & \dot{x} = f_0(t, x) + \sum_{i=1}^m f_i(t, x) \cdot u_i \\ & x^{u, x_0}[t_0] \models \varphi \end{aligned} \quad (2)$$

where  $\mathcal{U}^e$  denotes the set of event based control strategies. In this work we do not impose any restriction on the event-triggered framework other than the exclusion of Zeno behavior [15]. We provide sufficient conditions which, if satisfied, ensure that the trajectory of the designed event-triggered system will satisfy the logical constraint  $\varphi$ .

While temporal logic can express various types of logical constraints (see Section III), it comes with a cost that verifying whether a trajectory satisfies the logical constraints is PSPACE-complete [14]. Therefore synthesis of a controller is a hard problem in its own right, and synthesis of an event-based controller is harder for obvious reasons. However, there are some proposed techniques which can generate a (feedback) controller that satisfies the temporal logic constraints, see for example [16], [17].

Therefore, we divide the original problem into two sub-problems: Problem 2.2 and Problem 2.3.

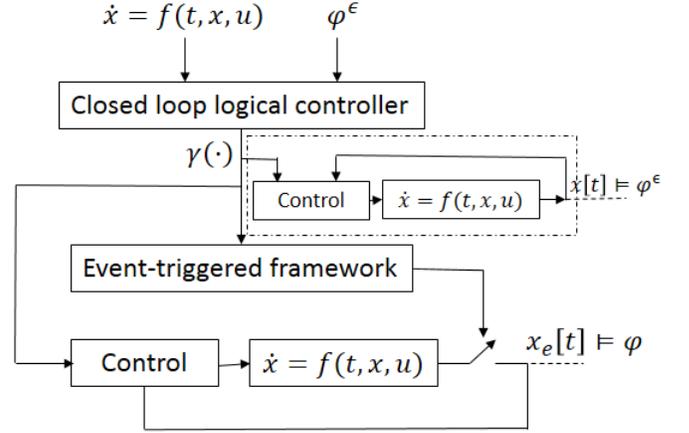


Fig. 1. Schematic of two-step event-triggered controller synthesis with logical constraints. The top most block receives the dynamics equations and the task specification to generate a feedback control  $\gamma(\cdot)$ . The dashed box shows that  $\varphi^\epsilon$  is satisfied by the system if  $\gamma(\cdot)$  is used. The event-trigger framework block receives the structure of the continuous feedback  $\gamma(\cdot)$  and controls the switch to regulate the number of transmission to the controller. The controller in the last layer computes the event-triggered control from the knowledge of  $\gamma(\cdot)$  and the measurements received.

**Problem 2.2:** Given the dynamics (1), design a feedback controller  $u_i(t) = \gamma_i(t, x(t))$  such that  $x^{\gamma, x_0}[t_0] \models \varphi^\epsilon$ .

where  $\varphi^\epsilon$  is another logical constraint derived from  $\varphi$ .  $\varphi^\epsilon$  is a stricter constraint than  $\varphi$  in the sense that  $x^{\gamma, x_0}[t_0] \models \varphi^\epsilon$  implies  $\xi[t_0] \models \varphi$  for all piecewise continuous curves  $\xi(\cdot) : [t_0, +\infty) \rightarrow \mathbb{R}^n$  such that  $\sup_{t \in [t_0, +\infty)} \|x(t) - \xi(t)\| \leq \epsilon$ . In the following sections we will explicitly explain how  $\varphi^\epsilon$  is related to  $\varphi$  for a given  $\epsilon \geq 0$ .

**Problem 2.3:** For all  $\epsilon > 0$ , given the dynamics (1) and a feedback control  $\gamma(t, x(t))$ , design an event-triggered controller  $\gamma^\epsilon(t, x(\tau_k))$  such that the trajectory of the event-triggered system  $x^{\gamma^\epsilon, x_0}[t_0]$  remains within an  $\epsilon$  neighborhood of the ideal trajectory associated with the feedback closed-loop system. As  $\epsilon \rightarrow 0$ ,  $\gamma^\epsilon(t, \cdot) \rightarrow \gamma(t, \cdot)$  pointwise  $\forall t$ .

Therefore, in this two step approach, we first design a feedback controller for satisfying the  $\epsilon$ -strict constraint  $\varphi^\epsilon$  (for some  $\epsilon > 0$ ). In the next stage, we use an event-triggering mechanism which provides sufficient condition(s) for ensuring that the trajectory of the event-triggered system will be within an  $\epsilon$  neighborhood of the actual feedback trajectory pointwise, i.e.  $\|x_e(t) - x(t)\| \leq \epsilon$  ( $x_e$  is the event-triggered trajectory and  $x$  is the ideal feedback trajectory) for all  $t$ . In Figure 1, we present our schematic for event-triggered controller synthesis using the proposed two-step approach. From this point onward, we will suppress the control and initial state in denoting a trajectory when these are apparent from the context i.e. we will represent  $x^{u, x_0}[t_0]$  as  $x[t_0]$  etc.

## III. PROPOSITIONAL TEMPORAL LOGIC

Like other families of propositional logic, temporal logic over the reals also requires a set of propositional variables  $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ . Associated with each propositional

variable  $\pi_i$ , there is a labelling function  $\mathcal{L}_i : \mathcal{X} \rightarrow \{0, 1\}$  which denotes whether the proposition  $\pi_i$  is true at some point in  $\mathcal{X}$ . Therefore,  $\Pi$  divides  $\mathcal{X}$  into subsets and assigns  $\pi_i$  with each of the subsets. A formula of a propositional logic is defined over a Boolean signal and in our case,  $\mathcal{L}_i(\cdot)$  maps the  $\mathbb{R}^n$  valued signal ( $x(t)$ ) to a Boolean signal. For example,  $\pi_1$  could be associated with the ball of radius 1 at the origin of  $\mathcal{X} = \mathbb{R}^2$ . Then  $\mathcal{L}_1(s) = 1$  for all  $x \in B_0(1)$  and 0 otherwise, where  $B_x(\delta) = \{y \in \mathcal{X} \mid \|y - x\|_2 \leq \delta\}$  is a ball of radius  $\delta$  centered at  $x$ . Note that it is not necessary that the regions associated with  $\pi_i$  are non-overlapping. We will use the notation  $\pi_i \cong \mathcal{X}_i (\subseteq \mathcal{X})$  to denote  $\mathcal{L}_i(x) = 1$  for all  $x \in \mathcal{X}_i$  and  $\mathcal{L}_i(x) = 0$  for all  $x \in \mathcal{X} \setminus \mathcal{X}_i$  (basically the indicator function of the set  $\mathcal{X}_i$ ). At this point, it should be noted that any algebraic constraint on  $x$  of the form  $G(x) \leq 0$  could be associated with a proposition  $\pi$  such that  $\pi \cong \{x \in \mathcal{X} \mid G(x) \leq 0\}$ .

However, the power of temporal logic is beyond capturing these algebraic constraints. Let us first provide an informal overview of the capability of the logic and then formally state the syntax and semantics of the logic. The RTL (Temporal Logic over Reals) [18] formulae are built on the propositional variables  $\Pi$  with the use of usual logical operators  $\neg$  (negation),  $\vee$  (conjunction) and  $\wedge$  (disjunction), and some special temporal operators e.g.  $\mathbf{U}$  (until),  $\diamond$  (eventually),  $\square$  (always) and other operators that could be derived from the mentioned operators. For example, the formula  $\diamond \square \pi$  (read as ‘‘Eventually Always in  $\pi$ ’’) where  $\pi \cong \mathcal{X}_i$ , when satisfied by a trajectory of the dynamics (1), means that eventually the trajectory enter the region  $\mathcal{X}_i$  and stay there for all future times. Similarly,  $(\neg \pi_1 \wedge \neg \pi_2 \wedge \neg \pi_3) \mathbf{U} \pi_4$  states the rule that region  $\mathcal{X}_4$  must be reached while avoiding regions  $\mathcal{X}_i$  for  $i = 1, 2, 3$  ( $\pi_j \cong \mathcal{X}_j$ ). Although the satisfaction of the formula tells us that the state trajectory will reach  $\mathcal{X}_4$ , it does not provide any interval of time within which it will reach the destination. This limitation can easily be circumvented by the traditional augmentation of a new state  $x_{n+1} = t$ .

Adding an extra equation  $\dot{x}_{n+1} = 1$  with  $x_{n+1}(t_0) = t_0$  in the dynamics (1), we can pose time dependent constraints as well. In this case, the augmented space is  $\mathcal{X} \times [t_0, T)$  ( $T$  could be  $+\infty$  for an infinite horizon problem). The formula  $\square(\pi_1 \vee \pi_2)$  where  $\pi_1 \cong \{(x, t) \mid x \in \mathcal{X}, t_0 \leq t < 6\}$  and  $\pi_2 \cong \{(x, t) \mid x \in \mathcal{X}_i, 5 \leq t\}$  requires the trajectory to enter the region  $\mathcal{X}_2$  no earlier than  $t = 5$  and the trajectory should remain within  $\mathcal{X}_2$  for all  $t \in [6, \infty)$ . The formula also mentions that the trajectory will be in  $\mathcal{X}_1$  when it is not in  $\mathcal{X}_2$ . Therefore, with this state-space augmentation, all the properties related to the timing aspect of a trajectory of the system (1) can be expressed.

**Definition 3.1:** *The syntax of RTL formulas are defined according to the following grammar rules:*

$$\phi ::= \top \mid \pi \mid \neg \phi \mid \phi \vee \phi \mid \phi \mathbf{U} \phi \mid \phi \mathbf{R} \phi$$

where  $\pi \in \Pi$ ,  $\top$  and  $\perp (= \neg \top)$  are the Boolean constants true and false respectively.  $\mathbf{R}$  symbolizes the *Release* operator. Other temporal logic operators can be represented using the grammar in definition 3.1 e.g. eventually ( $\diamond \phi = \top \mathbf{U} \phi$ ), always ( $\square \phi = \neg \diamond (\neg \phi)$ ) etc.

If  $x[t_0]$  denotes a trajectory starting at time  $t_0$ , the semantics of the grammar in Definition 3.1 is given as follows:

**Definition 3.2:** *The semantics of any formula  $\phi$  over the trajectory  $x[t_0]$  is recursively defined as:*

$$\begin{aligned} x[t_0] \models \pi & \text{ iff } \mathcal{L}_\pi(x(t_0)) = 1 \\ x[t_0] \models \neg \pi & \text{ iff } \mathcal{L}_\pi(x(t_0)) = 0 \\ x[t_0] \models \phi_1 \vee \phi_2 & \text{ iff } x[t_0] \models \phi_1 \text{ or } x[t_0] \models \phi_2 \\ x[t_0] \models \phi_1 \wedge \phi_2 & \text{ iff } x[t_0] \models \phi_1 \text{ and } x[t_0] \models \phi_2 \\ x[t_0] \models \phi_1 \mathbf{U} \phi_2 & \text{ iff } \exists s \geq t_0 \text{ s.t. } x[s] \models \phi_2 \text{ and } \forall t_0 \leq s' < s, x[s'] \models \phi_1. \\ x[t_0] \models \phi_1 \mathbf{R} \phi_2 & \text{ iff } \forall s \geq t_0 x[s] \models \phi_2 \text{ or } \exists s' \text{ s.t. } t_0 \leq s' < s, x[s'] \models \phi_1. \end{aligned}$$

More details on RTL grammar and semantics can be found in [14], [4].

### A. Construction of $\epsilon$ -Robust Formula

As described in Section II, the motivation behind constructing an  $\epsilon$ -robust formula  $\varphi^\epsilon$  is that any trajectory satisfying the stricter formula  $\varphi^\epsilon$  is robust in the sense that any perturbed trajectory with less than  $\epsilon$  perturbation will also satisfy the original constraint  $\varphi$ .

The idea is as follows: if the trajectory needs to visit a region  $\pi_i (\cong \mathcal{X}_i)$ , then we push the boundary of  $\mathcal{X}_i$  inwards by amount  $\epsilon$  and denote this new set (and proposition) by  $\mathcal{X}_i^\epsilon (\pi_i^\epsilon)$ . Similarly if the trajectory needs to avoid some region  $\pi_j (\cong \mathcal{X}_j)$  then the boundary of  $\mathcal{X}_j$  is expanded outwards by an amount  $\epsilon$ .

The RTL syntax presented in Definition (3.1) is in negative normal form (NNF) [19], and this enables us to detect which regions must be avoided by noting the presence of the negation ( $\neg$ ) operator immediately before the corresponding propositions  $\pi_i$ .

Note that, by our definition  $\pi_i \cong \mathcal{X}_i$  and  $\neg \pi_i \cong \mathcal{X} \setminus \mathcal{X}_i$ .

**Definition 3.3:** *In a given metric space  $(\mathcal{X}, \rho)$  the open ball centered at  $x \in \mathcal{X}$  of radius  $r$  is defined as  $B_x(r) = \{y \in \mathcal{X} \mid \rho(x, y) < r\}$ . Let  $\epsilon > 0$  be a given parameter, then the  $\epsilon$ -contraction of the set  $\mathcal{Y} \subseteq \mathcal{X}$  is denoted by  $\mathcal{Y}^\epsilon = \{y \in \mathcal{Y} \mid B_y(\epsilon) \subseteq \mathcal{Y}\}$ .*

*Similarly, the  $\epsilon$ -expansion of the set is denoted by  $\mathcal{Y}^{-\epsilon} = \{x \in \mathcal{X} \mid \exists y \in \mathcal{Y}, x \in B_y(\epsilon)\}$ .*

For a RTL formula  $\varphi$ , the  $\epsilon$ -robust formula is constructed as follows:

- 1) replace each  $\pi_i (\cong \mathcal{X}_i)$ , which is not preceded by any negation, by  $\pi_i^\epsilon$  where  $\pi_i^\epsilon \cong \mathcal{X}_i^\epsilon$ .
- 2) any  $\pi_j$  that is preceded by a negation ( $\neg$ ) should be replaced by  $\pi_j^{-\epsilon} \cong \mathcal{X}_j^{-\epsilon} \cup (\mathcal{X} \setminus \mathcal{X}^\epsilon)$ . When  $\mathcal{X} = \mathcal{X}^\epsilon$  for all finite  $\epsilon > 0$  (e.g.  $\mathcal{X} = \mathbb{R}^n$ ),  $\pi_j^{-\epsilon} \cong \mathcal{X}_j^{-\epsilon}$ .

In a similar way one can define  $\epsilon$ -robust formulas over the space  $\mathcal{X} \times [t_0, T)$ . However, for this work we will restrict ourselves to the robustness only in  $\mathcal{X}$  space.

**Proposition 3.4:** *Let the trajectory  $x[t_0]$  satisfies the RTL formula  $\varphi^\epsilon$  for some  $\epsilon > 0$ . Then for all  $\delta \leq \epsilon$  and for any curve  $y(\cdot) : [t_0, T) \rightarrow \mathcal{X}$  such that  $\sup_t \rho(x(t), y(t)) \leq \delta$ ,  $y[t_0] \models \varphi$ .*

The above proposition can be proved inductively; interested readers may see [4] for a proof.

**Remark 3.5:** In order to construct a  $\pi_i^\epsilon$  from  $\pi_i$ ,  $\mathcal{X}_i$  must have a non-empty interior.

Let us define the radius of a set  $\mathcal{X}_i$  in the following way  $r(\mathcal{X}_i) = \sup\{r \mid \exists x \in \mathcal{X}_i, B_x(r) \subseteq \mathcal{X}_i\}$ . Therefore for each  $\pi_i$ ,  $\pi_i^\epsilon$  is well defined for  $\epsilon \leq r(\mathcal{X}_i)$ , for  $\epsilon > r(\mathcal{X}_i)$ ,  $\pi_i^\epsilon \cong \emptyset$ . For the subsequent section we will implicitly assume that  $\min_i \{r(\mathcal{X}_i)\} > 0$  and moreover, each of these sets,  $\mathcal{X}_i$  is a polyhedron.

Therefore, to solve problem 2.2, our goal will be to design a controller that satisfies the robust RTL formula  $\varphi^\epsilon$ . The power of using these temporal logic formulae is that each formula can be represented by an equivalent (Büchi) automaton. Satisfaction of an RTL formula is equivalent to finding a path from the initial state to one of the accepting states of the automaton. The construction of such automata can be done automatically using the available tools SPIN and NuSMV [6], [7]. The dynamics (1) can be represented as a finite transition system (FTS), where the transitions are performed by selecting the control  $u$ . Thus, the dynamics constraint and logical constraint in (2) can jointly be represented by forming a product of the automata and the FTS. More detail on such construction and controller synthesis can be found in our earlier work [20]. To keep the paper short, we spare the details here, however, interested reader may see, for example, [21], [17].

#### IV. EVENT TRIGGERED CONTROLLER SYNTHESIS

In the previous section it is presented how the control inputs can be generated for the dynamics (1) so that the trajectory of the system satisfies the logical constraint  $\varphi^\epsilon$  for some  $\epsilon > 0$ . This section will focus on designing an event-triggered controller that will replace the feedback controller designed to satisfy  $\varphi^\epsilon$  in such a way that the trajectory of the event-triggered system will remain within  $\epsilon$  distance of the ideal feedback trajectory.

Let us denote the controller  $u(t) = \gamma(x(t))$  that achieves the satisfaction of  $\varphi^\epsilon$ . The closed-loop dynamics are:

$$\begin{aligned} \dot{x} &= f_0(t, x) + \sum_{i=1}^m f_i(t, x) \gamma_i(x) \quad (3) \\ x(t_0) &= x_0 \end{aligned}$$

Let  $x[t_0]$  denote the trajectory of the above closed loop system. We make the following assumptions on the system (3).

**Assumptions 4.1: (A1)**  $\gamma_i(\cdot)$  and for all  $t$ ,  $f_i(t, \cdot)$  are Lipschitz functions with Lipschitz constants  $L_\gamma^i, L_f^i$  respectively, for all  $i = 0, 1, 2, \dots, m$ .

**(A2)** For all  $i = 1, 2, \dots, m$  and  $\forall t$ ,  $f_i(t, x) \gamma_i(x)$  and  $f_0(t, x)$  are continuously differentiable functions w.r.t  $x$  with continuous first derivative.

In event-triggered framework, the controller is designed to be:

$$\gamma_i^\epsilon(t) = \gamma_i(x(t_k)) \quad \forall t \in [t_k, t_{k+1}) \quad (4)$$

where  $t_k$ s are the event-triggering times. An event-generator needs to be designed that will produce the  $t_k$  in certain way that is explained in the following.

Let us denote the event-triggered closed loop system as  $x_e(t)$  and the corresponding trajectory as  $x_e[t_0]$ . Thus,

$$\dot{x}_e = f_0(t, x_e) + \sum_{i=1}^m f_i(t, x_e) \gamma_i^\epsilon(x_e(t_k)) \quad (5)$$

$$x_e(t_0) = x_0. \quad (6)$$

Let us define the error  $e(t) = x(t) - x_e(t)$ . Note that the event-triggered controller  $\gamma_i^\epsilon(t, \cdot)$  for all  $t$  is an approximation of the ideal feedback controller  $\gamma_i(t, \cdot)$  by piecewise constant functions. Therefore, designing the  $\gamma_i$  at first makes the problem tractable for generating the event-triggered controller.

The dynamics of  $e(t)$  is given as:

$$\begin{aligned} \dot{e} &= F(t, x) - F(t, x_e) + \sum_{i=0}^m f_i(t, x_e) (\gamma_i^\epsilon(x_e(t_k)) - \gamma_i(x_e(t))) \\ e(t_0) &= 0, \end{aligned} \quad (7)$$

where  $F(t, x) = f_0(t, x) + \sum_{i=1}^m f_i(t, x) \gamma_i(x)$ .

**Assumption 4.2:**  $y(t)$  is exponentially stable with

$$\dot{y} = A(t)y \quad (8)$$

where  $A(t) = \left. \frac{\partial F(t, x)}{\partial x} \right|_{x=x(t)}$  and  $x(t)$  is the trajectory of (3).

We can write,

$$\dot{e} = A(t)e + \tilde{f}(t, x_e, e)e + \delta(t) \quad (9)$$

where  $\tilde{f}(t, x_e, e)e = F(t, x) - F(t, x_e) - A(t)e$  and  $\delta(t) = \sum_{i=0}^m f_i(t, x_e) (\gamma_i^\epsilon(x_e(t_k)) - \gamma_i(x_e(t)))$ .

Using Assumption 4.1 (A2),

$$F(t, x+h) = F(t, x) + \int_{s=0}^1 dF(t, x+sh) ds$$

where  $dF(t, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a linear map which is the derivative of the map  $F(t, \cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . Selecting  $h = x_e - x = -e$ , we obtain:

$$F(t, x_e) = F(t, x) - \int_0^1 dF(t, x-se) ds.$$

Therefore,

$$\tilde{f}(t, x_e, e) = \int_{s=0}^1 dF(t, x_e(t) + (1-s)e(t)) ds - A(t).$$

Also  $\tilde{f}(t, x_e, 0) = 0$ .

Since  $\dot{y} = A(t)y$  is the linearization of the system  $\dot{e} = A(t)e + \tilde{f}(t, x_e, e)e$  around  $e = 0$ , we can say that  $\dot{e} = A(t)e + \tilde{f}(t, x_e, e)e$  is locally exponentially stable due to Assumption 4.2. As a consequence of the Lyapunov converse theorem [22, Theorem 4.14], we have a (local) quadratic Lyapunov function that satisfies:

$$c_1 \|e\|^2 \leq V(t, e) \leq c_2 \|e\|^2 \quad (10)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial e} (A(t) + \tilde{f}(t, x_e, e))e \leq -c_3 \|e\|^2 \quad (11)$$

$$\left\| \frac{\partial V}{\partial e} \right\| \leq c_4 \|e\| \quad (12)$$

**Proposition 4.3:** For all  $t$ ,

$$\|e(t)\|_2 \leq \frac{c_4}{2c_1} \int_{t_0}^t e^{-(t-s)c_3/2c_2} \|\delta(s)\|_2 ds \quad (13)$$

*Proof:* A detailed proof of this can be found in our earlier work [12, Theorem 2.6]. The proposition is due to the BIBO (bounded input bounded output) stability of an exponential stable system. ■

Note that at each  $t_k$ ,  $\delta(t_k) = 0$ . We can bound  $\|\delta(t)\|$  to ensure a bound on  $\|e(t)\|$ .

**Remark 4.4:** Without computing  $x(t)$  real-time,  $e(t)$  can be bounded by observing the signal  $\delta(t)$  which depends only on  $x_e$ . Moreover, due to the Lipschitz assumptions on  $f_i(t, \cdot)$  and  $\gamma_i(\cdot)$ , it is sufficient to only monitor the difference signal  $x(t) - x(t_k)$ .

From (13), we have the sufficiency condition that  $\sup_t \|\delta(t)\| \leq \epsilon_1 = \frac{c_1 c_3}{c_2 c_4} \epsilon$  ensures  $\sup_t \|e(t)\| \leq \epsilon$ .

We propose the following event-trigger function:

$$g(t) = \frac{c_1 c_3}{c_2 c_4} \epsilon - \|\delta(t)\|. \quad (14)$$

An event is generated whenever  $g(t) \leq 0$  and the state value at that time ( $x(t_k)$ ) is sent to the controller. The set of triggering times is denoted by  $\mathcal{T} = \{t_1, t_2, \dots, t_k, \dots\}$  such that  $g(t_k) = 0$  and  $g(t) < 0$  otherwise.

There could be other event-triggering functions that can also ensure bounded error  $e(t)$ . In this paper, we consider (14) to carry out the analysis further and to perform the simulations.

**Lemma 4.5:** For all  $t$ ,

$$\|\delta(t)\| \leq \alpha \|x_e(t) - x_e(t_k)\|^2 + \beta(t) \|x_e(t) - x_e(t_k)\|. \quad (15)$$

for some  $\alpha, \beta(t) > 0$ .  $t_k$  is the latest triggering time at time  $t$ .

Interested readers may see [23] for a proof of Lemma 4.5.

In [12] a different bound on  $\delta$  was derived. There it was shown that:

$$\|\delta(t)\| \leq \kappa(t) \|(x_e(t_k)) - (x_e(t))\|. \quad (16)$$

$\kappa(t) = \max_i \{L_{\gamma}^i\} \sup_{x \in \Omega} \sum_{i=1}^m \|f_i(t, x)\|$ . Where the trajectory  $x(t)$  of (3) is bounded in the domain  $\Omega^\epsilon$ .

Comparing (16) with (15), we notice that the former is bounded linearly w.r.t.  $\|(x_e(t_k)) - (x_e(t))\|$  whereas the later is bounded by a quadratic form of  $\|(x_e(t_k)) - (x_e(t))\|$ . In most of the practical applications  $\epsilon \ll 1$  and hence  $\|(x_e(t_k)) - (x_e(t))\|$  is required to keep smaller than  $\epsilon$  (see Proposition 4.6). Therefore,  $\|(x_e(t_k)) - (x_e(t))\|^2$  can be bounded by  $\|(x_e(t_k)) - (x_e(t))\|$  with proper coefficient. Furthermore, the presence of sup operator over the whole domain  $\Omega^\epsilon$  in (16) implies that  $\kappa(t) \geq \beta(t)$  (in general  $\kappa(t) \gg \beta(t)$ ). Therefore, (15) could be a better approximation of  $\|\delta(t)\|$ .

**Proposition 4.6 (sufficiency):** For all  $t$ ,

$$\|x_e(t) - x_e(t_k)\| \leq \frac{\epsilon_1}{\epsilon_1 + \frac{\beta^2(t)}{4\alpha}} \frac{\beta(t)}{4\alpha}$$

ensures  $\|e(t)\| \leq \epsilon$ , for  $\epsilon_1 = \frac{c_2 c_4}{c_1 c_3} \epsilon$ .

The proof of this proposition can be found in [23]

As a comparison, in [12], the sufficient condition equivalent to Proposition 4.6 was  $\|x_e(t) - x_e(t_k)\| \leq \frac{\epsilon_1}{\kappa(t)}$ . If

$$\epsilon_1 < \frac{\beta(\kappa - \beta)}{4\alpha}$$

the sufficiency condition in Proposition 4.6 is relaxed than its counterpart in [12].

The following lemma ensures that the proposed event-triggering mechanism excludes Zeno behavior.

**Lemma 4.7:** The inter-trigger time  $\tau_k = t_k - t_{k-1}$  is bounded from below, i.e.  $\inf_k \tau_k \geq \alpha > 0$ . This ensures that within a finite interval  $[t_0, T)$  there will be a finite number of triggerings.

The lemma can be proved following the approach of [12, Theorem 3.1]; we omit it due to space limitation.

*Theorem 4.8 (Main Result):* If there exists  $\epsilon > 0$  and controllers  $\gamma_i(\cdot)$  such that the closed-loop trajectory  $x[t_0] \models \varphi^\epsilon$ , then the event triggered trajectory  $x_e[t_0] \models \varphi$  where events are generated whenever  $g(t) \leq 0$ .

The proof follows directly from Proposition (3.4) where  $y[t_0] = x_e[t_0]$  and we have ensured  $\sup_t \rho(x(t), x_e(t)) = \sup_t \|e(t)\| \leq \epsilon$ .

## V. EXAMPLES AND SIMULATIONS

### A. Example 2

In this example we consider a robotic motion planning task with temporal sequencing and obstacle avoidance. The robot dynamics considered here is a unicycle model as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w \quad (17)$$

where  $x, y \in \mathbb{R}^2$  is the physical position and  $\theta \in [0, 360^\circ)$  is the heading angle.  $v$  is velocity control, and  $w$  is the steering control input. The task is given as follows:

$$\varphi \models \diamond \pi_2 \wedge (\neg \pi_2 \cup \pi_1) \wedge \square \neg \pi_3 \quad (18)$$

where  $\pi_1, \pi_2$  and  $\pi_3$  corresponds to three circular regions as shown (denoted by  $R_1, R_2$  and  $R_3$ ) in Figure 2. The RTL formula defines the task of avoiding  $R_2$  until reaching  $R_1$  and eventually reaching  $R_2$ , and during the whole time the trajectory should avoid  $R_3$ . We adopt a potential function based approach [24] to generate the control laws for navigating the robot. As presented in previous sections, we expand and contract the appropriate regions while synthesizing the control. The region  $R_3$  has been expanded by  $\epsilon = 0.25$  (the dashed boundary around  $R_3$  shows the expansion in Figure 2) while  $R_1$  was contracted and  $R_2$  has been both expanded and contracted (since both  $\pi_2$  and  $\neg \pi_2$  are present), however, we do not explicitly show them in Figure 2. In figure 3, we show the triggering instances for this problem.

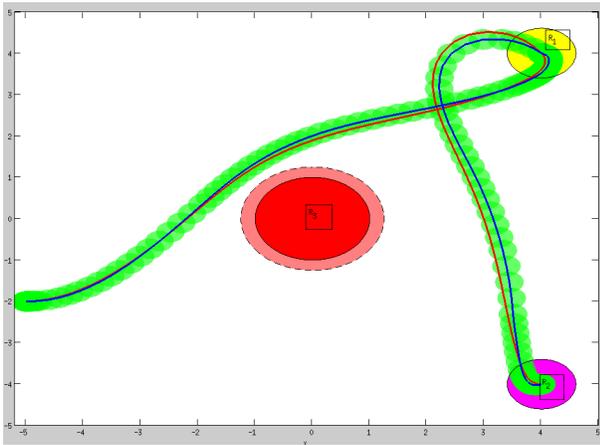


Fig. 2. The closed loop trajectory is plotted using blue line and the event triggered trajectory with red line. The green tube around the nominal trajectory has radius 0.25. The initial position and orientation  $([x, y, \theta])$  of the robot is  $(-5, -2, 0)$

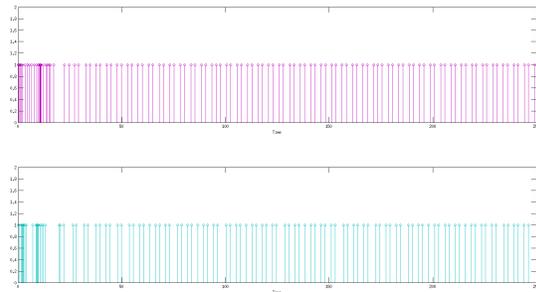


Fig. 3. The upper graph shows the triggering instances for the trajectory from initial position to  $R_1$ . The lower graph shows the same for the other segment of the trajectory.

## VI. CONCLUSION

In this work, we have proposed a framework for integrating the event-triggered controller synthesis and logic based controller synthesis. Our solution is based on composition of two independent controller synthesis framework. It is noteworthy that not any pair of logic-based-controller and event-trigger-controller has this unique composability property. We have derived an explicit event triggering mechanism to bound the trajectory within an  $\epsilon$ -tube. With the notion of robust logic constraints, the resulting trajectory finally satisfies the logical constraint. Simulation results show the significant reduction in communicating the state value for updating the controller. This reduces the communication and computation costs.

## REFERENCES

- [1] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [2] M. Antoniotti and B. Mishra, "Discrete event models+ temporal logic= supervisory controller: Automatic synthesis of locomotion controllers," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 1441–1446, IEEE, 1995.

- [3] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.
- [4] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.
- [5] A. Ulusoy, S. L. Smith, X. C. Ding, and C. Belta, "Robust multi-robot optimal path planning with temporal logic constraints," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4693–4698, IEEE, 2012.
- [6] G. J. Holzmann, "The model checker spin," *IEEE Transactions on software engineering*, vol. 23, no. 5, pp. 279–295, 1997.
- [7] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella, "Nusmv 2: An opensource tool for symbolic model checking," in *International Conference on Computer Aided Verification*, pp. 359–364, Springer, 2002.
- [8] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 3270–3285, IEEE, 2012.
- [9] M. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski, "On self-triggered full-information H-infinity controllers," in *Hybrid Systems: computation and control*, pp. 371–384, Springer, 2007.
- [10] W. Bian and M. French, "General fast sampling theorems for nonlinear systems," *Systems & control letters*, vol. 54, no. 11, pp. 1037–1050, 2005.
- [11] D. Maity and J. S. Baras, "Event based control of stochastic linear systems," in *Event-based Control, Communication, and Signal Processing (ECCSCP), 2015 International Conference on*, pp. 1–8, IEEE, 2015.
- [12] D. Maity and J. S. Baras, "Event based control for control affine nonlinear systems: A lyapunov function based approach," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 3767–3772, IEEE, 2015.
- [13] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009.
- [14] M. Reynolds, "Continuous temporal models," in *Australian Joint Conference on Artificial Intelligence*, pp. 414–425, Springer, 2001.
- [15] A. D. Ames, A. Abate, and S. Sastry, "Sufficient conditions for the existence of Zeno behavior," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 696–701, IEEE, 2005.
- [16] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 61–70, 2007.
- [17] D. C. Conner, A. A. Rizzi, and H. Choset, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies," *Robotics Institute*, p. 124, 2006.
- [18] M. Reynolds, "The complexity of temporal logic over the reals," *Annals of Pure and Applied Logic*, vol. 161, no. 8, pp. 1063–1096, 2010.
- [19] E. M. Clarke, O. Grumberg, and D. Peled, *Model checking*. MIT press, 1999.
- [20] D. Maity and J. S. Baras, "Motion planning in dynamic environments with bounded time temporal logic specifications," in *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pp. 940–946, IEEE, 2015.
- [21] S. R. Lindemann and S. M. LaValle, "Smoothly blending vector fields for global robot navigation," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pp. 3553–3559, IEEE, 2005.
- [22] H. K. Khalil and J. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall New Jersey, 1996.
- [23] D. Maity and J. S. Baras, "Event-triggered controller synthesis for dynamical systems with temporal logic constraints," *ArXiv e-prints*, 2018.
- [24] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on robotics and automation*, vol. 8, no. 5, pp. 501–518, 1992.