2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Self-triggered Control for Safety Critical Systems Using Control Barrier Functions

Guang Yang[1], Calin Belta[2] and Roberto Tron[2]

Abstract— **We propose a real-time control strategy that combines self-triggered control with Control Lyapunov Functions (CLF) and Control Barrier Functions (CBF). Similar to related works proposing CLF-CBF-based controllers, the computation of the controller is achieved by solving a Quadratic Program (QP). However, we propose a Zeroth-Order Hold (ZOH) implementation of the controller that overcomes the main limitations of traditional approaches based on periodic controllers, i.e., unnecessary controller updates and potential violations of the safety constraints. Central to our approach is the novel notion of safe period, which enforces a strong safety guarantee for implementing ZOH control. In addition, we prove that the system does not exhibit a Zeno behavior as it approaches the desired equilibrium.**

I. Introduction

Real-time control is central to many cyber-physical systems, such as autonomous cars, building automation systems, and robots. The design of a real-time controller requires the consideration of several factors, including computational resource constraints, actuator limitations, stability, and safety. An effective way to address the last two objectives is to use Control Lyapunov Functions (CLF) [1] for stability and Control Barrier Functions (CBF) [2] for safety. This formalism was first used in adaptive cruise control [3]. It was also adopted in other safety-critical applications, such as lane keeping in autonomous driving [4], quadrotor control [5], and control of bipedal robot walking [6]. The recently introduced notion of Exponential Control Barrier Function (ECBF) [7] greatly reduced the complexity of designing CBFs for systems with higher relative degree, as compared to [6], [8]. These works compute the desired control using simple optimization problems (typically Quadratic Programs), where the stability and safety requirements are encoded as linear constraints, even for non-linear systems. This formalism, however, is based on a continuous time formulation, which is in contradiction with the reality that these controllers are implemented on digital platforms, where the updates to the control law can happen only at discrete times. In this paper, we address the problem of implementing a continuous CLF-CBF controller on a digital platform with discrete time updates, while preserving stability and safety properties.

Traditionally, digital controllers are implemented using discretized periodic control inputs. A popular discretization method is the Zeroth-Order Hold (ZOH). This approach has two potential major drawbacks when naively combined with the CLF-CBF formalism for safety-critical applications. First, given a fixed update period, there is no guarantee that the safety constraints will hold. Since the plant is sampled at a fixed frequency, the system could violate the safety constraints in between two sampled time instances. Second, there are unnecessary computations and control updates due to fixed-time sampling. This is cumbersome for a system with constrained computational resources and actuator life.

In this paper, we propose to use self-triggered control [9] to address these issues. Self-triggered control was introduced in [10], and related works include [11], [12],[13] and [9]. The core of all self-triggered controllers consists of two parts. First, a designed feedback controller computes the control input at a given time instance. Second, it proactively determines the next controller update time instance based on current state and control objective.

In this work, we propose a novel self-triggered controller that pre-computes the next update time instance given the current state, control objective, and safety requirements. While the controller is applied in a ZOH manner, we ensure that the constraints are not violated between updates.

II. Preliminaries

A. Notation

We use $\mathbb{Z}$ and $\mathbb{R}^n$ to denote the set of integers and the set of real numbers in n dimensions, respectively. The Lie derivative of a smooth function $h(x(t))$ along dynamics $\dot{x}(t) = f(x(t))$ is denoted as $\mathcal{L}_f h(x) := \frac{\partial h(x(t))}{\partial x(t)} f(x(t))$. We use $\mathcal{L}_f^{r_b} h(x)$ to denote a Lie derivative of higher order r_b, where $r_b \geq 0$. A function $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ is called *Lipschitz continuous* on $\mathbb{R}^n$ if there exists a positive real constant $L \in \mathbb{R}^+$, such that $\|f(y) - f(x)\| \leq L\|y - x\|, \forall x, y \in \mathbb{R}^n$. Given a smooth function $h : \mathbb{R}^n \mapsto \mathbb{R}$, we denote h^{r_b} as its r_b-th derivative with respect to time t. A continuous function $\alpha : [-b, a) \mapsto [-\infty, \infty)$, for some $a, b > 0$, belongs to the extended class $\mathcal{K}$ if α is strictly increasing and $\alpha(0) = 0$.

B. Safety Constraints and Control Barrier Functions

Consider a continuous time dynamical control system

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $f(x)$, $g(x)$ are locally Lipschitz continuous. Let $x_0 := x(t_0) \in \mathbb{R}^n$ denote the initial state. For any initial condition x_0, there exists a maximum time interval $\mathrm{I}(x_0) = [t_0, t_{max})$ such that $x(t), \forall t \in \mathrm{I}(x_0)$ is a unique solution. Next, we define a set of safety constraints.

This material is based upon work partially supported by the National Science Foundation under Grant NSF IIS-1723995 and NSF CMMI-1728277.

[1]Guang Yang is with Division of Systems Engineering at Boston University, Boston, MA 02215 USA. Email: gyang101@bu.edu

[2]Roberto Tron,[2]Calin Belta are with the Department of Mechanical Engineering at Boston University, Boston, MA 02215 USA. Email: tron@bu.edu, cbelta@bu.edu

Given a continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$, we define a closed safety set C:

$$\begin{aligned} C &= \{x \in \mathbb{R}^n | h(x) \geq 0\}. \\ \partial C &= \{x \in \mathbb{R}^n | h(x) = 0\}, \\ Int(C) &= \{x \in \mathbb{R}^n | h(x) > 0\}. \end{aligned} \tag{2}$$

The set C is called *forward invariant* for system (1) if $x_0 \in C$ implies $x(t) \in C, \forall t \in \mathrm{I}(x_0)$.

Given a continuously differentiable $h : \mathbb{R}^n \mapsto \mathbb{R}$, and dynamics (1), the relative degree $r_b \geq 0$ is defined as the smallest natural number such that $\pounds_g \pounds_f^{r_b - 1} h(x) u \neq 0$. The time derivative of h are related to the Lie derivatives by:

$$h^{r_b}(x) = \pounds_f^{r_b} h(x) + \pounds_g \pounds_f^{r_b - 1} h(x) u. \tag{3}$$

To ensure forward invariance for systems with higher relative degrees, the authors of [7] introduced the notion of Exponential Control Barrier Function (ECBF). Before formally reviewing its definition, a transverse variable is defined as $\xi_b(x) = \left[h(x), \dot{h}(x), \cdot, \cdot, h^{r_b}(x)\right]^T$ together with a virtual control $\mu = (\pounds_g \pounds_f^{r_b - 1} h(x))^{-1}(\mu - \pounds_f^{r_b} h(x))$. The input-output linearized system corresponding to (1) is

$$\begin{aligned} \dot{\xi}_b(x) &= A_b \xi_b(x) + B_b \mu, \\ y &= C_b \xi_b(x) = h(x), \end{aligned}$$

with

$$A_b = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B_b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{4}$$

$$C_b = \begin{bmatrix} 1 \; 0 \; \cdots \; 0 \end{bmatrix}. \tag{5}$$

Definition 1: Consider the dynamical system in (1) and the closed set C in (2). Given a continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$ with relative degree $r_b = 1$, if there exits a locally Lipschitz extended class $\mathcal{K}$ function α and a set C, such that

$$\inf_{u \in U} [\pounds_f h(x) + \pounds_g h(x) u + \alpha(h(x))] \geq 0, \forall x \in Int(C),$$

then $h(x)$ is a Zeroing Control Barrier Function (ZCBF) [14] and it implies *forward invariance* of system (1).

Definition 2: Consider a dynamical system (1), the safety set C defined in (2) and $h(x)$ with relative degree $r_b \geq 1$. Then $h(x)$ is an Exponential Control Barrier Function (ECBF) [7] if there exists $K_b \in \mathbb{R}^{1 \times r_b}$, such that

$$\inf_{u \in U} [\pounds_f^{r_b} h(x) + \pounds_g \pounds_f^{r_b - 1} h(x) u + K_b \xi_b(x)] \geq 0, \forall x \in Int(C), \tag{6}$$

where the row vector K_b must be selected such that the matrix $A_b - B_b K_b$ has eigenvalues with negative real parts.

Remark 1: As pointed out in [7], the ZCBF is a special case of ECBF with relative degree $r_b = 1$.

C. Stabilization with Control Lyapunov Functions

Definition 3: Given the system (1), a continuously differentiable function $V : \mathbb{R}^n \mapsto \mathbb{R}$ is an Exponentially-Stabilizing Control Lyapunov Function (ES-CLF)[15] if there exist positive constants $c_1, c_2, \epsilon \geq 0$, such that

$$\begin{aligned} & c_1 \|x\|^2 \leq V(x) \leq c_2 \|x\|^2, \\ & \inf_{u \in U} [\pounds_f V(x) + \pounds_g V(x) u + \epsilon V(x)] \leq 0, \quad \forall x \in \mathbb{R}^n. \end{aligned} \tag{7}$$

The existence of a ES-CLF implies that there exists a set of controllers

$$K_{ES-CLF} = \{u \in U : \pounds_f V(x) + \pounds_g V(x) u + \epsilon V(x)] \leq 0\},$$

such that the system is exponentially stabilized [15], i.e.

$$x(t) \leq \sqrt{\frac{c_2}{c_1}} e^{-\frac{\epsilon}{2} t} \|x_0\|, \quad \forall t \geq 0 \tag{8}$$

D. Zeroth-Order Hold

The Zeroth-Order Hold (ZOH) control mechanism holds the control signal at t_k over a period of time, i.e. $u(s) = u(t_k), \forall s \in [t_k, t_{k+1})$. The sequence of control update time instants $\{t_k\}_{k \in \mathbb{N}}$ is strictly increasing.

III. Problem formulation

Let the continuous dynamical system defined in (1) with an initial state $x_0 \in Int(C)$. We want to stabilize the system to a desired state $x_d \in \mathbb{R}^n$ under discretized control input while guaranteeing *forward invariance* of the safety set defined in (2). We propose a self-triggered controller that uses a Quadratic Program (QP) to compute the control signal, and that actively computes the next update instance given the safety constraints and control objective. In particular, we introduce the notions of a safe periods for the safety constraints (τ_{CBF}) and for the stability constraints (τ_{CLF}). These safe periods are computed by means of a lower bound on the ECBF constraints, upper bounds on the CLF, and bounds on the trajectories of the system (i.e., we do not require an explicit integration of the dynamics (1)).

IV. Self-triggered Control using CBF

In this section, we define the CLF-CBF QP for our controller. Next, the notion of safe periods for the CBF and CLF constraints is introduced. Lastly, we present the complete controller update strategy.

A. CBF-CLF Quadratic Program formulation

Given (1), the CBF-CLF QP is defined as

$$\begin{aligned} \min_{u \in \mathrm{U}} \quad & u^T u \\ \text{s.t.} \quad & \pounds_f^{r_b} h(x) + \pounds_g \pounds_f^{r_b - 1} h(x) u + K_b \xi_b \geq 0, \\ & \pounds_f V(x) + \pounds_g V(x) u + \epsilon V(x) \leq 0, \\ & x(t_k) \in Int(C), \end{aligned} \tag{9}$$

The control input is constrained to be in a convex set U, which can be used to model practical actuation limits (e.g., for $u \in \mathbb{R}$, we might have lower and upper bounds u_l and u_u, respectively).

At every update instance t_k, we solve (9) to compute the optimal control input u_k. This control is applied in a ZOH manner until the next update instance t_{k+1}. At a high level, the strategy used by our self-triggered controller is to evaluate whether, with u_k applied in a ZOH manner, the ECBF constraint in (9) will still hold in the interval $t_{k+1} \geq t \geq t_k$, and whether the CLF will decrease at the end of the same period.

B. Distance bound on a system trajectory

For the computation of the safe periods for the ECBF constraints, we rely on bounds for the inequalities in (9). More specifically, we propose to find a bound on the system trajectory that exclusively depends on general properties of the system dynamics. Since we evaluate the trajectory bound at every t_k, we denote $r_{t_k}(t) = r(t+t_k), \forall t \geq t_k$. Our upper bound of r_{t_k} is denoted as $\overline{r}_{t_k}$.

Proposition 1: Given the dynamical system defined in (1), starting at $x(t_k)$, the distance between the trajectory $x(t+t_k)$ and $x(t_k)$ is bounded by $\overline{r}_{t_k}(t) = \frac{1}{L}\|f(x(t_k)) + g(x(t_k))u_k\|(e^{L(t-t_k)} - 1), \forall t \geq t_k$.

Proof: Let $r_{t_k}(t) = \|x(t_k + t) - x(t_k)\|$. Its derivative $\dot{r}(t)$ can be calculated as $\dot{r}(x(t + t_k)) = \frac{(x(t+t_k)-x(t_k))^T}{\|x(t+t_k)-x(t_k)\|} f(x(t + t_k), u)$. Since $\frac{(x(t+t_k)-x(t_k))}{\|x(t+t_k)-x(t_k)\|}$ is a unit vector, we have

$$\begin{aligned} \dot{r}_{t_k} &\leq \|f(x(t + t_k), u)\| \\ &\leq \|f(x(t + t_k), u) - f(x(t_k), u)\| + \|f(x(t_k), u)\|. \end{aligned} \tag{10}$$

Because of the assumption on the Lipschitz continuity of the system dynamics, the following condition holds $\|f(x(t + t_k), u) - f(x(t_k), u)\| \leq L\|x(t + t_k) - x(t_k)\|$, with L as the Lipschitz constant for f. By plugging the inequality into (10), we get $\dot{\overline{r}}_{t_k}(t) \leq L\overline{r}(t+t_k)+\|f(x(t_k), u)\|$. In this case, $\|f(x(t_k), u)\| = \|f(x(t_k)) + g(x(t_k))u_k\|$. The solution is

$$\overline{r}_{t_k}(t) = r_0 e^{L(t-t_k)} - \frac{1}{L}\|f(x(t_k)) + g(x(t_k))u_k\|. \tag{11}$$

The constant r_0 is determined by the condition $\overline{r}_{t_k}(0) = r_{t_k}(0)$, that is $r_0 = \|f(x(t_k)) + g(x(t_k))u_k\|/L$. We then have $r_{t_k} < \overline{r}(t_k)$, based on the comparison theorem. ■

Once we have $\overline{r}_{t_k}(t)$, we can define a ball that bounds the trajectory under system dynamics (1) as $B_{\overline{r}_{t_k}} = \{x \in \mathbb{R}^n : \|x(t) - x(t_k)\| \leq \overline{r}_{t_k}\}$.

C. CBF Safe Period

Definition 4 (Safe Period): For the dynamical system in (1), starting at $x(t_k) \in Int(C)$, if there exists a τ_{CBF} such that $x(t_k + \tau_{\text{CBF}}) \in Int(C)$ for all t in the safe time window $[t_k, t_k + \tau_{\text{CBF}}]$ under a constant control input u_k, then τ_{CBF} is a safe period for this system and control at t_k.

Based on (6), we define the ECBF constraint as

$$\zeta_{\text{ECBF}}(x(t)) = \pounds_f^{r_b} h(x) + \pounds_g \pounds_f^{r_b - 1} h(x)u + K_b \xi_b(x) \tag{12}$$

for $x \in Int(C)$. Based on (6), the system is *forward invariant* if and only if $\zeta_{\text{ECBF}}(x(t)) \geq 0$. We can determine the safe period τ_{CBF} by using $\overline{r}_{t_k}(t)$ to obtain lower bound $\underline{\zeta}_{\text{ECBF}}(x(t))$, so that we do not need to rely on the closed-form solution of $x(t)$; in other words, we will rely on the implication

$$\underline{\zeta}_{\text{ECBF}}(t) \geq 0 \implies \zeta_{\text{ECBF}}(x(t)) \geq 0, \forall t_{k+1} \geq t \geq t_k.$$

At an update instance t_k, we define the initial condition $\underline{\zeta}_{\text{ECBF}}(t_k) = \zeta_{\text{ECBF}}(x(t_k))$. To simplify the notation for the remainder of the section, we define $\zeta(x(t)) := \zeta_{\text{ECBF}}(x(t))$ (with a similar definition for $\underline{\zeta}$). Then $\underline{\zeta}(t)$ can be obtained by another application of the comparison theorem to the following:

$$\underline{\zeta}(t) = \dot{\underline{\zeta}}(t)t + \zeta(t_k), \tag{13}$$

where $\dot{\underline{\zeta}}(t) \leq \dot{\zeta}(t), \forall t_{k+1} \geq t \geq t_k$. To find $\dot{\underline{\zeta}}(t)$, we first denote the derivative of $\zeta(x(t))$ as

$$\dot{\zeta}(x(t)) = \frac{\partial \zeta(x(t))}{\partial x}(f(x(t)) + g(x(t))u)$$

After factoring out each term i.e., $\frac{\partial \zeta(x(t))}{\partial x} f(x(t))$ and $\frac{\partial \zeta(x(t))}{\partial x} g(x(t))u$, we will get an expression in terms of state $x(t)$ and control u. Since the control u is constant under ZOH, we only need to consider the bound on the state. By using proposition (1), we can use $\overline{r}_{t_k}(t)$ to bound the state and use the Lipschitz conditions ζ, f and g to get $\underline{\zeta}(t)$ (details are omitted due to space constraints).

Remark 2: Notice $\underline{\zeta}(t)$ is time dependent because we replace state $x(t)$ with $r(t)$ in our original safety constraint $\zeta(x(t))$. We do not need to calculate a closed-form solution from (1) to evaluate the safety constraint.

With the lower bound $\underline{\zeta}(t)$, we can determine the safe period τ_{CBF}, such that $\underline{\zeta}(t_k + \tau_{\text{CBF}}) = 0$. The problem is equivalent to finding a root for $\underline{\zeta}$. If the closed-form solution of (13) in terms of t is difficult to obtain, we can use algorithms such as the secant method [16] to find its roots. If there are multiple CBF constraints, we denote i-th constraint to be ζ_i. The safe period that satisfies all CBF constraints is

$$\tau_{\text{CBF}} = min(\tau_{\text{CBF},i}), \forall i. \tag{14}$$

D. CLF Update Period

In addition to the safety constraints (12), the CLF constraint is also used to calculate the next update time t_{k+1}. Intuitively, the resulting trajectory might overshoot the equilibrium if we naively apply the following update rule $t_{k+1} := t_k + \tau_{CBF}$.

Because the QP formulation is solved point-wise in time, we cannot guarantee the property of exponential convergence to the desired state on a ZOH implementation. To achieve at least asymptotic stability, we define a CLF update period which guarantees that the Lyapunov function decreases at every step.

Definition 5 (CLF Update Period): For the dynamical system defined in (1), the τ_{CLF} is a CLF update period, if $V(x(t_k + \tau_{\text{CLF}})) - V(x(t_k)) \leq 0$.

For systems that do not have a closed-form solution for their trajectories, we need to find an upper bound $\overline{V}(t)$ such that $\overline{V}(t) \geq V(x(t)), \forall t_{k+1} \geq t \geq t_k$,

$$\overline{V}(t) \leq 0 \implies V(x(t)) \leq 0, \forall t_{k+1} \geq t \geq t_k. \tag{15}$$

The upper bound for $V(x(t))$ can be found using the descent lemma [17]. The following inequality holds $\forall t_{k+1} \geq t \geq t_k$

$$V(t) \leq V(t_k) + (t - t_k)V'(t_k) + (t - t_k)^2 \frac{D}{2} \doteq \overline{V}(t). \tag{16}$$

where $D := \max_{x \in Int(C)} V''$, and we used the notation $V(t) = V(x(t))$; see [17] for a proof.

Remark 3: We can get sharper bounds on D by maximizing the second derivative on $C \cap \{x : V(x) < V(x(t_k))\}$.

Remark 4: We use different bounds for computing τ_{CBF} and those used for τ_{CLF} because otherwise in many common situations we would start with a bound very close to zero near the equilibrium, thus implying a vanishing τ_{CLF}.

Since $\overline{V}(t)$ is a quadratic function in terms of t, there exists a closed-form solution for the roots. The condition that we want to enforce when determining τ_{CLF} is $\overline{V}(t) - V(x(t_k)) \leq 0$, leading to the non-zero root of $\overline{V}(t) = 0$ as

$$\tau_{CLF} = \frac{-2V'(x(t_k))}{D}. \tag{17}$$

Assumption 1: By using the following inequality constraint $V'(x(t)) \leq -\epsilon V(x(t))$ defined in (9), we assume there is a neighborhood of equilibrium such that for the optimal solution from solving the QP, the inequality above becomes the equality $V'(x(t)) = -\epsilon V(x(t))$. We expect that this assumption is valid given the nature of the cost and constraints of the QP, i.e., satisfying the CLF constraint while minimizing control effort. As the system approaches to equilibrium, the Lyapunov Function $V(x)$ decreases toward zero, and optimal control input u will also converge to zero so the CLF constraint will be the only active constraint, while all the others become inactive.

Proposition 2: Given the continous time system (1),

$$\lim_{x \to x_d} \tau_{CLF} > 0, \tag{18}$$

that is, as the system converges the sequence of τ_{CLF} is bounded away from zero, thus avoiding Zeno behaviour.

Proof: We need to show the limit (18) becomes a constant strictly greater than zero as the system approaches to the desired state, i.e. Given the Assumption 1, we can get

$$\tau_{CLF} = \frac{2\epsilon V(x(t_k))}{D} = \frac{2\epsilon V(x(t_k))}{\max_{x \in Int(C)} V''(x(t))}. \tag{19}$$

In addition, there exits a closed-form solution for control u with respect (9). Given the equality assumption, we can analytically determine the control input as

$$u^* = \frac{-\epsilon V(x(t)) - \pounds_f V(x(t_k))}{\pounds_g V(x(t_k))}. \tag{20}$$

Since $V''(x(t))$ depends on both state $x(t)$ and control u. By using the closed form of optimal control input (20), the numerator and denominator of (19) have the same order in terms of $V(x(t))$. Therefore, τ_{CLF} becomes a constant as the system approach the desired equilibrium. ■

The self-triggered control algorithm is summarized in Algorithm 1.

Algorithm 1 Self-Triggered Control with CBF Constraints

```
procedure SELFTRIGGERED(x_0, K_b, h(x))
    x(t_k) := x_0,   ∀x_0 ∈ Int(C)
    while x(t_k) ∉ Goal do
        Calculate optimal u_k by solving (9)
        Calculate the safe period τ_CBF from (14)
        Calculate the CLF update period τ_CLF from (17)
        t_{k+1} := t_k + min(τ_CBF, τ_CLF)
        For system (1), hold u_k between [t_k, t_{k+1}]
    end while
end procedure
```

V. Application to a Second Order Integrator

In this section, we concretely apply the previous theory to the case of a simple second order integrator. Let us define the system to be

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \tag{21}$$

Given the dynamic system (21), we define safety sets $h_1(x) = x_1(t) - x_{1,min}$, $h_2(x) = -x_1(t) + x_{1,max}$, $h_3(x) = x_2(t) - x_{2,min}$, $h_4(x) = x_{2,max}$, where $x_{1,min}, x_{1,max}, x_{2,min}, x_{2,max}$ are constants. The goal is to stabilize our system to a desired state $[x_{1,d}, x_{2,d}]^T$, while maintaining forward invariance of the set $\mathcal{C} = x \in \mathbb{R}^2 : h_i(x) \geq 0, i \in \{1, \ldots, 4\}\}$.

A. CBF-CLF formulation

Given (21), and setting $\alpha(h(x)) = kh(x)$, where k is a relaxation constant, we have the following four CBF constraints,

$$\begin{aligned}
\zeta_1 &= u + k_1 x_2 + k_2(x_1 - x_{1,min}), \\
\zeta_2 &= -u + k_1(-x_2) + k_2(-x_1 + x_{1,max}), \\
\zeta_3 &= u + k(x_2 - x_{2,min}), \\
\zeta_4 &= -u + k(-x_2 + x_{2,max}).
\end{aligned}$$

If $\zeta_i \geq 0, i = 1, ..., 4$ holds, then our system is forward invariant. We define the control objective to be $x_{1,d} = 5$ and $x_{2,d} = 0$. The Lypaunov Function candidate for this particular example is

$$V(x) = \begin{bmatrix} x_1 - x_{1,d} \\ x_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} x_1 - x_{1,d} \\ x_2 \end{bmatrix}. \tag{22}$$

Given (7), we define the CLF constraint to be

$$\eta(x) = [2x_2 + (x_1 - x_{1,d})]u + x_2(2(x_1 - x_{1,d}) + x_2) + \epsilon V. \tag{23}$$

The QP formulation for system (21) is

$$\begin{aligned}
\min_{u \in \mathrm{U}} \quad & u^T u \\
\text{s.t.} \quad & \zeta_i \geq 0, i = 1, ..., 4, \ \eta \leq 0 \\
& x(t_k) \in Int(C), \ u_l \leq u \leq u_u.
\end{aligned} \tag{24}$$

B. Computation of the CBF safe period

To obtain the lower bounds for ζ_i, we first calculate the derivatives $\dot{\zeta_i}$: $\dot{\zeta_1} = k_1 x_2 + k_2 u_k, \dot{\zeta_2} = -k_1 x_2 - k_2 u_k, \dot{\zeta_3} = u_k$ and $\dot{\zeta_4} = -u_k$. Given the trajectory bound $r_{t_k}(t)$, we can obtain derivative bounds $\underline{\dot{\zeta_i}}$ for $\dot{\zeta_i}$. The resulting CBF constraint bounds are shown as the following:

$$\begin{aligned}
\underline{\zeta_1} &= (k_1(x_2(t_k) - r_{t_k}(t)) - k_2\|u_k\|)t + \zeta_1(t_k), \\
\underline{\zeta_2} &= (-k_1(x_2(t_k) + r_{t_k}(t)) - k_2\|u_k\|)t + \zeta_2(t_k), \\
\underline{\zeta_3} &= -k\|u_k\|t + \zeta_3(t_k), \\
\underline{\zeta_4} &= -k\|u_k\|t + \zeta_4(t_k).
\end{aligned}$$

Remark 5: Note $\underline{\zeta_i}, i = 1, ..., 4$ do not depend on $x(t), \forall t > t_k$. We can therefore obtain safe period τ_i by directly finding the roots of $\underline{\zeta_i}$, i.e. $\underline{\zeta_i}(t_k + \tau_i) = 0$.

C. Computation of CLF update period

For an update instance t_k, the $\overline{V}(t)$ is obtained from Taylor-expansion at t_k. Given $V(x(t))$ defined in (22), and letting $x_2 := x_2(t_k), x_1 := x_1(t_k)$, its first derivative is

$$V'(x(t_k)) = 2x_2(x_1 - x_{1,d}) + x_2^2 + ((x_1 - x_{1,d}) + 2x_2)u_k$$

Moreover, to find an appropriate value D, we obtain the second derivative as

$$V''(x(t_k)) = 2x_2^2 + 2u_k(x_1 - x_{1,d}) + 3x_2 u_k + 2u_k^2,$$

where control input u_k and desired states $x_{1,d}$ are constants.

Remark 6: Given the candidate Lyapunov Function (22), the following inequality holds $\|x_1(t) - x_{1,d}\| \leq \sqrt{V(x(t))}$, $\|x_2(t)\| \leq \sqrt{V(x(t))}$.

We use Remark 6 to find the maximum value of $V''(x(t))$. With $x_{t_k} := [x_1(t_k), x_2(t_k)]^T$, the D is chosen as

$$\begin{aligned}
D &= \max V''(x(t)) \\
&= 2V(x_{t_k}) + 2|u_k|\sqrt{V(x_{t_k})} + 3|\sqrt{V(x_{t_k})}||u_k| + 2|u_k|^2
\end{aligned} \tag{25}$$

Next, we would like to show that τ_{CLF} is finite as the system approaches the equilibrium, so that the controller does not update infinitely fast as we approach to the desired state. Substituting D in (19), we have

$$\lim_{x_1 \to x_{1,d}, x_2 \to 0} \frac{2\epsilon V(x(t))}{2V(x(t)) + 5\sqrt{V(x(t))}|u^*| + |u^*|^2}; \tag{26}$$

substituting u^* from (20), it can be shown that the numerator and denominator have the same rate of convergence as $x_1 \to x_{1,d}$ and $x_2 \to 0$. Therefore, $\lim \tau_{CLF}$ is a constant when the system approaches the equilibrium.

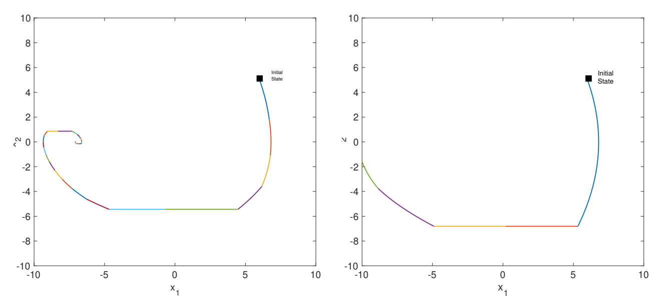
Now we can obtain $\overline{V}(t)$ that is defined in (16), where D is calculated using (25) at each update instance. The CLF update period is

$$\tau_{CLF} = \frac{-2(2x_2(x_1 - x_{1,d}) + x_2^2 + ((x_1 - x_{1,d}) + 2x_2)u_k)}{D}.$$

D. Simulation

Given the double integrator system (21) and an initial state $x_0 = [6, 5]^T$, the objective is to reach $x_{1,d} = -7$, $x_{2,d} = 0$. The safety constraints are $x_{1,min} = -10$, $x_{1,max} = 10$, $x_{2,min} = -10$ and $x_{2,max} = 10$. The two approaches: self-triggered and periodic controls, are both used for comparison. We define self-triggered control updating interval as t_s and periodic control updating interval as t_p. To solve the CBF-CLF QP problem, we set $\epsilon = 0.8$, $L = 1$, $K_b = [105, 20.5]$, $[u_l, u_u] = [-20, 20]$, $t_p = 0.75$, and $t_s = \min(\tau_{\text{CBF}}, \tau_{\text{CLF}})$. At each controller update instance t_k, the CBF-CLF Quadratic Program (24) is solved in Matlab 2018a with Core i5-8259U CPU. The elapsed time for solving each QP problem is around 0.0019s.

The result is illustrated in Figures 2(a) and 2(b). In the case of self-triggered control, it is clear that the controller only updates when the system is about to violate CBF constraints or the system is deviating away from the desired states. Notice that, the update interval for self-triggered controller becomes a lot faster as the system approaches to the unsafe region ($x_1 < x_{1,\text{min}}$) in order to prevent violation on safety constraint. Moreover, the CLF update period converges to 0.3166s and remains as a constant as system approaches to equilibrium, which validate the proposition (2). In the periodic controller case, the position x_1 violates $x_{1,min}$ constraint for $t \in [3, 4]$. (See Figure 1 for a different perspective). It clearly demonstrates the issues with the periodic controller in real-world situation i.e., the controller neither knows the correct sampling rate in-advance, nor has the ability to adjust it real-time. All the safety and convergence properties from CBF and CLF formulation could fail when applying the controller in this manner.

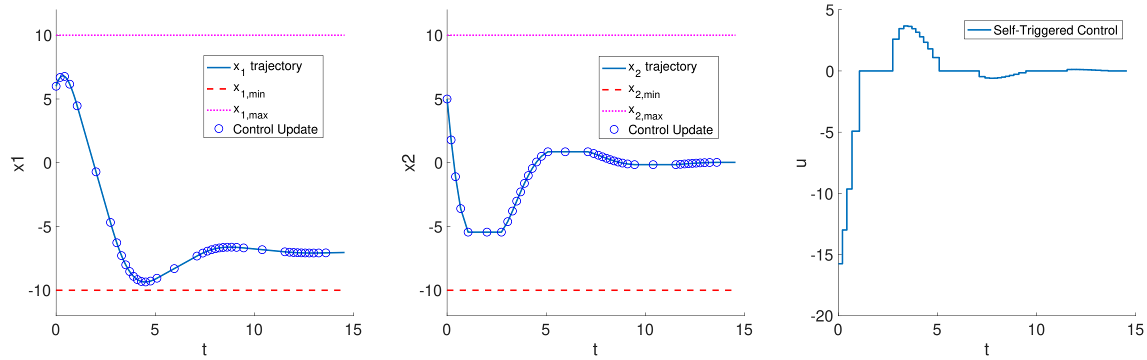


(a) Self-triggered control. (b) Periodic control.

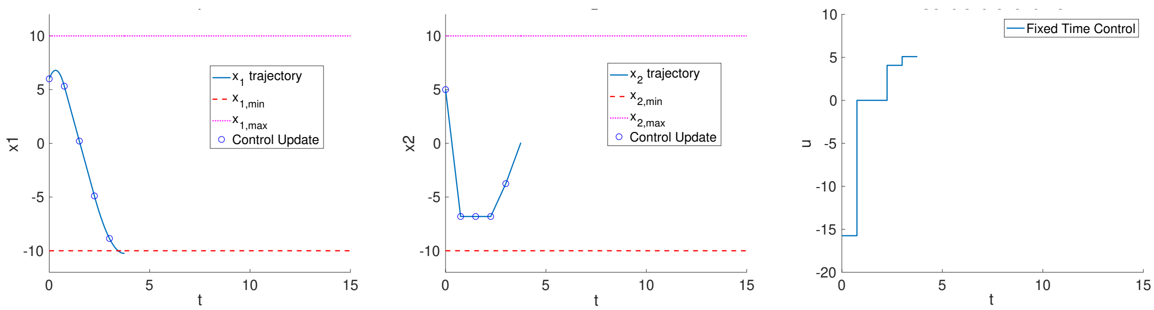
Fig. 1: System trajectories for the two types of controllers.

VI. Conclusion

In this paper, we proposed a self-triggered controller with CBF-CLF based QP formulation that guarantees the safety of our system under ZOH updates while avoiding Zeno behaviour toward convergence. It is a starting point to bridge the gap between theoretical work and real-life implementation based on digital computers. We validated our approach on a double integrator. In the future, we plan to apply our techniques to more complex systems.



(a) Self-triggered control with variable time step.



(b) Periodic control with constant time step (the trajectory is interrupted due to violation of a constraint on x_1).

Fig. 2: Position, velocity, and control inputs for the two types of controller.

References

[1] H. K. Khalil, "Noninear systems," *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.

[2] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[3] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 6271–6278.

[4] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[5] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," *arXiv preprint arXiv:1702.01075*, 2017.

[6] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 4542–4548.

[7] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 322–328.

[8] G. Wu and K. Sreenath, "Safety-critical geometric control for systems on manifolds subject to time-varying constraints,," *IEEE Transactions on Automatic Control (TAC), in review*, 2016.

[9] 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.

[10] M. Velasco, J. Fuertes, and P. Marti, "The self triggered task model for real-time control systems," in *Work-in-Progress Session of the 24th IEEE Real-Time Systems Symposium (RTSS03)*, vol. 384, 2003.

[11] M. Mazo and P. Tabuada, "Input-to-state stability of self-triggered control systems," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 928–933.

[12] M. Mazo Jr, A. Anta, and P. Tabuada, "An iss self-triggered implementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310–1314, 2010.

[13] X. Wang and M. D. Lemmon, "Self-triggered feedback control systems with finite-gain l2 stability," *IEEE transactions on automatic control*, vol. 54, no. 3, p. 452, 2009.

[14] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *arXiv preprint arXiv:1612.01554*, 2016.

[15] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[16] R. P. Brent, *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[17] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.