Hierarchical Event-Triggered Systems: Safe Learning of Quasi-Optimal Deadline Policies

Ong, Pio; Mazo, M.; Ames, Aaron D.

DOI
10.1109/CDC56724.2024.10886395

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the IEEE 63rd Conference on Decision and Control, CDC 2024

2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Hierarchical Event-Triggered Systems: Safe Learning of Quasi-Optimal Deadline Policies

Pio Ong, Manuel Mazo Jr., and Aaron D. Ames

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10886395

***Abstract*— We present a hierarchical architecture to improve the efficiency of event-triggered control (ETC) in reducing resource consumption. This paper considers event-triggered systems generally as an impulsive control system in which the objective is to minimize the number of impulses. Our architecture recognizes that traditional ETC is a greedy strategy towards optimizing average inter-event times and introduces the idea of a deadline policy for the optimization of long-term discounted inter-event times. A lower layer is designed employing event-triggered control to guarantee the satisfaction of control objectives, while a higher layer implements a deadline policy designed with reinforcement learning to improve the discounted inter-event time. We apply this scheme to the control of an orbiting spacecraft, showing superior performance in terms of actuation frequency reduction with respect to a standard (one-layer) ETC while maintaining safety guarantees.**

I. Introduction

A common goal in the development of networked control systems is the reduction of communications bandwidth needed to close the control loop. Similarly, in other applications it is desirable to reduce the amount of actuation changes—to prevent actuators wear or reduce limited actuation resources use. An example of the latter can be found in space applications where fuel availability is limited. The study of how to reduce feedback updates, communication and/or actuation, in control systems has received a lot of attention in the past couple of decades. Much of the work has focused on the minimum data-rates necessary to stabilize and guarantee performance of control loops, see, e.g. [1]–[3]. Alternatively, methods that deviate from the periodic control paradigm have been considered [4], [5]. In particular, Lyapunov based event-triggered control (ETC) and self-triggered control (STC), pioneered in [6], [7], emerged as paradigms attracting much attention from the control community.

The basic idea behind ETC is to monitor (a proxy for) the evolution of certificate functions, e.g., Lyapunov or barrier functions certifying stability/safety, and performance of the control system to determine when feedback updates are required. In STC, rather than continuously monitoring the certificates, the approach is to predict its evolution based on the last received measurements and determine beforehand when controller updates will be required. For a more detailed introduction to these topics see [8]. While the initial focus of this line of research was on stabilization, recent extensions have extended the ETC principles to guarantee safety [9]–[11] employing barrier functions. Two major challenges faced by aperiodic control paradigms, as ETC and STC, are their scheduling and the analysis of feedback usage. Both of these issues are linked to predicting the possible inter-sampling patterns exhibited by such systems [12]–[14].

In [15] the observation was made that ETC, as well as most STC strategies, are greedy optimizers towards the goal of minimizing average feedback resources' usage. Analytically predicting and optimizing aperiodic sampling patters for control is an extremely challenging problem. The authors of [15] circumvent the problem by proposing a STC strategy for LTI systems, based on symbolic abstractions and formal synthesis, resulting in aperiodic implementations that optimize the average inter-event times. Similarly, following a more heuristic approach [11] aimed also at optimizing inter-sample times in the context of safe satellite control.

Hierarchical, also called layered, approaches are a common approach in control engineering to address complex problems [16]. The maxim applied is that of divide-and-conquer, splitting a complex task into smaller/easier to address tasks at different levels of abstraction. This approach appears in biological systems [17], and has been extensively applied in robotics [18], [19], often also taking advantage of the layering to produce multi-rate implementations [20]. A remaining problem in ETC is how to optimize long-term metrics of resource usage, usually as a function of inter-event times, while retaining performance and safety guarantees, particularly when the dynamics are not linear.

In this paper, we draw inspiration from hierarchical structures, and the preliminary work in [21], to propose a layered approach to ETC for general nonlinear systems that optimizes long-term metrics of resources' use while providing strong safety/performance guarantees. The proposed architecture relies on a lower layer providing hard guarantees via well established ETC designs, combined with a supervisory control layer on top that optimize the discounted inter-event times. In particular, we build on Q-learning [22] and propose a faster algorithm (by exploiting the structure of our problem) in the top layer to learn how to prescribe deadlines for the lower layer that enforce feedback updates possibly earlier than the lower layer ETC mechanism prescribes. The proposed architecture thereby enables *safe learning* of strategies, including online optimization during safe operation. We demonstrate the effectiveness of our approach on the application domain of safe control of orbiting spacecraft.

This research is supported in part by TII under project #A6847.

Pio Ong and Aaron D. Ames are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125, USA. {pioong,ames}@caltech.edu

Manuel Mazo Jr. is with the Delft Center for Systems and Control, Faculty of Mechanical Engineering, Delft University of Technology, The Netherlands. m.mazo@tudelft.nl

II. Event-triggered Control Preliminaries

We begin by reviewing key concepts in event-triggered control and establish the motivation for the problem this paper addresses.

Event-triggered control (ETC) is a controller implementation strategy towards resource conservation. To illustrate the main ideas behind ETC, we review its use in the setting of sample-and-hold controller implementations. Consider a sample-and-hold control system[1]:

$$\begin{bmatrix}\dot{\eta}\\ \dot{u}\end{bmatrix} = \begin{bmatrix}F(\eta,u)+d\\ 0\end{bmatrix} \tag{1a}$$

$$\begin{bmatrix}\eta^+\\ u^+\end{bmatrix} = \begin{bmatrix}\eta\\ u\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ \kappa(\eta)-u\end{bmatrix}}_{=\Delta u},\ t\in\{t_i\}_{i\in\mathbb{N}}^{\infty} \tag{1b}$$

with system state $\eta(t)\in\mathbb{R}^q$ and control input variable $u(t)\in\mathbb{R}^m$. The state evolves along the vector field $F:\mathbb{R}^q\times\mathbb{R}^m\to\mathbb{R}^q$, subject to a disturbance $d(t)\in\mathbb{R}^q$. The motivation for sample-and-hold implementations is that the state-feedback $u(t)=\kappa(x(t))$ with a controller $\kappa:\mathbb{R}^q\to\mathbb{R}^m$ cannot be updated in a continuous manner on a digital platform. Rather, common practice involves sampling the values of the controller periodically at different time instants $\{t_i\}_{i\in\mathbb{N}}^{\infty}$ at a high enough frequency to ensure that the sample-and-hold system behaves similarly to the one with continuous feedback. In such implementations, the jump map (1b) occurs at regular intervals.

In many settings, it is undesirable to make frequent changes to the control, e.g., due to communication constraints, or actuation limitations. To this end, ETC arises as an alternative approach for controller sampling. ETC aims at reducing how often the controller is sampled and updated – how often the jump map (1b) is executed – while ensuring desirable behaviors, e.g. stability and safety. The approach consists in monitoring the system behavior and only updating the controller when necessary. In the case of stabilization problems without any disturbance ($d\equiv 0$), for example, ETC sampling schemes aim at guaranteeing a Lyapunov function $W:\mathbb{R}^n\to\mathbb{R}$ decreases along trajectories by determining the update times according to:

$$t_{i+1}=\min\big\{t\geq t_i \mid \mathcal{L}_F W(\eta(t),u(t))\geq 0\big\}, \tag{2}$$

which enforces a controller update whenever the condition $\mathcal{L}_F W(\eta(t),u(t))<0$ with $u(t)=\kappa(\eta(t_i))$ no longer holds. The ETC sampling scheme builds on holding each computed control input for as long as possible. In this sense, the scheme is a greedy approach towards the goal of avoiding overspending resources in control updates. Many works in the literature have demonstrated in practice that this heuristic approach reduces the resource usage in comparison to periodic sampling. Nevertheless, the shift to an aperiodic sampling raises the possibility of Zeno behavior – infinite sampling instances within a finite period of time. As such, the drawback of ETC is that it requires additional analysis to establish a minimum inter-event time (MIET) in order to ensure it can be implemented in practice.

[1]Throughout this paper, we use the notation $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$ for the set of real, nonnegative real, and positive real, respectively. For a vector $x\in\mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. We use $\mathcal{L}_f h=\frac{\partial h}{\partial x}\Big|_x f(x)$ for the Lie derivative of the function $h:\mathbb{R}^n\to\mathbb{R}$ along the vector field $f:\mathbb{R}^n\to\mathbb{R}^n$.

III. Problem description

In this paper, we model event-triggered systems more generally as an impulsive control system with the flow:

$$\dot{x}=f(x)+d, \tag{3a}$$

where $x(t)\in\mathcal{X}\subseteq\mathbb{R}^n$ is the system state. Here we take an impulsive modeling perspective of the event-triggered system. In this setting, the control input does not appear explicitly in the flow[2]. Instead, we consider the effect of the controller to be completely captured in the jump dynamics:

$$x(t_i^+)=g(x(t_i),v(t_i)),\ t\in\{t_i\}_{i\in\mathbb{N}}^{\infty} \tag{3b}$$

where $v(t)\in\mathcal{V}\subseteq\mathbb{R}^p$ represents an impulsive input variable that can instantly influence the state x through the jump map $g:\mathcal{X}\times\mathcal{V}\to\mathcal{X}$. We are especially interested in scenarios where the actuation of v implies the use of some scarce resource, which we would like to minimize. As a result, we wish the time instances $\{t_i\}_{i\in\mathbb{N}}^{\infty}$ to be as *sparse* as possible. To this end, we assume the time sequence is determined iteratively using a triggering condition:

$$t_{i+1}=\min\{\min\big\{t\geq t_i \mid \Xi(x(t),x(t_i))\geq 0\big\}, t_i+\delta_{\max}\}, \tag{4}$$

with an *objective-based triggering condition* $\Xi:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$. In (4), the trigger deadline $\delta_{\max}$ denotes a maximum allowable time between events, sometimes called the system *heartbeat*, used to enforce that a minimum amount of feedback is always present.

Definition 1. *(Objective-based Triggering Condition):* A triggering condition Ξ is objective-based if its value along the trajectory being negative, $\Xi(x(t),x(t_i))<0$, at all time implies that the system control objectives, e.g., stability and safety, are met. •

Note that the triggering condition could be generalized to be time-dependent and potentially depend on all the past states x and past impulsive inputs v. Here we simplify Ξ to the form most commonly found in the literature. In addition, the formulation of the system thus far is a generalization of the sample-and-hold problem discussed earlier where the objective-based triggering condition is based on a Lyapunov condition for stability.

Event-triggered control generally yields an aperiodic time sequence $\{t_i\}_{i\in\mathbb{N}}^{\infty}$. This makes it difficult to assess the sparsity of the time instances along trajectories and their variability across different initial conditions. In order to simplify the problem and make the presentation cleaner, we take the following assumption.

[2]For sample-and-hold systems (1), the continuous control input u is included in the state $x=(\eta,u)$. Note u does not depend on time t but only on time instances $\{t_i\}_{i=0}^{\infty}$ in this formulation because $\dot{u}=0$

Assumption 1. *(Deterministic System):* A state-feedback control policy $\Pi : \mathcal{X} \to \mathcal{V}$ determining the control impulses $v = \Pi(x)$ is given. Furthermore, the disturbance $x \mapsto d(x)$ stems from model uncertainty and is state-dependent. •

This assumption can be relaxed, and stochastic noise can be considered. The assumption allows us to consider inter-event times as being fully determined by the state $x(t_i)$ at the last actuation instance, i.e., there exists a (possibly not analytic) function $\tau_\Xi : \mathcal{X} \to \mathbb{R}_{\geq 0}$ such that $\tau_\Xi(x(t_i)) = t_{i+1} - t_i$. This defines the minimum inter-event time (MIET):

$$\tau_\Xi^* = \inf_{x \in \mathbb{R}^n} \tau(x), \tag{5}$$

which is the smallest time between two consecutive jump instances, from any possible system trajectory using triggering condition Ξ. Most works on ETC provide a MIET in order to rule out Zeno behavior.

Assumption 2. *(Positive Minimum Inter-Event Time):* Consider system (3) using an objective-based triggering condition Ξ to determine iteratively the time instances $\{t_i\}_{i \in \mathbb{N}}^\infty$ as in (4). We assume the MIET $\tau_\Xi^* > 0$ is positive. •

While the existence of a strictly positive MIET is useful to ensure the practicality of implementations, it is an insufficient metric to measure the efficiency of the implementation with respect to use of resources, in the sense that the MIET is a very myopic, worst-case, metric. A better metric for measuring resource consumption should measure long-term distributions of inter-event times. An example of such a metric is the average inter-event time (AIET), which, for a given initial condition x_0, can be computed for the corresponding trajectory as:

$$\bar{\tau}(x_0) = \liminf_{s \to \infty} \frac{1}{s+1} \sum_{i=0}^{s} \tau_\Xi(x(t_i)). \tag{6}$$

Although AIET is a good metric for resource consumption over time, it is more practical to place higher importance to earlier time due to the prediction inaccuracies that may grow over time from system model imperfection, always present in real applications. An alternative metric, that allows us to control the weight given to transient sampling patterns, is the discounted inter-event time (DIET):

$$\bar{\tau}_\gamma(x_0) = \liminf_{s \to \infty} \sum_{i=0}^{s} \gamma^i \tau_\Xi(x(t_i)). \tag{7}$$

where $\gamma \in (0, 1)$ is a discount factor. In this paper we consider DIET as the metric to optimize for the additional reason that it facilitates the implementation of learning approaches, in particular Q-learning, for its optimization. Nonetheless, this can be extended employing available alternatives [23] for the optimization of average costs as the one defined by the AIET. The goal of our paper is to find strategies that maximize the DIET.

Problem 1. *(Discounted Inter-Event Time Problem):* Consider the impulsive system (3). Design an event-triggered control strategy (an objective-based triggering condition Ξ and a deadline $\delta_{\max}$) to maximize the DIET function $\bar{\tau}_\gamma(x_0)$ for all x_0 while maintaining $\Xi(\cdot) < 0$ at all time. •

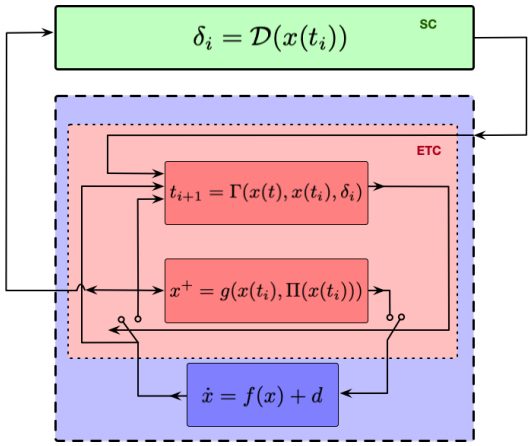

Fig. 1. Schematic description of the layered ETC architecture

Notice that our starting point in the problem is a given objective-based triggering condition. Theoretically, there may exist a triggering condition that can directly achieve our goals, but even if it exists, constructing such condition is extremely difficult. This paper provides a way to modify an existing triggering condition to improve the DIET metric through using a hierarchical framework.

IV. A Hierarchical Architecture

We propose a two-layer hierarchical framework to simplify Problem 1 by decoupling the optimization of the DIET from satisfying the control objectives. The lower layer guarantees safety by leveraging the existing triggering condition to maintain safety of the control objectives. At the same time, we modify the ETC mechanism to include deadlines in order to reduce the greediness of the traditional ETC. The values of these deadlines are determined at the higher layer, and are adjusted each time a jump occurs. The higher layer aims to optimize the DIET by implementing a deadline selection policy for the event-triggered system.

A. Safety layer - satisfying control objectives

By definition, given a Zeno-free objective-based triggering condition Ξ, we can directly ensure the satisfaction of the control objectives by guaranteeing Ξ remains negative across time. Standard ETC approaches extend inter-event times as much as possible for each immediate event, essentially implementing a greedy policy towards reducing the overall number of events. In general, there may exist a triggering condition Ξ' that has a shorter immediate inter-event time $\tau_{\Xi'}(x) < \tau_\Xi(x)$, but whose DIET $\bar{\tau}_\gamma$ is larger because it leads the system towards better (with longer deadlines) triggering states $\{x(t_i)\}_{i \in \mathbb{N}}^\infty$. To ensure that the control objectives are maintained, the new condition Ξ' must be dominated by the original condition Ξ.

Definition 2. *(Dominating Triggering Condition):* A triggering condition Ξ dominates a triggering condition Ξ' if

$\Xi'(x, \bar{x}) < 0 \implies \Xi(x, \bar{x}) < 0$ for all $x, \bar{x} \in \mathbb{R}^n$. •

The task of designing a dominated triggering condition Ξ' that optimizes the DIET can generally be very challenging, so this paper takes a different approach to optimizing the DIET. Inspired by the idea of enforcing early triggering for periodic ETC, put forth by [21], we enrich the ETC scheme to include dynamic deadlines as:

$$\begin{aligned} t_{i+1} &= \min\{\min\left\{t \geq t_i \mid \Xi(x(t), x(t_i)) \geq 0\right\}, t_i + \delta_i\} \\ &:= \Gamma(x(t), x(t_i), \delta_i), \end{aligned} \tag{8}$$

where δ_i are the deadlines available to the upper layer as a control input to affect the triggering condition. The following result shows that we can recover any dominated triggering condition through the use of deadline policies $\mathcal{D} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ to determine these deadline inputs.

Proposition 1. *(Equivalent Deadline Policy): Consider the triggering conditions Ξ and Ξ' such that Ξ dominates Ξ'. Then, there exists a deadline policy $\mathcal{D} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ such that the ETC scheme* (4) *using Ξ' and the scheme* (8) *using Ξ with the feedback $\delta_i = (\mathcal{D}(x(t_i))$ determine the same triggering times $\{t_i\}_{i=0}^{\infty}$.*

Proof. By Definition 2, the condition $\Xi(x(t), x(t_i)) \geq 0$ cannot be met if $\Xi'(x(t), x(t_i)) < 0$. Therefore, the inter-event time function is bounded as $\tau_{\Xi'}(x) < \tau_{\Xi}(x)$ for all $x \in \mathcal{X}$. Let the deadline policy $\mathcal{D}(x) := \tau_{\Xi'}(x)$ be the inter-event times of Ξ'. It follows that (8) evaluates to $t_{i+1} = \tau_{\Xi'}(x)$. Hence the two ETC schemes produce the same sequence $\{t_i\}_{i\in\mathbb{N}}^{\infty}$, concluding the proof. □

Proposition 1 suggests we can use the deadline policy as a proxy for obtaining the optimal triggering strategy. However, we need to rule out the possibility of a deadline policy generating an invalid triggering condition that does not enforces control objectives.

Proposition 2. *(Triggering with Deadlines Enforces Control Objectives): Consider the ETC scheme* (8) *using a triggering condition Ξ and any deadline policy $\mathcal{D} : \mathcal{X} \to [\delta_{\min}, \delta_{\max}]$ for feedback $\delta_i = \mathcal{D}(x(t_i))$ with $\delta_{\min} > 0$. Then, there exists an equivalent ETC scheme in the form of* (4) *using a Ξ' such that Ξ dominates Ξ'. As a consequence, the trigger scheme enforces $\Xi(x(t), x(t_i)) < 0$ at all time.*

Proof. The ETC scheme (8) with the deadline policy $\mathcal{D}$ determines the triggering time as:

$$t_{i+1} = t_i + \min\{\tau_{\Xi}(x(t_i)), \mathcal{D}(x(t_i))\}.$$

Therefore, the ETC scheme (4) using he triggering condition $\Xi'(\cdot) = \min \tau_{\Xi}(x(t_i)), \mathcal{D}(x(t_i))$ is equivalent. Then the trigger design does not exhibit Zeno behavior because it has a MIET $\min\{\tau^*, \delta_{\min}\}$ due to $\mathcal{D}(x) \geq \delta_{\min}$. Consequently, $\Xi(\cdot) < 0$ at all time because the trigger can only occur before $\tau_{\Xi}(x(t_i))$. □

Note that, similarly to Assumption 2, we impose a restriction on the deadlines with a positive lower bound $\delta_{\min}$ in order to restrict ourselves to non-Zeno trigger conditions with a MIET. The implication of Proposition 1 is that we decouple Problem 1 into designing a trigger to enforce control objectives and designing the deadline policy to optimize for the DIET. Regarding the former, the following remark discusses one of the main benefits of our framework.

Remark 1. *(Greedy triggering conditions):* The effectiveness of our framework relies on a greedy nominal triggering condition. If the original triggering condition (4) enforces a large inter-event time, the optimal DIET value will improve due to an increase in available choices of effective deadlines δ. Therefore, our proposed framework incentivizes objective-based triggering conditions that extend each inter-event time as much as possible. To this end, the trigger design task is reduced to finding the least conservative condition that guarantees the control objective. •

B. Optimization layer - improving DIET

With the control objectives guaranteed by the lower layer, the event-triggered system can now be viewed from the higher layer simply as a discrete-time control system with the deadline $\delta \in [\delta_{\min}, \delta_{\max}]$ as an input:

$$x_{i+1} = G(x_i, \delta_i), \tag{9}$$

where $G : \mathcal{X} \times [\delta_{\min}, \delta_{\max}] \to \mathcal{X}$ and $x_{i+1} = x(t_i)$. This system abstracts away the event-times $\{t_i\}_{i\in\mathbb{N}}^{\infty}$ in the lower layer. Nonetheless, the inter-event times can be exposed through a *reward* associated to each discrete time-step:

$$r(x, \delta) := \min\{\tau_{\Xi}(x), \delta\}. \tag{10}$$

From the perspective of the optimization layer, we reformulate the optimization of DIET as an optimal control problem.

Problem 2. *(Optimization layer optimal control problem):* Consider the discrete time system (9)-(10). Find the optimal deadline policy $\mathcal{D}^* : \mathcal{X} \to [\delta_{\min}, \delta_{\max}]$ that maximizes the DIET $\bar{\tau}_{\gamma}(x_0) = \liminf_{s\to\infty} \sum_{i=0}^{s} \gamma^i r(x_i, \mathcal{D}^*(x_i))$ for all initial conditions $x_0 \in \mathcal{X}$. •

The main obstruction to Problem 2 is the map G being nonlinear and difficult to model. Obtaining the map G capturing accurately the corresponding evolution of $x(t_i)$ at event-times is extremely challenging, particularly for general nonlinear systems. Symbolic abstractions have been proposed to this end for both linear [12], [15] and nonlinear [14] systems. However, these abstractions can be computationally demanding to construct and/or conservative, particularly in the case of nonlinear (disturbed) systems. Therefore, instead of constructing such a model and design a (model-based) optimal deadline control policy, we propose to employ a model-free alternative approach.

C. Reinforcement learning for solving Problem 2

Reinforcement learning (approximately) solves optimal control problems by exploring different available actions and observing the obtained rewards to update the control policy. In our context, the reward the agent receives after executing an action is already clearly defined. In this paper, we restrict

the presentation to the simple Q-learning method, which we review in what follows.

We start with a discretization of our state space $\mathcal{X}$ and action space $[\delta_{\min}, \delta_{\max}]$ into the finite sets $\mathcal{S}$ and $\mathcal{A}$, respectively, so that:

$$\mathbf{x}_{i+1} = \mathbf{G}(\mathbf{x}_i, \delta_i) + \mathbf{e}$$

with new state $\mathbf{x} \in \mathcal{S}$, deadline $\delta \in \mathcal{A}$, map $\mathbf{G} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, and the (nondeterministic) error $\mathbf{e}$ due to this discretization. While technically this error is not stochastic, we treat it as if it would be and use this model to approximate the optimal deadline policy. Under this stochastic perspective, we denote by $P_{\mathbf{x}\mathbf{x}'}(\delta)$ the probability of $\mathbf{G}(\mathbf{x}, \delta) + \mathbf{e} = \mathbf{x}'$, and analogously we denote by $R(\mathbf{x}, \delta)$ the random variable associated to the observed reward $r(x, \delta)$, where $\mathbf{x}$ is the discretized state associated to the actual state x.

The Q-learning algorithm associates a *value* function to each state $\mathbf{x}$ under a policy $\tilde{\mathcal{D}} : \mathcal{S} \to \mathcal{A}$:

$$V^{\tilde{\mathcal{D}}} : (\mathbf{x}) := \mathbb{E}[R(\mathbf{x}, \tilde{\mathcal{D}}(\mathbf{x}))] + \gamma \sum_{\mathbf{x}' \in \mathcal{A}} P_{\mathbf{x}\mathbf{x}'}(\tilde{\mathcal{D}}(\mathbf{x})) V^{\tilde{\mathcal{D}}}(\mathbf{x}'),$$

i.e., the sum of an expected instantaneous reward plus the expected discounted value of the next state. The optimal value function V^* represents the optimal (expected) discounted sum of rewards from each state. By the Bellman principle of optimality, V^* satisfies: $V^*(\mathbf{x}) = \max_{\delta \in \mathcal{A}} \{\mathbb{E}[R(\mathbf{x}, \delta)] + \gamma \sum_{\mathbf{x}' \in \mathcal{A}} P_{\mathbf{x}\mathbf{x}'}(\delta) V^*(\mathbf{x}')\}$. The goal in Q-learning is to approximate a function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{>0}$ which when optimized over the actions yields the optimal policy, i.e. $V^{\tilde{\mathcal{D}}^*} = V^*$, with

$$\tilde{\mathcal{D}}^*(\mathbf{x}) := \operatorname*{argmax}_{\delta \in \mathcal{A}} Q^*(\mathbf{x}, \delta). \tag{11}$$

Furthermore, the optimal Q^* satisfies $Q^*(\mathbf{x}, \delta) = \mathbb{E}[R(\mathbf{x}, \delta)] + \gamma \sum_{\mathbf{x}' \in \mathcal{A}} P_{\mathbf{x}\mathbf{x}'}(\delta) V^*(\mathbf{x}')$.

In Q-learning, the optimal Q^* is approximated iteratively through experimentation. Starting from a Q_0 arbitrarily initialized, after each observed transition the approximation of Q^* is adjusted employing the observed reward following:

$$Q_{i+1}(\mathbf{x}, \delta) = (1 - \alpha_i) Q(\mathbf{x}, \delta) + \alpha_i \big(R(\mathbf{x}, \delta) + \gamma V_i(\mathbf{x}')\big) \tag{12}$$

where $\mathbf{x}' = G(\mathbf{x}, \delta) + \mathbf{e}$ (the observed next state) $V_i(\mathbf{x}') = \max_d Q_i(\mathbf{x}', d)$, and $\alpha_i \in (0, 1)$ is the (possibly time varying) learning rate. Under certain conditions of boundedness of rewards and quadratic sum convergence for the rates α_j iterating Q according to (12) leads to the convergence [22] to the optimal Q^*, $\lim_{j \to \infty} Q_j \to Q^*$. Then, the "optimal" deadline policy $\tilde{\mathcal{D}}^*$ can be recovered via the relationship (11).

The Q-learning algorithm yields an approximation to the optimal deadline policy for our original problem, as we employ a discretized model approximation, and in practice, the algorithm must be terminated after a finite number steps.

Remark 2. *(Safe online learning):* One of the strong features of our framework is that the learning can be safely implemented online because Proposition 2 guarantees $\Xi(\cdot) > 0$ regardless of the deadline policy $\mathcal{D}_j$ being used. This opens many doors for using transfer learning approaches that allow the users to train a policy offline with a reduced model and then adapt online the policy to the real dynamics. •

We propose Algorithm 1 for Q-learning specific to our deadline policy, in order to speed up the learning process. The idea is based on the fact that when an action δ is taken at $\mathbf{x}$, we can also observe along the trajectory the rewards $R(\mathbf{x}, \delta')$ and the states $\mathbf{x}_{i+1}^{\delta'} = \mathbf{x}(t_i + \delta')$ for actions $\delta' < \delta$. In addition, if the event is dictated by Ξ rather than the deadline condition, we can also record the same rewards for $\delta' > \delta$. Thus, we may update the function $Q(\mathbf{x}_i, \delta')$ for an array of actions δ' for the state $\mathbf{x}_i$ with each triggering event, even when only one action δ is actually taken.

Algorithm 1 Updating Q for multiple actions per trigger in deadline learning

Input: Q_i, t_i, t_{i+1}, δ, $[t_i, t_{i+1}] \mapsto x(t)$
Output: Q_{i+1}

for $\delta' \in \mathcal{A}$ and $\delta' \leq t_{i+1} - t_i$ **do**
  $R \leftarrow \delta'$
  $V \leftarrow \max_{a \in \mathcal{A}} Q(x(t_i + \delta), a)$
  $Q_{i+1}(\mathbf{x}, \delta') \leftarrow (1 - \alpha) Q_i(\mathbf{x}, \delta') + \alpha(R + \gamma V)$
end for
if $t_{i+1} - t_i < \delta$ **then**
  for $\delta' \in \mathcal{A}$ and $\delta' > t_{i+1} - t_i$ **do**
    $R \leftarrow t_{i+1} - t_i$
    $V \leftarrow \max_{a \in \mathcal{A}} Q(x(t_{i+1}), a)$
    $Q_{i+1}(\mathbf{x}, \delta') \leftarrow (1 - \alpha) Q_i(\mathbf{x}, \delta') + \alpha(R + \gamma V)$
  end for
end if

V. Application to Satellite Orbiting Control

We demonstrate our hierarchical framework with a satellite orbit safety problem.

A. An orbit safety problem

The satellite's dynamics are given by the Newton's gravitational model:

$$\frac{d}{dt} \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} = \begin{bmatrix} \vec{v} \\ -\frac{\mu}{r^3} \vec{r} \end{bmatrix} + \begin{bmatrix} 0 \\ d \end{bmatrix},$$

where the states $\vec{r}(t) \in \mathbb{R}^3$ and $\vec{v}(t) \in \mathbb{R}^3$ are the satellite's position and velocity, respectively and $\mu > 0$ is the *gravitational parameter* of the central body of the orbit. We use the short hand notation $r = \|\vec{r}\|$ for the satellite's orbital radius. Due to the disturbance $d(t) \in \mathbb{R}^3$ in the acceleration, the satellite may drift away from a safe orbit. In response, it can actuate its thruster to change its velocity. The effect of the thruster is typically modelled as an impulse due to the difference in the timescale between the thruster actuation and satellite free-flow orbit (seconds vs. days):

$$\begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix}^+ = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta\vec{v} \end{bmatrix}.$$

where $\Delta\vec{v} \in \mathbb{R}^3$ is the control input.

We consider a satellite orbiting the asteroid 25143 Itokawa. The satellite is subjected to disturbances from

the higher-order gravity field (state-dependent model uncertainty) and the effect of the central body rotating on its axis (time-varying disturbance). Our control objective is to maintain the satellite within safe radius range, $1.6R < r < 2.4R$ where R is the asteroid's mean body radius.

B. Safety layer setup

We use a barrier function $h(r) = (0.4R)^2 - (r - 2R)^2$ to define the set of safe states as:

$$\mathcal{C} = \big\{(\vec{r}, \vec{v}) \in \mathbb{R}^3 \times \mathbb{R}^3 \mid h(r) \geq 0\big\}.$$

The barrier condition [24] for safety is then given by:

$$\underbrace{-2(r - 0.4R)(\vec{r} \cdot \vec{v})/r}_{=\mathcal{L}_f h(\vec{r},\vec{v})} - \left\| \left.\frac{\partial h}{\partial x}\right|_x d \right\| \geq -\alpha(h(r))$$

where we select $\alpha(z) = z/600$. Furthermore, we assume the term with the disturbance is bounded, and we propose the nominal triggering condition as:

$$\Xi(\vec{r}, \vec{v}) = -1200(r - 0.4R)(\vec{r} \cdot \vec{v})/r + h(r) - 0.0005$$

In regard to the impulse feedback control policy $\Delta v = \Pi(\vec{r}, \vec{v})$, we directly calculate the desired $\vec{v}^+$ using orbital mechanics as follows. First, we calculate a "safety factor" $S = \frac{r-2R}{0.4R}$. We use it to compute a target radius of the *peri/apoapsis* of a new orbit that the satellite will be injected into:

$$r_{\text{tgt}} = 2R - (0.2R)S = 3R - 0.5r.$$

Similarly, we select a *true anomaly* for the satellite in the new orbit according to:

$$\nu = \begin{cases} -\pi + \pi S, & S > 0 \\ -(\pi/2)S, & S \leq 0 \end{cases}$$

The reason to pick the target radius and the *true anomaly* with a bias based on S is so that the jump will result in a positive $\mathcal{L}_f h(\vec{r}, \vec{v})$. In turn, the barrier condition is satisfied strictly, ruling out Zeno behavior. Based on r_{tgt} and ν, the *eccentricity* and the *semi-latus rectum* can be computed:

$$e = (r - r_{\text{tgt}})(\text{sgn}(S) r_{\text{tgt}} - r\cos\nu)$$
$$p = r_{\text{tgt}}(1 + \text{sgn}(S)e).$$

Then the desired velocity is given by:

$$\vec{v}^+ = \sqrt{\mu/p}(-\sin\nu \vec{P} + (e + \cos\nu)\vec{Q})$$

where $\vec{P}, \vec{Q}$ are the basis vector of the *perifocal coordinate system* [25, Sec. 2.2.4]. Note that $\Delta v = \vec{v}^+ - \vec{v}$.

C. Optimization layer setup

Now that we have the safety layer enforcing the control safety objective, we can focus on resource conservation via the optimization layer. Here we are concerned with reducing the number of thruster usages because they may interfere with the satellite's mission or generate unnecessary heat. We note that we do not yet consider in this paper the total propellant uses from applying Δv, but reducing the frequency of actuation is nevertheless a good proxy for it. Future work will include propellant usage in the optimization.

Remark 3. *(Reward function and actions):* The reward function considered in this paper only depends on the inter-event times. In the future, we want to incorporate the state x and the impulses v. This would allow us to pose richer optimal control problems, as well as including the impulsive control input v in the optimization. •

For the learning of the deadline policy, we discretize the states into 400 buckets based solely on the radius r and the actions into 10000 different deadlines, ranging from 50 seconds to 100 hours, spaced exponentially. We update the Q function over 180 generations of 100 episodes. Each episode simulates a trajectory from a random initial condition until 20 events take place. Fig. 2 shows the improvement of DIET over generations. We note that because the plot displays the DIET in logarithmic scale, the improvement in the observed maximum DIET over generation is downplayed and the decrease in minimum DIET observed is exaggerated.

D. Results

To test the efficiency of our learned deadline policy, we ran simulations of the closed-loop system from 100 different initial conditions. For each initial condition, we recorded the DIET over 50 events, for both the traditional ETC (greedy) policy and the learned policy. We report the average (across the 100 trajectories) of the DIET to be 2653 and 4422 hours per trigger, respectively, suggesting an improvement of roughly 1.7 times overall.

We plot the learned deadline policy for each of the 400 state buckets in Fig. 3, and we notice that the deadlines differ drastically from the greedy approach (using the maximum deadline) for radii closer to the boundary of the safe set. We ran 100 simulations for a fixed initial radius of 2.3R where the deadline policy drastically change. We report the average of the DIET as 110 and 4644 hours for the greedy policy and the learned policy, respectively. Here, the improvement is staggeringly by the factor of 42. We conclude that the proposed hierarchy can provide great improvement in specific regions. The plots of 10 system trajectories, cf. Fig. 4, shows that system safety is satisfied. The learned policy has learned that the inter-event time for states away from the boundary are naturally longer, so it takes advantage by triggering early than safety requires, in order to dwell at the states away from the boundary.

VI. Conclusion

We have presented a hierarchical framework for optimizing ETC trigger designs for less resources' spending. The framework allows the users to employ any available ETC design, retaining its corresponding guarantees. Furthermore, our proposal decouples the control objectives, guaranteed by the ETC design, from the optimization of long-term inter-event times' metrics via deadline policies. To address the latter, an optimal control problem over abstracted dynamics, not explicitly constructed, is proposed and proposed to be

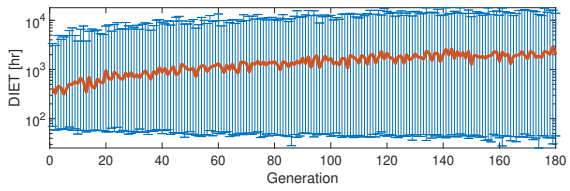

Fig. 2. Improvement of DIET over generations of learning. For each generation, we plot the average (red) of DIET of the 100 trajectories within the generation, and the error bar (blue) encompasses the range of DIET observed. The DIET axis is displayed in logarithmic scale to also reveal the worst-case.

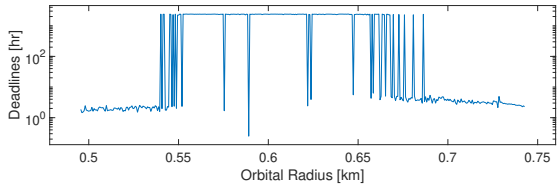

Fig. 3. Learned deadline policy. The plot shows the deadline values for each of the 400 state buckets. For radii close to the boundary of the safe set, the learned deadline is much lower than 100 hours, which is used by the traditional ETC (greedy) policy.

solved via model-free methods, as e.g., RL. In addition, we exploited the structure of deadline generation in a RL implementation with improved learning speed. Future work includes exploration of different learning algorithms, including online transfer learning, richer objective functions in the higher layer optimal control problem, and the incorporation of stochastic noise in the framework.

REFERENCES

[1] D. Liberzon and J. P. Hespanha, "Stabilization of nonlinear systems with limited information feedback," *IEEE Transactions on Automatic Control*, vol. 50, no. 6, pp. 910–915, 2005.

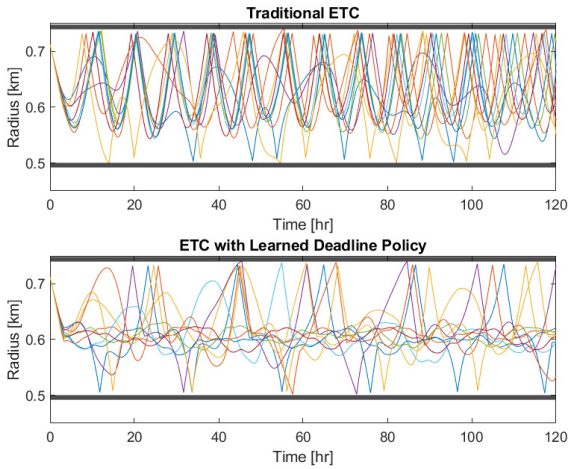

Fig. 4. Ten trajectory comparisons between the greedy deadline policy (top) and learned deadline policy (bottom). In both cases, the underlying ETC strictly enforces safety. However, the learned policy intelligently triggers in the way that trajectories dwell in the region where inter-event times are longer, leading to an overall increase in DIET.

[2] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1056–1068, 2004.
[3] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.
[4] K. J. Astrom and B. M. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *IEEE Conf. on Decision and Control*, vol. 2, pp. 2011–2016, IEEE, 2002.
[5] M. Velasco, J. Fuertes, and P. Marti, "The self triggered task model for real-time control systems," in *24th IEEE Real-Time Systems Symposium*, vol. 384, pp. 67–70, 2003.
[6] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.
[7] 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.
[8] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 3270–3285, Dec. 2012.
[9] P. Ong and J. Cortés, "Performance-barrier-based event-triggered control with applications to network systems," *IEEE Transactions on Automatic Control*, vol. 69, no. 7, 2024. To appear.
[10] A. J. Taylor, P. Ong, J. Cortés, and A. Ames, "Safety-critical event triggered control via input-to-state safe barrier functions," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 749–754, 2021.
[11] P. Ong and A. D. Ames, "Intermittent safety filters for event-triggered safety maneuvers with application to satellite orbit transfers," in *IEEE Conf. on Decision and Control*, (Marina Bay Sands, Singapore), Dec. 2023. To appear.
[12] A. S. Kolarijani and M. Mazo Jr., "Formal traffic characterization of lti event-triggered control systems," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 274–283, 2016.
[13] G. de Albuquerque Gleizer and M. Mazo Jr., "Chaos and order in event-triggered control," *IEEE Transactions on Automatic Control*, 2023.
[14] G. Delimpaltadakis and M. Mazo Jr., "Abstracting the traffic of nonlinear event-triggered control systems," *IEEE Transactions on Automatic Control*, 2022.
[15] G. de A. Gleizer and M. Mazo Jr., "Computing the sampling performance of event-triggered control," in *Hybrid systems: Computation and Control*, pp. 1–7, 2021.
[16] N. Matni, A. D. Ames, and J. C. Doyle, "Towards a theory of control architecture: A quantitative framework for layered multi-rate control," *arXiv preprint arXiv:2401.15185*, 2024.
[17] J. Merel, M. Botvinick, and G. Wayne, "Hierarchical motor control in mammals and machines," *Nature communications*, vol. 10, no. 1, p. 5489, 2019.
[18] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.
[19] J. Kim, R. T. Fawcett, V. R. Kamidi, A. D. Ames, and K. A. Hamed, "Layered control for cooperative locomotion of two quadrupedal robots: Centralized and distributed approaches," *IEEE Transactions on Robotics*, 2023.
[20] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multirate control: From low-level actuation to high-level planning," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6627–6640, 2022.
[21] R. de Ruijter, "Deep reinforcement learning for the synthesis of self-triggered sampling strategies," Master's thesis, Delft University of Technology, 2022.
[22] P. Dayan and C. J. C. H. Watkins, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.
[23] S. Mahadevan, "Average reward reinforcement learning: Foundations, algorithms, and empirical results," *Machine learning*, vol. 22, pp. 159–195, 1996.
[24] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *European Control Conference*, (Naples, Italy), pp. 3420–3431, June 2019.
[25] R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. New York: Dover Publications, 1971.