Event-Triggered Polynomial Control for Trajectory Tracking by Unicycle Robots

Harini V[1], Anusree Rajan[1], Bharadwaj Amrutur[1], and Pavankumar Tallapragada[1]

***Abstract*— This paper proposes an event-triggered polynomial control method for trajectory tracking by unicycle robots. In this method, each control input between two consecutive events is a polynomial and its coefficients are chosen to minimize the error in approximating a continuous-time control signal. We design an event-triggering rule that guarantees uniform ultimate boundedness of the tracking error and non-Zeno behavior of inter-event times. We illustrate our results through a suite of numerical simulations and experiments, which indicate that the number of events generated by the proposed controller is significantly less compared to that by a time-triggered controller or a event-triggered controller based on zero-order hold while guaranteeing similar tracking performance.**

I. INTRODUCTION

A. Motivation

Trajectory tracking for mobile robots is a well-studied problem with many applications, such as industrial automation, military surveillance and multi-robot coordination. An important challenge in these applications may be constrained resources, such as communication, computation, and energy. Event-triggering [1]–[3] is a popular method for control under such resource constraints. The zero-order-hold (ZOH) technique is mostly used in the event-triggered control (ETC) method to design control laws. However, ZOH control may lead to a reduction in the efficiency of usage of communication resources due to under utilization of each communication packet. Whereas, non-ZOH control could improve the usage of each communication packet. With this motivation, in this work, we present an event-triggered polynomial control method for trajectory tracking of unicycle robots.

B. Literature Review

The literature on trajectory tracking with event-triggered communication includes [4], which proposes an event-triggered tracking control method for non-linear systems that guarantees uniform ultimate (UU) boundedness of the tracking error and absence of Zeno behavior of inter-event times (IETs). Similarly, references [5]–[7] propose Lyapunov based event or self-triggered tracking controllers for mobile robots by emulating a continuous time controller and guarantee ultimate boundedness of the tracking error. Reference [8] obtains a linear system model for a Pioneer robot by system identification and designs an adaptive self-triggered tracking controller. Reference [9] proposes an event-triggered optimal tracking control method for nonlinear systems using ideas from reinforcement learning. Whereas, reference [10] presents a self-triggered model predictive control (MPC) strategy for trajectory tracking of unicycle-type robots with input constraints and bounded disturbances. Reference [11] deals with the tracking control of quadrotors with external disturbances and proposes an event-triggered sliding mode control strategy. The recent paper [12] proposes both event-triggered and self-triggered saturated feedback control strategies for trajectory tracking of unicycle mobile robots. All these papers, except [10], are based on the ZOH control technique.

Model-based ETC [13], [14] is one among the limited number of works on non-ZOH ETC. This method allows for generating a non-ZOH control input to the plant by using a model of the plant at the actuator. Other control methods based on non-ZOH control are event/self-triggered MPC [15], [16] and event-triggered dead-beat control (DBC) [17]. In these methods, the controller sends a control sequence to the actuator intermittently and this control sequence is applied to the plant between two comunication instants. References [18], [19] extend this idea based on first-order-hold (FOH), where, the control input is linearly interpolated between sampling points in the prediction horizon.

In our recent papers, we present a novel ETC method referred as event-triggered parameterized control based on a non-ZOH technique. We first showcase this method for stabilization of linear systems [20], and later we apply the same idea to nonlinear systems with external disturbances [21]. A couple of recent papers [22], [23] use a parameterized control law in an MPC setting without event-triggering.

C. Contributions

The major contributions of this work are given below:

- In this work, we propose an event-triggered polynomial control (ETPC) method for trajectory tracking of unicyle robots. The proposed controller guarantees UU boundedness of the tracking error and non-Zeno behavior of IETs.
- We present the results of practical implementation on a ground robot. This is a contribution to the as yet limited literature on practical implementations of event-triggered controllers.
- Through simulations and experiments, we show significant reduction in communication compared to the standard ZOH based ETC.
- Compared to the existing non-ZOH ETC methods, such as model based ETC, event/self-triggered MPC and event-triggered DBC, the proposed method allows for generating a non-ZOH control input using limited

[1] Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bengaluru. {hariniv, anusreerajan, amrutur, pavant}@iisc.ac.in

computational resources at the actuator and transmitting limited information over the communication network.

- The only papers that consider an event-triggered control method similar to the one proposed is [20], [21], where the objective is stabilization. Whereas, in the current paper, the context is trajectory tracking for unicycle robot models. The results in [21], which considers a time invariant nonlinear system, are not directly applicable here for the stability analysis of tracking error as the tracking error dynamics explicitly depends on time t. Moreover, in this paper, we validate our results practically through several experiments.

D. Notation

Let the set of all real numbers, positive integers, and non-negative integers be, respectively, denoted as $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{N}_0$. Let the euclidean norm of y, for any $y \in \mathbb{R}^n$, be denoted as $\|y\|$. Let $g(t^+) := \lim_{s \to t^+} g(s)$ for any right continuous function $g : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\forall t \geq 0$. Let $\langle f, g \rangle := \int_0^t f(s)g(s)\mathrm{d}s$ for any two functions $f, g : [0,t] \to \mathbb{R}$.

II. PROBLEM FORMULATION

This paper proposes a tracking control method that works best in cases where communication is significantly more costly than computation. In this section, we present the system model and the control objective.

A. System Dynamics

Consider the unicycle model of a robot,

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{1}$$

where (x,y) denotes the position of the robot and θ denotes the orientation of the robot, which is the angle between the heading direction of the robot and the $x-$axis. v and ω denote, respectively, the linear velocity and the angular velocity of the robot, which are the control inputs. The position and the orientation of the robot are continuously available to the controller. The robot has to track a given reference trajectory which satisfies the following model,

$$\dot{x}_r = v_r\cos\theta_r, \quad \dot{y}_r = v_r\sin\theta_r, \quad \dot{\theta}_r = \omega_r, \tag{2}$$

where (x_r, y_r) and θ_r denote, respectively, the reference position and the reference orientation. v_r and ω_r denote the inputs to the reference system. We make the following assumption on the reference inputs, just as in [5].

(A1) There exists $M \geq 0$ such that $|v_r(t)|, |\omega_r(t)|, |\dot{v}_r(t)|$ and $|\dot{\omega}_r(t)|$ are upper bounded by M, $\forall t \geq t_0$. Moreover, there exists a $c > 0$ such that $|v_r(t)| \geq c$, $\forall t \geq t_0$. •

The first condition in Assumption **(A1)** is very reasonable because it just says that the reference signals and their derivatives are bounded. It is also a common assumption in the literature [5], [24]. As discussed in [5], there are several reference trajectories that satisfy the second condition in Assumption **(A1)**, see Section V for examples. Also, practically, for low velocities there may be other effects that one may need to consider such as friction. At the same time, we could also use another controller that we could switch to during the time intervals when the magnitude of the reference velocity is below the threshold c. We also assume that the reference trajectory and the reference inputs are available to the controller a priori.

The tracking error, in the robot frame, can be represented as follows,

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} := \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \tag{3}$$

The evolution of tracking error in robot frame is given by

$$\dot{X} = F(X,t) + G(X)u, \tag{4}$$

where $X = \begin{bmatrix} x_e & y_e & \theta_e \end{bmatrix}^\top$, $u = \begin{bmatrix} v & \omega \end{bmatrix}^\top$,

$$F(X,t) = \begin{bmatrix} v_r\cos\theta_e & v_r\sin\theta_e & \omega_r \end{bmatrix}^\top,$$

$$G(X) = \begin{bmatrix} -1 & 0 & 0 \\ y_e & -x_e & -1 \end{bmatrix}^\top.$$

In this paper, we wish to design a controller for trajectory tracking by unicycle robots, where the controller communicates with the actuator over a communication network intermittently. We wish to design a non-ZOH control law while transmitting limited information at each communication instance. So, we consider a polynomial control input whose coefficients are updated at each event. In particular, each control input is a p^{th} degree polynomial between any two consecutive events. Now, let $u_1(t) := v(t)$ and $u_2(t) := \omega(t)$. Then, for $i \in \{1,2\}$ $\forall \tau \in [0, t_{k+1} - t_k)$,

$$u_i(t_k + \tau) = f(\mathbf{a}_i(k), \tau) := \sum_{j=0}^{p} a_{ji}(k)\tau^j. \tag{5}$$

Here $\mathbf{a}(k) := [a_{ji}(k)] \in \mathbb{R}^{(p+1)\times 2}$ denotes the coefficients of the polynomial control input during the interval $[t_k, t_{k+1})$ where $(t_k)_{k\in\mathbb{N}_0}$ denotes the sequence of time instants at which the controller updates $\mathbf{a}(k)$, and communicates the same to the actuator. Here, we denote $\mathbf{a}_i(k)$ as the i^{th} column of $\mathbf{a}(k)$. Note that, the results in this paper are easily extendable to the more generalized parameterized control law proposed in [20].

Note here that, at a communication time instant t_k for any $k \in \mathbb{N}_0$, the controller only has to send $\mathbf{a}(k)$ to the actuator. Once the actuator receives this information, it can easily generate the time-varying control input to the plant.

Figure 1 depicts the block diagram of the ETPC system proposed in this paper. Here, the controller has continuous

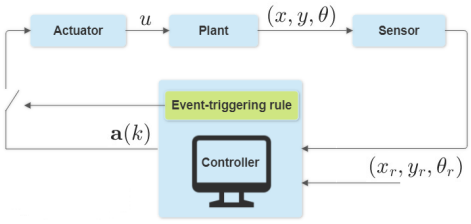


Fig. 1: Configuration of ETPC

access to the plant state (x,y,θ) and the reference trajectory

(x_r, y_r, θ_r). We also assume that the controller has sufficient computational capacity to determine $(t_k)_{k\in\mathbb{N}_0}$ and to recompute the coefficients $\mathbf{a}(k)$ at t_k for any $k \in \mathbb{N}_0$.

B. Objective

The objective of this paper is to determine the coefficients of the polynomial control law and to design an event-triggering rule (ETR) that implicitly determine the communication time instants, so that the tracking error is uniformly ultimately bounded. Our objective also includes ensuring the absence of Zeno behavior of IETs.

III. DESIGN OF THE ETPC SYSTEM

This section deals with the design of a polynomial control law as well as an ETR that achieve the objective of this paper.

A. Design of Polynomial Control Law

We first consider the continuous time feedback control signal, $\hat{u} := \begin{bmatrix}\hat{v} & \hat{\omega}\end{bmatrix}^T$, that was proposed in [24]. In particular,

$$\begin{aligned}\hat{v}(\hat{X},t) &:= v_1(\hat{X},t) + c_1(\hat{x}_e - c_3\hat{\omega}(\hat{X},t)\hat{y}_e),\\ \hat{\omega}(\hat{X},t) &:= \omega_r + \gamma\hat{y}_e v_r \operatorname{sinc}\hat{\theta}_e + c_2\gamma\hat{\theta}_e,\end{aligned} \tag{6}$$

where

$$\begin{aligned}v_1(\hat{X},t) :=& v_r\cos\hat{\theta}_e + c_3\hat{\omega}(\hat{X},t)(\hat{\omega}(\hat{X},t)\hat{x}_e - v_r\sin\hat{\theta}_e)\\ &- c_3 v_2(\hat{X},t)\hat{y}_e,\\ v_2(\hat{X},t) :=& \dot{\omega}_r + \gamma v_r \operatorname{sinc}\hat{\theta}_e(-\hat{\omega}(\hat{X},t)\hat{x}_e + v_r\sin\hat{\theta}_e) +\\ &\gamma\hat{y}_e\dot{v}_r\operatorname{sinc}\hat{\theta}_e + (\gamma\hat{y}_e v_r \operatorname{sinc}'\hat{\theta}_e + c_2\gamma)(\omega_r - \hat{\omega}(\hat{X},t)).\end{aligned} \tag{7}$$

Here $\hat{X} := \begin{bmatrix}\hat{x}_e & \hat{y}_e & \hat{\theta}_e\end{bmatrix}^T$ evolves as,

$$\dot{\hat{X}} = F(\hat{X},t) + G(\hat{X})\hat{u}(\hat{X},t), \quad \forall t \in [t_k, t_{k+1}), \tag{8}$$

where $\hat{X}(t_k) = X(t_k)$ for all $k \in \mathbb{N}_0$. Note also that $\operatorname{sinc}'\hat{\theta}_e$ denotes the derivative of $\operatorname{sinc}\hat{\theta}_e$ with respect to $\hat{\theta}_e$ and $c_1, c_2, c_3, \gamma > 0$ are design parameters. Reference [24] considers the tracking error evolution (4) with $u = \hat{u}(X,t)$ and shows global convergence of tracking error to zero under some conditions on the reference inputs.

Our idea is to find the best polynomial approximation of the control signal (6). At each communication time instant t_k, the controller updates $\mathbf{a}(k)$ by solving the following optimization problems, $\forall i \in \{1,2\}$,

$$\begin{aligned}\mathbf{a}_i(k) \in \underset{a\in\mathbb{R}^{p+1}}{\arg\min} & \int_0^T \left|f(a,\tau) - \hat{u}_i(\hat{X},t_k+\tau)\right|^2 + \delta_i\left|f(a,\tau)\right|^2 \mathrm{d}\tau,\\ \text{s.t.}\quad & f(a,0) = \hat{u}_i(\hat{X}(t_k),t_k),\end{aligned} \tag{9}$$

where $\hat{u}_1 := \hat{v}$ and $\hat{u}_2 := \hat{\omega}$. Here, $T > 0$ is an important parameter which is to be designed. $\delta_1, \delta_2 \geq 0$ are design parameters which are useful for penalizing large magnitudes of the control input signal. Note that, to solve the optimization problem (9), we need the values of $\hat{v}$ and $\hat{\omega}$ over the time horizon $(t_k, t_k + T]$. These values are estimated by numerically simulating the system (8).

Note that, the only constraint in optimization problem (9) fixes the value of a_0. Hence, letting

$$\bar{u}_i(\tau) := \hat{u}_i(\hat{X}, t_k + \tau), \quad \mu_i := \bar{u}_i(0),$$

we can rewrite (9) as the following unconstrained optimization problem, for $i \in \{1,2\}$,

$$\bar{\mathbf{a}}_i(k) \in \underset{a\in\mathbb{R}^p}{\arg\min} J_i(a), \tag{10}$$

where,

$$\begin{aligned}J_i(a) =& \langle\bar{u}_i,\bar{u}_i\rangle + (1+\delta_i)\mu_i^2 T - 2\mu_i\langle\bar{u}_i,1\rangle - 2\sum_{j=1}^{p} a_j\langle\bar{u}_i,\tau^j\rangle\\ &+ (1+\delta_i)\sum_{j=1}^{p}\left[\sum_{l=1}^{p} a_j a_l \frac{T^{j+l+1}}{j+l+1} + 2\mu_i a_j \frac{T^{j+1}}{j+1}\right].\end{aligned}$$

Thus, we have

$$\mathbf{a}_i(k) = \begin{bmatrix}\mu_i & \bar{\mathbf{a}}_i^T(k)\end{bmatrix}^T.$$

Viewed this way, we see that Problem (9) is always feasible. Note that, given the coefficients of the polynomial control input that are obtained by solving (10), the control input that is applied by the actuator is as given in (5).

Remark 1. *(Control signal for $\tau > T$). As discussed in Remark 1 of [21], even if $t_{k+1} - t_k > T$, the control input $u(t_k + \tau)$ is well defined $\forall \tau \in [t_k, t_{k+1})$.* •

Proposition 2. *The finite horizon optimization problem (10) is strictly convex.*

Proof. Proof of this result follows along similar lines as in the proof of Proposition 2 in [21]. For all $i \in \{1,2\}$, the Hessian matrix of $J_i(.)$, denoted as $\mathbf{H}_i$, is as follows,

$$\mathbf{H}_i = 2(1+\delta_i)\begin{bmatrix}\frac{T^3}{3} & \frac{T^4}{4} & \cdots & \frac{T^{p+2}}{p+2}\\ \frac{T^4}{4} & \frac{T^5}{5} & \cdots & \frac{T^{p+3}}{p+3}\\ \cdots & \cdots & \cdots & \cdots\\ \frac{T^{p+2}}{p+2} & \frac{T^{p+3}}{p+3} & \cdots & \frac{T^{2p+1}}{2p+1}\end{bmatrix}, \quad \forall i \in \{1,2\}.$$

Observe that $\mathbf{H}_i$ is $2(1+\delta_i)$ times the Gram matrix for the functions in $\{\tau^j : [0,T] \to \mathbb{R}\}_{j=1}^p$. As it is a set of linearly independent functions, we can say that $\mathbf{H}_i$ is a positive definite matrix $\forall i \in \{1,2\}$. This implies strict convexity of the cost function in (10). As there are no constraints in (10), it is a strictly convex optimization problem. □

Note also that the computational requirement of the proposed controller is similar to that of the existing control methods like the event-triggered MPC or the event-triggered DBC.

B. Design of ETR

We consider the following Lyapunov-like function,

$$V(X,t) = \frac{1}{2}x_1^2 + \frac{1}{2}y_e^2 + \frac{1}{2\gamma}\theta_e^2, \tag{11}$$

where $x_1 := x_e - c_3\hat{\omega}(X,t)y_e$, to design the event-triggering rule. Please note that in x_1, it is indeed $\hat{\omega}(X,t)$ and not $\hat{\omega}(\hat{X},t)$. Letting $e(t) := u(t) - \hat{u}(X,t)$, we see that the

derivative of V along the trajectories of the sampled data system (4)-(5) can be expressed as

$$\begin{aligned}\dot{V} &= \frac{\partial V}{\partial t} + \frac{\partial V}{\partial X}\dot{X} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial X}(F(X,t)+G(X)u) \\ &= -\Sigma(X,t)+\Lambda(X,e,t)\end{aligned} \tag{12}$$

where

$$\Sigma(X,t) := c_1 x_1^2 + c_2\theta_e^2 + c_3\hat{\omega}^2(X,t)y_e^2, \tag{13}$$

$$\Lambda(X,e,t) := \frac{\partial V}{\partial X}G(X)e(t). \tag{14}$$

The derivation of equation (12) is given in the appendix. Now, we define the (ETR) as follows,

$$t_{k+1} := \min\{t > t_k : \dot{V} \geq -\sigma\Sigma(X,t) \text{ and } V(X,t) \geq \varepsilon^2\}, \tag{15}$$

where $\sigma \in (0,1)$, $\varepsilon^2 > 0$ are design parameters and $t_0 := 0$.

In summary, the complete system, $\mathcal{S}$, is the combination of the reference system (2), the unicycle robot model (4), the polynomial control law (5), with coefficients updated by solving (10) at the time instants determined by the ETR (15). That is,

$$\mathcal{S}: \ (2),(4),(5),(10),(15). \tag{16}$$

IV. ANALYSIS OF THE ETPC SYSTEM

In this section, we analyze the performance of the designed ETPC system. We show that for system (16), there exists a UU bound on the tracking error and a uniform positive lower bound on the IETs. We present a couple of lemmas.

Lemma 3. *Consider system (16) and Lyapunov function (11). Let Assumption (A1) hold and $\varepsilon_k^2 := V(X(t_k),t_k)$. Then, for any $c_1,c_2,c_3,\varepsilon^2 > 0$ and $\gamma > 0$ sufficiently large, $V(X(t),t) \leq \varepsilon_k^2, \forall t \in [t_k,t_{k+1})$ and $\forall k \in \mathbb{N}$.*

Proof. Note that, as $e(t_k^+) = 0$ for any $k \in \mathbb{N}_0$, $\dot{V}(X(t_k^+),t_k^+) = -\Sigma(X(t_k^+),t_k^+)$. Further the ETR (15) implies that $\varepsilon_k^2 \geq \varepsilon^2$ for all $k \in \mathbb{N}$ and hence $\dot{V}(X(t_k^+),t_k^+) < 0$. The last inequality follows from Lemma 4 in [5], which states that there exists $\mathcal{V} > 0$ for any $c_1,c_2,c_3,\varepsilon^2 > 0$ and $\gamma > 0$ sufficiently large such that $V(X,t) \geq \varepsilon^2$ implies $\Sigma(X,t) \geq \mathcal{V} > 0$. Now, let us suppose that the statement $V(X(t),t) \leq \varepsilon_k^2, \forall t \in [t_k,t_{k+1})$ and $\forall k \in \mathbb{N}$, is not true. Then, as $V(X,t)$ is a differentiable function of time, there must exist $\bar{t} \in (t_k,t_{k+1})$, for some $k \in \mathbb{N}$, such that $V(X(\bar{t}),\bar{t}) = \varepsilon_k^2$ and $\dot{V}(X(\bar{t}),\bar{t}) > 0$. However, as $\bar{t} < t_{k+1}$, we can say that the triggering condition is not satisfied at $t = \bar{t}$ and hence $\dot{V}(X(\bar{t}),\bar{t}) < -\sigma\Sigma(X(\bar{t}),\bar{t}) < 0$. As it contradicts with the previous statement, we can say that there does not exist such a $\bar{t}$ and hence the result is true. □

Remark 4. *Under Assumption (A1), $V(X,t)$ is a continuously differentiable positive definite radially unbounded function of X.* •

Next, we show that the control u and its time derivative are uniformly bounded between any two consecutive events. This result helps us to prove that the IETs generated by the proposed ETPC method do not exhibit Zeno behavior.

Lemma 5. *Consider system (16) and Lyapunov function (11). Let Assumption (A1) hold and $\varepsilon_k^2 := V(X(t_k),t_k)$. Then, there exist monotonically increasing functions $\beta_1 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$, $\beta_2 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ such that $\|u(t)\| \leq \beta_1(\varepsilon_k^2)$ and $\|\dot{u}(t)\| \leq \beta_2(\varepsilon_k^2)$, $\forall t \in [t_k, \min\{t_{k+1}, t_k+T\})$, $\forall k \in \mathbb{N}$.*

Proof. Note first that, $\forall i \in \{1,2\}$ and for any $k \in \mathbb{N}$, $u_i(t)$ for $t \in [t_k,t_{k+1})$ is determined by solving the unconstrained strictly convex optimization problem (10). Thus, the stationarity condition $\frac{\partial}{\partial a}J_i(a) = 0$ is necessary and sufficient for a to be the optimizer of problem (10) for $i \in \{1,2\}$. As a result, the optimizers of problem (10) are the solutions of the equation $\mathbf{H}_i\bar{\mathbf{a}}_i(k) = D_i(k)$, where

$$\begin{aligned}D_i(k) =& 2\begin{bmatrix}\langle \bar{u}_i,\tau^1\rangle & \langle \bar{u}_i,\tau^2\rangle & \dots & \langle \bar{u}_i,\tau^p\rangle\end{bmatrix}^\top \\ & - 2(1+\delta_i)\mu_i\begin{bmatrix}\frac{T^2}{2} & \frac{T^3}{3} & \dots & \frac{T^{p+1}}{p+1}\end{bmatrix}^\top,\end{aligned}$$

and $\mathbf{H}_i$ is the Hessian matrix of $J_i(.)$. As $\mathbf{H}_i$ is invertible, there is a unique optimal solution $\bar{\mathbf{a}}_i(k)$ to the problem (10) and is equal to $\bar{\mathbf{a}}_i(k) = \mathbf{H}_i^{-1}D_i(k)$.

Now, note that, $V(\hat{X}(t),t) \leq V(\hat{X}(t_k),t_k)$ for all $t \in [t_k, \min\{t_{k+1}, t_k+T\})$ and for any $k \in \mathbb{N}$ as $\dot{V}(\hat{X},t) = -\Sigma(\hat{X},t) \leq 0$ where $\hat{X}$ evolves as (8). As $\hat{X}(t_k) = X(t_k)$, $V(\hat{X}(t_k),t_k) = V(X(t_k),t_k)$ for each $k \in \mathbb{N}$. According to Remark 4, we can say that there exists a class $\mathscr{K}$ function $\alpha'(.) > 0$ such that $\|\hat{X}(t)\| \leq \alpha'(\varepsilon_k^2)$ for all $t \in [t_k, \min\{t_{k+1}, t_k+T\}]$ for each $k \in \mathbb{N}$. This implies that, for each $i \in \{1,2\}$, for all $\tau \in [0, \min\{t_{k+1}-t_k, T\})$ and $\forall k \in \mathbb{N}$, $|\bar{u}_i(\tau)|$ is upper bounded by a monotonically increasing positive real valued function of ε_k^2. By using this fact, we can say that there exists a monotonically increasing function $\beta'(.)$ such that $\|\mathbf{a}(k)\| \leq \beta'(\varepsilon_k^2)$, $\forall k \in \mathbb{N}$. Thus, we can say that there exists monotonically increasing functions $\beta_1,\beta_2 : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ such that $\forall t \in [t_k, \min\{t_{k+1}, t_k+T\})$, $\forall k \in \mathbb{N}$, $\|u(t)\| \leq \|\mathbf{a}(k)\| \left\|\begin{bmatrix}1 & t-t_k & \dots & (t-t_k)^p\end{bmatrix}^\top\right\| \leq \beta_1(\varepsilon_k^2)$ and $\|\dot{u}(t)\| \leq \|\mathbf{a}(k)\| \left\|\begin{bmatrix}0 & 1 & \dots & p(t-t_k)^{p-1}\end{bmatrix}^\top\right\| \leq \beta_2(\varepsilon_k^2)$. This proves Lemma 5. □

Now, we introduce the key result of this paper that shows non-Zeno behavior of IETs and UU boundedness of the tracking error.

Theorem 6. *(Absence of Zeno behavior and UU boundedness of tracking error). Consider system (16). Suppose Assumption (A1) holds. Then,*

- *there exists a uniform positive lower bound on the IETs, $t_{k+1}-t_k$ for $k \in \mathbb{N}$, that depends on the initial tracking error.*
- *moreover, the lower bound on the IETs converges to a positive real number, which is independent of the initial tracking error, in finite time.*
- *there exists a UU bound on the tracking error.*

Proof. Note first that, for each $k \in \mathbb{N}$, $\dot{V}(X(t_k^+),t_k^+) = -\Sigma(X(t_k),t_k)$ as $e(t_k^+) = 0$. Hence, according to the ETR (15) for each $k \in \mathbb{N}$, the IET $t_{k+1}-t_k$ must be greater than the time it takes $\Lambda(X,e,t)$ to grow from 0 to $(1-\sigma)\Sigma(X,t)$.

Let $\varepsilon_k^2 := V(X(t_k),t_k)$. Given Lemma 3 and (14), we can find an upper bound on $\Lambda(X,e,t)$ as follows,

$$\Lambda(X,e,t) \leq L(\varepsilon_k^2)\|e\|, \quad \forall t \in [t_k,t_{k+1}),$$

where $L:\mathbb{R}_{>0}\to\mathbb{R}_{>0}$ is a continuous function defined as,

$$L(R)\geq\max_{V\leq R}\left\|\frac{\partial V}{\partial X}G(X)\right\|.$$

According to Remark 4, the right hand side of the above inequality exists as any sub-level set of V is compact. Also note that the ETR (15) implies that for each $k\in\mathbb{N}$, $V(X(t_k),t_k)\geq\varepsilon^2$. Hence, we can say that for any $k\in\mathbb{N}$, the IET $t_{k+1}-t_k$ is lower bounded by the time it takes $\|e\|$ to grow from 0 to $(1-\sigma)\frac{\mathcal{V}}{L(\varepsilon_k^2)}$, where $\mathcal{V}>0$ is the same constant mentioned in the proof of Lemma 3.

Next, note that, $\hat{u}(X,t)$ is a continuously differentiable function of time. Thus, by using Lemma 3, we can also say that $\left\|\dot{\hat{u}}(X,t)\right\|$ is upper bounded by a monotonically increasing positive real valued function of ε_k^2, $\forall t\in[t_k,t_{k+1})$, $\forall k\in\mathbb{N}$.

Then, $\forall t\in[t_k,\min\{t_k+T,t_{k+1}\})$, $\forall k\in\mathbb{N}$,

$$\frac{d}{dt}\|e(t)\|\leq\|\dot{e}(t)\|\leq\|\dot{u}(t)\|+\left\|\dot{\hat{u}}(X,t)\right\|\leq\alpha_e(\varepsilon_k^2),$$

for some monotonically increasing function $\alpha_e:\mathbb{R}_{>0}\to\mathbb{R}_{>0}$. The last inequality follows from Lemma 5. Thus, for each $k\in\mathbb{N}$, $t_{k+1}-t_k$ is lower bounded by $(1-\sigma)\frac{\mathcal{V}}{\alpha_e(\varepsilon_k^2)L(\varepsilon_k^2)}>0$. Now, from Lemma 3, we know that $\varepsilon_1^2\geq\varepsilon_k^2$, for $k\in\mathbb{N}$, where $\varepsilon_1^2:=V(X(t_1),t_1)\geq\varepsilon^2$. Hence, we can say that the IETs, for $k\in\mathbb{N}$, are uniformly lower bounded by $(1-\sigma)\frac{\mathcal{V}}{\alpha_e(\varepsilon_1^2)L(\varepsilon_1^2)}>0$. Thus, we complete the proof of the first statement of this result.

As the IETs do not exhibit Zeno behavior, the ETR (15) along with Lemma 3 implies that $\dot{V}(X(t),t)<-\sigma\Sigma(X(t),t)<0$ for all $t\geq t_0$ such that $V(X(t),t)\geq\varepsilon^2$. Thus, $\exists\bar{t}\in[t_0,\infty)$ such that $V(X(t),t)\leq\varepsilon^2\ \forall t\geq\bar{t}$. Under Assumption **(A1)**, this implies that the tracking error X is uniformly ultimately bounded. Moreover, we can also say that there exists a finite $\bar{k}\in\mathbb{N}$ such that $\varepsilon_k^2=\varepsilon^2$ for all $k\geq\bar{k}$. Thus, the lower bound on the IETs converges to $(1-\sigma)\frac{\mathcal{V}}{\alpha_e(\varepsilon^2)L(\varepsilon^2)}>0$ in finite time and events. Thus, we prove the last two statements of this theorem. □

Note that the UU bound on the tracking error can be made arbitrarily small by choosing the design parameter ε arbitrarily small. Note also that the first statement in Theorem 6 implies that the IETs do not exhibit Zeno behavior. Another way of ensuring non-Zeno behavior of IETs is by explicitly imposing a positive lower bound on IETs while designing the ETR. However, choosing an explicit lower bound that also guarantees the tracking objective is not easy.

V. Simulation and Experimental Results

In this section, we present results of our simulations and experiments of the proposed ETPC for trajectory tracking. We compare the proposed method with time-triggered control (TTC) and the zero-order hold based ETC algorithm described in [5]. We present the simulation and experimental results for four reference trajectories that were generated using the unicycle model (2). The resulting paths in these four cases are shown in Figure 2. The reference velocity for these trajectories is 15cm/s. In paths 1 and 3, the reference

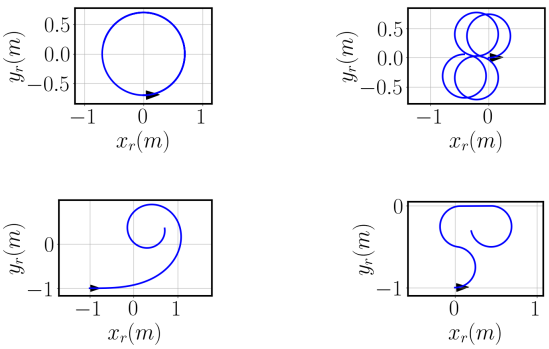


Fig. 2: Reference trajectories under consideration a) Path 1 , b) Path 2, c) Path 3, d) Path 4

angular velocity, ω_r is constant and smoothly changing, respectively, while in paths 2 and 4, ω_r is piecewise constant.

Evaluation metrics: Let T_e be the total simulation/experiment time duration. We define the transient period, (TP) as the time interval $[0,T_c]$, where

$$T_c:=\min\{t\geq 0\ :V(t)\leq\varepsilon^2\},$$

that is T_c is the time the Lyapnuov function V takes to first hit a value below ε^2. We define the steady state period, (SS) as the time interval $(T_c,T_e]$. Let N_t and N_s be the total number of events in TP and SS, respectively.

We compare the proposed ETPC with ETC and TTC in terms of the number of events, N_t and N_s, and the convergence time T_c. In order to do a fair comparison with TTC, we compare ETC and ETPC against TTC with two different transmission frequencies/periods. In particular, henceforth, TTC1 and TTC2 refer to TTC with average transmission frequency over T_e for ETC and ETPC respectively, in the corresponding simulation or experiment.

In general, the UU bound of $V(t)$ for TTC1 and TTC2 is higher than ε^2. In practical experiments with ETC and ETPC also the UU bound is often higher than ε^2. This is due to several unmodeled features, including sampling rate for the motion capture system (restricted to 240 frames per second here), measurement latency, error in obtained pose information, computation times for solving the optimization problem (9), communication delays and latency, delay introduced by onboard serial communication on robot, environmental conditions such as slip and non-uniform surface friction. Even the kinematics of the robot may not exactly be unicycle model. Given all this, another evaluation metric we employ is $\bar{\varepsilon}^2\geq\varepsilon^2$, the UU bound for $V(t)$. Specifically, we define $\bar{\varepsilon}^2$ as

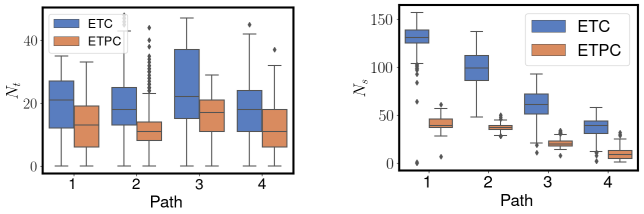
$$\bar{\varepsilon}^2:=\max_{t\geq T_c}\{V(t)\}.$$

A. Simulations

Simulations were done for the system (16) with four reference trajectories that generate the paths shown in Figure 2. In the simulations, the integration time step was a fixed value of 5ms for all the algorithms. The design parameters were chosen as $\gamma=100$, $c_1=0.02$, $c_2=0.05$, $c_3=0.01$, $\sigma=0.5$

and $\varepsilon = 0.1$. The prediction time horizon for ETPC, T was chosen as 1 second and polynomial degree was chosen as 3. The initial pose error was sampled uniformly from the set $[(-2\text{m}, 2\text{m}), (-2\text{m}, 2\text{m}), (-0.2\text{radians}, 0.2\text{radians})]$ to get a set of 1000 initial conditions. Simulations were conducted for each algorithm, for each of the four paths in Figure 2, for each of these 1000 initial conditions.

Figure 3 depicts the number of events for paths 1 to 4 in the transient and steady state period for ETC and ETPC. It



(a) Number of events in TP (b) Number of events in SS

Fig. 3: Comparison of number of events for simulated paths for algorithms under consideration.

is observed that the median of N_t for ETPC is reduced by 38.1%, 38.9%, 22.8% and 38.9% in comparison to median of N_t for ETC for paths 1 to 4, respectively. Similarly, the median of N_s for ETPC shown in Figure 3 is reduced by 70.3%, 62.7%, 67.3% and 77% in comparison to median of N_s for ETC for paths 1 to 4, respectively. The third quartile of N_s for all paths is much lower with ETPC than even the first quartile with ETC. Thus, we can say that our algorithm requires far fewer number of events than ETC in steady state and also, to a lesser extent, during the transient period.

Figure 4 shows the UU bound of V for TTC1 and TTC2 algorithms for all the paths. The UU bound of V for ETC and ETPC is within numerical tolerance of ε^2 bound. TTC1 and TTC2 algorithms often give a UU bound in the order of hundreds. Thus, we can conclude that the tracking performance with ETC or ETPC is significantly better than that of TTC with comparable frequency of transmissions.

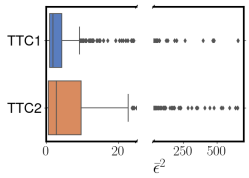
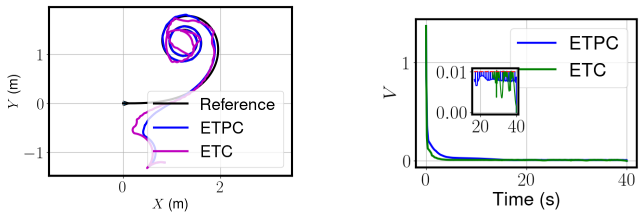


Fig. 4: UU bound for all simulated paths.

Figures 5 illustrates the result of one simulation for Path 3 with the initial pose error $(-1.02\text{m}\ , 1.08\text{m}\ , 0.142\text{rad})$. Figure 5a shows the reference path and the paths traced by ETC and ETPC while Figure 5b shows the evolution of Lyapunov function. It is observed that the error reduces over time and then oscillates to ensure that V stays within ε^2 bound. It is observed that the Lyapunov function under both ETC and ETPC stays within the ε^2 bound once it enters it. Notice that in Figure 6, θ_e oscillates significantly even at the end of the simulation. However the corresponding V stays within ε^2 bound once it enters it as seen in Figure 5b. This happens because the contribution of θ_e to the Lyapunov function V in (11) is significantly low with $1/(2\gamma) = 1/200$.



(a) Path traced by robot (b) Evolution of V

Fig. 5: Results of a simulation of robot tracking the reference trajectory that generates Path 3.

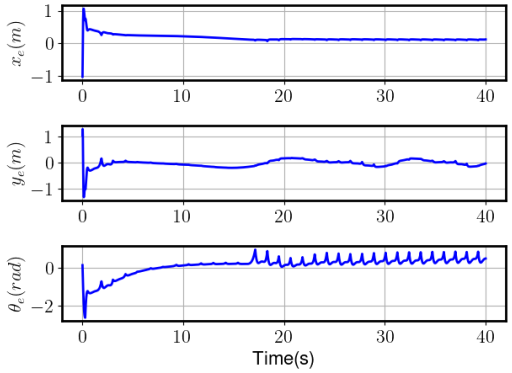


Fig. 6: Evolution of pose error for simulated Path 3 with initial pose error of $(-1.02\text{ m}, 1.08\text{ m}, 0.142\text{rad})$ with ETPC.

B. Practical Experiments

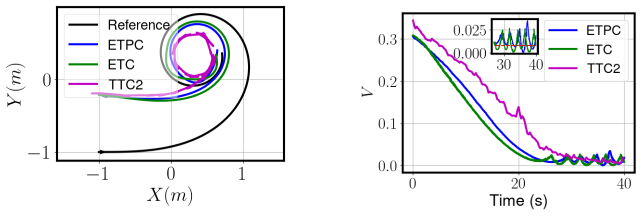
Experiments have been conducted on a 3pi+ 32U4 robot manufactured by Pololu Robotics and ElectronicsThe robot is equipped with two quadrature encoders, which are utilized for robot level wheel velocity control done using a conventional PID controller. The microcontroller on the robot is an AtMega32U4 with 28kB of memory available to the user. An RNXV WiFly module is interfaced with the robot using appropriate electronics for wireless communication with a desktop computer. The desktop computer has a 64-bit Windows 11 operating system with installed RAM of 40 GB and an Intel i7-8700 CPU with clock speed of 3.20 GHz. The desktop computer is also interfaced with OptiTrack motion capture system [25], which provides the pose measurements of the robot, with mean error in position of 5.4 mm. The sampling rate of the motion capture system is 240 frames per second, which translates to a sampling period of about 4.1ms. The motion capture system communicates the measurements over a wired single hop network connection and the latency in this communication could be

up to 1ms. The desktop computer monitors the ETR and also computes the coefficients of ETPC whenever required. This computational latency on the computer depends on the prediction horizon T and integration time step and could go up to 25ms. After an event occurs and the desktop computer computes the ETPC coefficients, it communicates them to the robot wirelessly. The sum of latency in the wireless network, control loops on the robot and the latency caused by onboard serial communication on the robot is on average between 15ms to 20ms but sometimes could go up to 100 ms.

The design parameters for experiments were chosen so that ETC and ETPC have a similar UU bound on V, denoted as $\bar{\varepsilon}^2$. In particular, the chosen parameters are $\gamma = 1$, $c_1 = 0.5$, $c_2 = 0.8$, $c_3 = 0.7$, and $\sigma = 0.9$. All quantities to be communicated are restricted to a precision of two decimal places. In our experiments, we use Transmission Control Protocol (TCP) which is an upper layer protocol to the Internet Protocol(IP). The size of each IP packet is fixed at 64 Bytes, in which 46 Bytes are reserved for data [26]. For ETC, TTC1 and TTC2, the size of payload (actual intended message) is 4 Bytes while it is 16 Bytes for ETPC. A packet for tranmission is constructed by padding the actual payload to acheive the packet sizes required by the protocol. Thus, in comparison to ETC, TTC1, and TTC2, ETPC communicates a lot more information in a single packet less often without affecting communication overheads in each packet.

For each path shown in Figure 2, two initial pose errors were considered and for each initial pose error and each algorithm, 10 experiments were conducted. The specific initial pose errors, in (m,m,radians), were chosen as $(0,0,0)$, $(1,0.2,1.57)$ for Path 1; $(0,0,0)$, $(0.2,0.2,1.57)$ for Path 2; $(0,0,0)$, $(0,-0.8,0)$ for Path 3; and $(0,0,0)$, $(-0.05,-0.5,-0.02)$ for Path 4.

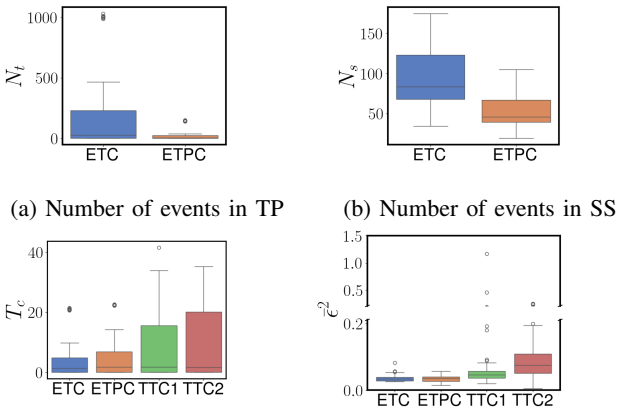
Fig. 7a shows the path traced by ETC, ETPC and TTC2 for Path 3. It is seen that the robot eventually starts tracking the path with ETC and ETPC algorithms as seen by the corresponding decrease in Lyapunov function as shown in Fig. 7b with V eventually staying within a small uniform ultimate bound. For TTC2, we see that the UU bound on V is much higher than for ETC and ETPC and the tracking behaviour is not good. Note that due to the unmodeled high computational and communication latency as well as other disturbances and modeling errors, the UU bound of V for ETC and ETPC are also higher than the designed ε^2.



(a) Path traced by robot (b) Evolution of V

Fig. 7: Results of an experiment of robot tracking the reference trajectory that generates Path 3.

In transient period, the median of N_t for ETPC is reduced by 80.4% in comparison to the median of N_t for ETC as seen in Figure 8a. Similarly, Figure 8b shows the number of steady state events for all paths. It is observed that the median of N_s for ETPC is reduced by 45% in comparison to median of N_s for ETC. Thus, we conclude that in both transient and steady state, our algorithm has fewer number of events. Figure 8c shows the convergence time to ε^2 bound for all algorithms. We see that, in all cases, ETC and ETPC have lower convergence times in comparison to TTC even when outliers are considered. The convergence times for ETC are slightly better than those of ETPC. We also observe that the UU bound as seen in Figure 8d is much lower for ETC and ETPC than for TTC and somewhat similar for ETC and ETPC. Figure 8d also indicates the robustness of the proposed ETPC method as it guarantees UU boundedness of V even in the presence of disturbances and modeling errors. From the suite of simulations and experiments, we can conclude that our algorithm has fewer number of events than ETC while ensuring similar tracking behaviour. This implies that the ETPC method improves the usage of communication resources compared to the ETC method.



(a) Number of events in TP (b) Number of events in SS

(c) Convergence to ε^2 bound (d) UU bound of V

Fig. 8: Results of practical experiments for all the paths and all algorithms.

VI. CONCLUSION

In this work, we presented an ETPC method for trajectory tracking for unicyle model of robots where the reference trajectory is modeled as the solution of a reference unicycle model. We designed an ETR that guarantees UU boundedness of tracking error and non-Zeno behavior of IETs. The proposed method works best in cases where communication is significantly more costly than computation. We illustrated the results through numerical simulations and experiments. We also showed that the number of events generated by the proposed controller is significantly less compared to a time-triggered controller and an event-triggered controller based on zero-order hold, while guaranteeing similar tracking performance. Future work includes control under input

disturbances, time delays, quantization of the parameters and multi-robot control.

VII. Appendix

A. Time Derivative of the Candidate Lyapunov Function V

Consider the candidate Lyapunov function V given in (11). We can determine the time derivative of V as follows,

$$\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial X}(F(X,t) + G(X)(\hat{u}(X,t) + e(t))),$$

where

$$\frac{\partial V}{\partial t} = -x_1 c_3 y_e(\dot{\omega}_r + \gamma y_e \dot{v}_r \text{sinc}\theta_e),$$

$$\frac{\partial V}{\partial X} = \begin{bmatrix} x_1 \\ x_1(-c_3\hat{\omega}(X,t) - c_3\gamma y_e v_r \text{sinc}\theta_e) + y_e \\ -x_1 c_3 y_e(\gamma y_e v_r \text{sinc}'\theta_e + c_2\gamma) + \frac{\theta_e}{\gamma} \end{bmatrix}^T.$$

Following equations (4), (6), and (7), we can simplify the above expression as follows,

$$\begin{aligned}\dot{V} =& -x_1 c_3 y_e(\dot{\omega}_r + \gamma y_e \dot{v}_r \text{sinc}\theta_e) + x_1(v_r\cos\theta_e - \hat{v} + y_e\hat{\omega}) \\ &+ (x_1(-c_3\hat{\omega} - c_3\gamma y_e v_r \text{sinc}\theta_e) + y_e)(v_r\sin\theta_e - x_e\hat{\omega}) \\ &+ (-x_1 c_3 y_e(\gamma y_e v_r \text{sinc}'\theta_e + c_2\gamma) + \frac{\theta_e}{\gamma})(\omega_r - \hat{\omega}) \\ &+ \frac{\partial V}{\partial X}G(X)e(t).\end{aligned}$$

Note that, for ease of representation, here we omit the arguments of $\hat{v}(X,t)$ and $\hat{\omega}(X,t)$.

$$\begin{aligned}\dot{V} =& -x_1 c_3 y_e(\dot{\omega}_r + \gamma y_e \dot{v}_r \text{sinc}\theta_e + \gamma v_r \text{sinc}\theta_e(v_r\sin\theta_e - x_e\hat{\omega})) \\ &- x_1 c_3 y_e(\omega_r - \hat{\omega})(\gamma y_e v_r \text{sinc}'\theta_e + c_2\gamma) \\ &+ x_1(v_r\cos\theta_e - \hat{v} + y_e\hat{\omega} - c_3\hat{\omega}(v_r\sin\theta_e - x_e\hat{\omega})) \\ &+ y_e(v_r\sin\theta_e - x_e\hat{\omega}) + (\omega_r - \hat{\omega})(\frac{\theta_e}{\gamma}) + \frac{\partial V}{\partial X}G(X)e(t).\end{aligned}$$

$$\begin{aligned}\dot{V} &= x_1(v_1 - \hat{v}) + y_e\hat{\omega}(x_1 - x_e) + y_e v_r\sin\theta_e \\ &\quad - (\gamma y_e v_r \text{sinc}\theta_e + c_2\gamma\theta_e)\frac{\theta_e}{\gamma} + \frac{\partial V}{\partial X}G(X)e(t), \\ &= -(c_1 x_1^2 + c_2\theta_e^2 + c_3\hat{\omega}^2(X,t)y_e^2) + \frac{\partial V}{\partial X}G(X)e(t), \\ &= -\Sigma(X,t) + \Lambda(X,e,t),\end{aligned}$$

where $\Sigma(.)$ and $\Lambda(.)$ are defined as given in (13) and (14), respectively.

References

[1] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[2] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3270–3285.

[3] M. Lemmon, "Event-triggered feedback in control, estimation, and optimization," in *Networked control systems*. Springer, 2010, pp. 293–358.

[4] P. Tallapragada and N. Chopra, "On event triggered tracking for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2343–2348, 2013.

[5] R. Postoyan, M. C. Bragagnolo, E. Galbrun, J. Daafouz, D. Nešić, and E. B. Castelan, "Event-triggered tracking control of unicycle mobile robots," *Automatica*, vol. 52, pp. 302–308, 2015.

[6] C. Xie, Y. Fan, and J. Qiu, "Event-based tracking control for nonholonomic mobile robots," *Nonlinear Analysis: Hybrid Systems*, vol. 38, p. 100945, 2020.

[7] C. Santos, F. Espinosa, E. Santiso, and D. Gualda, "Lyapunov self-triggered controller for nonlinear trajectory tracking of unicycle-type robot," *International Journal of Control, Automation and Systems*, vol. 18, no. 7, pp. 1829–1838, 2020.

[8] C. Santos, M. Mazo Jr, and F. Espinosa, "Adaptive self-triggered control of a remotely operated p3-dx robot: Simulation and experimentation," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 847–854, 2014.

[9] K. G. Vamvoudakis, A. Mojoodi, and H. Ferraz, "Event-triggered optimal tracking control of nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 4, pp. 598–619, 2017.

[10] Q. Cao, Z. Sun, Y. Xia, and L. Dai, "Self-triggered MPC for trajectory tracking of unicycle-type robots with external disturbance," *Journal of the Franklin Institute*, vol. 356, no. 11, pp. 5593–5610, 2019.

[11] P. Gao, G. Wang, Y. Ji, Q. Li, J. Zhang, Y. Shen, and P. Li, "Event-triggered tracking control scheme for quadrotors with external disturbances: Theory and validations," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8929–8935.

[12] P. Zhang, T. Liu, and Z.-P. Jiang, "Tracking control of unicycle mobile robots with event-triggered and self-triggered feedback," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2261–2276, 2023.

[13] E. García and P. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Transactions on Automatic Control*, vol. 58, pp. 422–434, 02 2013.

[14] W. Heemels and M. Donkers, "Model-based periodic event-triggered control for linear systems," *Automatica*, vol. 49, no. 3, pp. 698–711, 2013.

[15] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.

[16] H. Li, W. Yan, and Y. Shi, "Triggering and control codesign in self-triggered model predictive control of constrained systems: With guaranteed performance," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 4008–4015, 2018.

[17] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, "On the trade-off between communication and control cost in event-triggered dead-beat control," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2973–2980, 2017.

[18] A. Li and J. Sun, "Resource limited event-triggered model predictive control for continuous-time nonlinear systems based on first-order hold," *Nonlinear Analysis: Hybrid Systems*, vol. 47, p. 101273, 2023.

[19] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered model predictive control for nonlinear input-affine dynamical systems via adaptive control samples selection," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 177–189, 2017.

[20] A. Rajan and P. Tallapragada, "Event-triggered parameterized control for stabilization of linear systems," in *62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 3903–3910.

[21] ——, "Event-triggered parameterized control of nonlinear systems," *IEEE Control Systems Letters*, vol. 8, pp. 1673–1678, 2024.

[22] A. Rajan, A. Kattepur, and P. Tallapragada, "Co-design of polynomial control law and communication scheduling strategy for multi-loop networked control systems," *Indian Control Conference*, 2024.

[23] S. Das, S. Ganguly, M. Anjali, and D. Chatterjee, "Towards continuous-time mpc: A novel trajectory optimization algorithm," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 3276–3281.

[24] Z.-P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, no. 7, p. 1393 – 1399, 1997.

[25] "NaturalPoint, "Motion Capture Systems -OptiTrack Webpage."," https://optitrack.com, accessed: 2023-08-12.

[26] C. Hornig, "A standard for the transmission of ip datagrams over ethernet networks," Internet Requests for Comments, Symbolics Cambridge Research Center, RFC 894, 4 1984. [Online]. Available: https://www.rfc-editor.org/rfc/rfc894.txt