

# Autonomous Drone Racing: Time-Optimal Spatial Iterative Learning Control within a Virtual Tube

Shuli Lv, Yan Gao, Jiaying Che, Quan Quan\*

**Abstract**—It is often necessary for drones to complete delivery, photography, and rescue in the shortest time to increase efficiency. Many autonomous drone races provide platforms to pursue algorithms to finish races as quickly as possible for the above purpose. Unfortunately, existing methods often fail to keep training and racing time short in drone racing competitions. This motivates us to develop a high-efficient learning method by imitating the training experience of top racing drivers. Unlike traditional iterative learning control methods for accurate tracking, the proposed approach iteratively learns a trajectory online to finish the race as quickly as possible. Simulations and experiments using different models show that the proposed approach is model-free and is able to achieve the optimal result with low computation requirements. Furthermore, this approach surpasses some state-of-the-art methods in racing time on a benchmark drone racing platform. An experiment on a real quadcopter is also performed to demonstrate its effectiveness.

## I. INTRODUCTION

Drones are increasingly used for delivery, photography, search, and rescue [1], often in complex real-world environments. Propelled by these applications, autonomous drones have made significant progress in navigation and control, but the performance is still far from that of the human pilot. Therefore, many studies and innovations are needed to fully exploit the physical capabilities of drones. For this purpose, several autonomous drone racing competitions have been launched[2], such as the AlphaPilot Challenge [3], AirSim Drone Racing Lab [4], and Intelligent UAV Racing Championship (drone.sjtu.edu.cn) by SEIEE in 2021 (Our lab ranked first in a simulation race and second in a real-flight race). These competitions enable the generation of racing track orchestration and come with a suite of gate assets. As shown in Fig. 1, the drone is required to pass through several gates in the shortest time to win the race.

Given the conditions of the known environment, pushing drones to their physical limits presents challenges to researchers. There are also many existing solutions to autonomous competitions, including the use of continuous-time polynomial trajectory planning [5], the time-discrete trajectories method with reinforcement learning (RL) methods [6], [7], search and sampling-based methods [8], and model-based optimization methods [9]. Continuous-time polynomial trajectory planning has high computational efficiency, but

these polynomials are inherently smooth, therefore, cannot represent rapid states or input changes, making polynomial methods suboptimal [10]. Optimization methods allow drones to select any input within each discrete time step range but often have a great demand for computing resources.

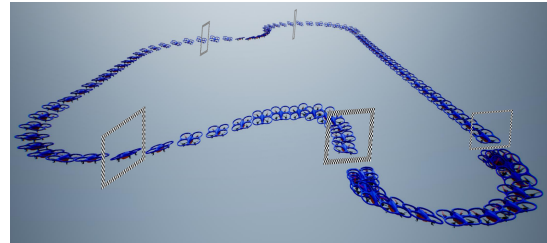


Fig. 1. The nonlinear drone's poses and location captured in RflySim (rflsim.com) [11].

Accurate model information of the controlled object is often challenging to obtain. Therefore, model-free control methods are necessary. Iterative learning control (ILC) is an essential model-free control method and is suitable for dynamic systems with repetitive operation characteristics [12], [13]. ILC can complete accurate tracking tasks within a limited time[14]. With ILC, the tracking error in the last iteration is used to correct the unsatisfactory control command, thereby improving the tracking accuracy of the system iteratively[15]. However, the existing ILC is only used for a specific trajectory tracking problem [16] and cannot be directly applied to drone races.

In this paper, a model-free online time-optimal spatial ILC approach inspired by the training experience of top racing drivers is proposed for drone racing. First, a virtual tube is established based on the autonomous drone racing scene. Then, a time-optimal spatial ILC approach is proposed, with the drone's trajectory within the virtual tube. The primary control principle is to make acceleration and deceleration decisions according to different positions of the drone in the virtual tube. Finally, the proposed approach can iteratively learn the control commands to pass through gates quickly.

The contributions of this paper are mainly divided into two aspects.

- Following top racers' experience, a new and high-efficiency model-free approach to autonomous drone races is proposed. Through theoretical analysis and experimental comparisons, it is found that this approach can achieve the optimal result with low computation requirements.
- A new contribution to ILC theory to broaden the scope

Shuli Lv, Yan Gao, Jiaying Che, Quan Quan (Corresponding Author) are with School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, P.R. China lvshuli@buaa.edu.cn, buaa\_gaoyan@buaa.edu.cn, chejiaying@163.com, qq\_buaa@buaa.edu.cn

This work was supported by the National Natural Science Foundation of China under Grant 61973015.

of its applications. The proposed approach addresses time-optimal control problems with a new controller structure, which offers a new start to exploit the potential of ILC.

## II. PROBLEM FORMULATION

In this section, “path” is used to replace “trajectory” for the elimination of time dimensional. The drone model will be transformed into the space domain from the time domain. The general racing missions and problem transformations are also presented.

### A. Drone Model

Many groups or companies have designed open-source Semi-Autonomous Autopilots (SAAs) or offered SAAs with Software Development Kits (SDKs) for drones. Not only can it avoid the trouble of modifying the low-level source code of autopilots, but also it can utilize reliable commercial autopilots to achieve targets. This will simplify the complete design. With SAAs, for simplicity, the drone is considered as a mass point in  $\mathbb{R}^2$  as

$$\begin{cases} \dot{\mathbf{p}}(t) = \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) = \mathbf{a}(t) \end{cases}, \quad (1)$$

where  $\mathbf{p}(t)$ ,  $\mathbf{v}(t)$ ,  $\mathbf{a}(t) \in \mathbb{R}^2$  indicate the position, velocity and acceleration of the drone. Then, its acceleration is decided by

$$\mathbf{a}(t) = -\tau(\mathbf{v}(t) - \mathbf{v}_c(t)), \quad (2)$$

where  $\mathbf{v}_c(t) \in \mathbb{R}^2$  indicates the control command. The maneuver parameter  $\tau > 0$  is proportional to the drone’s maneuverability; different drones always have different  $\tau$  to represent their mobility. The drone model proposed in this part is only used for analysis. In fact, the proposed approach design does not rely on the model information. During the experiments and simulations,  $\tau$  is unknown and the model in AirSim [17] cannot be directly represented by (1) and (2) as well.

The command  $\mathbf{v}_c$  should be bounded. As a result, the velocity  $\mathbf{v}$  is also bounded. Therefore, it is necessary to make the control command  $\mathbf{v}_c$  subject to a saturation constraint as

$$\begin{aligned} \mathbf{v}_c &= \text{sat}(\mathbf{v}'_c, v_{\max}), \\ &= \kappa_{v_m}(\mathbf{v}'_c) \mathbf{v}'_c, \end{aligned} \quad (3)$$

$$\text{sat}(\mathbf{v}'_c, v_{\max}) \triangleq \begin{cases} \mathbf{v}'_c & \|\mathbf{v}'_c\| \leq v_{\max} \\ v_{\max} \frac{\mathbf{v}'_c}{\|\mathbf{v}'_c\|} & \|\mathbf{v}'_c\| > v_{\max} \end{cases}, \quad (4)$$

$$\kappa_{v_m}(\mathbf{v}'_c) \triangleq \begin{cases} 1 & \|\mathbf{v}'_c\| \leq v_{\max} \\ \frac{v_{\max}}{\|\mathbf{v}'_c\|} & \|\mathbf{v}'_c\| > v_{\max} \end{cases},$$

where  $\mathbf{v}'_c \in \mathbb{R}^2$  indicates the original control command, and  $v_{\max}$  indicates the maximum allowable control command. With the spatial differentiator map  $\nabla = \frac{d}{dl}$  mentioned in [18], [19], the drone model (1) is mapped in the space domain as

$$\begin{cases} \nabla \mathbf{p}(l) = \frac{1}{v(l)} \mathbf{v}(l) \\ \nabla \mathbf{v}(l) = -\frac{\tau}{v(l)} (\mathbf{v}(l) - \mathbf{v}_c(l)) \end{cases}. \quad (5)$$

in which  $v(l) > 0$  indicates the drone’s tangential pace along the path. In order to facilitate the transformation between  $t$  and  $l$ , the relationship between the temporal coordinate  $t$  and spatial coordinate  $l$  is obtained for later analysis. For  $v = dl/dt$ ,  $l = \int_0^t v(s) ds$  is obtained. With  $v > 0$ ,  $l$  strictly increases, so with the function mapping  $l = f(t)$  given, the inverse mapping exists globally. Since  $\tau > 0$ , given any constant  $\mathbf{v}_c(t)$ ,  $\mathbf{v}(t) \rightarrow \mathbf{v}_c(t)$  as  $t \rightarrow \infty$ . The velocity can be written as

$$\mathbf{v}(l) = \mathbf{v}_c(l) + \Delta \mathbf{v}(l). \quad (6)$$

To simplify the proof process, an assumption is made.

**Assumption 1.** The perturbation  $\Delta \mathbf{v}(l)$  satisfies  $\|\Delta \mathbf{v}(l)\| \leq \varepsilon v(l)$ , where  $\varepsilon > 0$  is bounded.

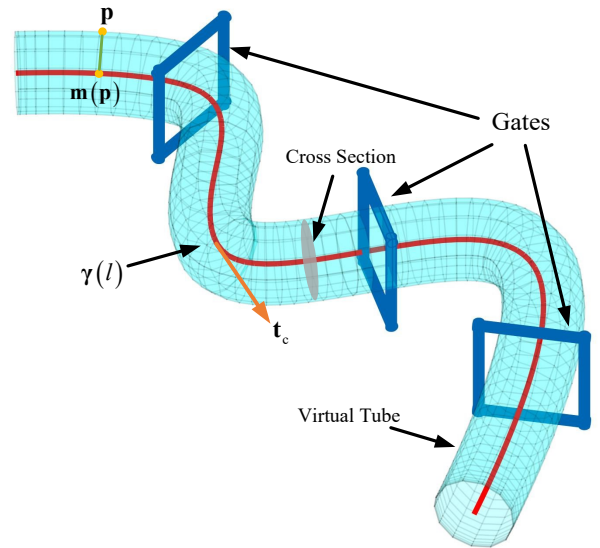


Fig. 2. Virtual tube suitable for the racing track.

### B. Racing Mission Description

Given the current standard drone competition environment, a task description is required. These competitions enable the generation of racing track orchestration and come with a structure of drone racing with gates as a set course. As is shown in Fig. 2, each gate has its pose and size, which indicate the landmarks in the race environment. During racing, the object is to pass through all the gates at minimum possible time, without collision.

Based on the above background, the following mission description is given. In the scene, there are  $N$  gates to be passed through. Since each gate has a crossing position, there are  $N+2$  waypoints (including the start point and end point)  $\tilde{\mathbf{p}} = [\tilde{\mathbf{p}}^0 \tilde{\mathbf{p}}^1 \dots \tilde{\mathbf{p}}^N \tilde{\mathbf{p}}^{N+1}]^T \in \mathbb{R}^{(N+2) \times 2}$  that must be passed through, as shown in Fig. 2. Let set  $\mathcal{G}$  indicate spatial distribution of the gates’ frame. The drone needs to pass through these gates in the shortest time while avoiding the collision.

From the above description, a spatial continuous virtual tube  $\mathcal{T}_v \in \mathbb{R}^2$  suitable for racing scenes is established to represent the spatial constraints. The waypoints form the path

$\gamma \in \mathbb{R}^2$ . The virtual tube is generated with the path and the constraints. The established virtual tube does not intersect with the gates' frame, namely  $\mathcal{T}_V \cap \mathcal{G} = \emptyset$ .

### C. Problem Transformation

According to the racing mission description, the drone racing problem becomes the drone passing through the virtual tube to reach the destination as soon as possible. Formally, the problem is formulated as a time-optimal problem

$$\begin{aligned} \min_{\mathbf{v}_c(l) \in \mathbb{R}^2, l \in [0, L]} T \\ \text{s.t. } \nabla \mathbf{p}(l) &= \frac{1}{v(l)} \mathbf{v}(l) \\ \nabla \mathbf{v}(l) &= -\frac{\tau}{v(l)} (\mathbf{v}(l) - \mathbf{v}_c(l)) \\ \|\mathbf{v}_c(l)\| &\leq v_{\max} \\ \mathbf{p}(l) &\in \mathcal{T}_V \\ L &= \int_0^T v(t) dt, \end{aligned} \quad (7)$$

in which  $L$  indicates the length of the virtual tube's center path along the unit tangent vector. This paper aims to achieve the goal of a drone passing through the virtual tube as quickly as possible by using the spatial ILC method. Obviously, the object here is definitely different from that of the traditional ILC.

## III. SPATIAL ITERATIVE LEARNING CONTROLLER DESIGN

This section first establishes a spatial virtual tube with the path and obstacle information to suit the racing scene. Furthermore, the time-optimal spatial ILC within the virtual tube is designed, inspired by human racers' training strategy. Convergence proof is made. Finally, why the spatial ILC can solve the time-optimal optimization is analyzed.

### A. Virtual Tube Modeling Based on Gates

In our previous work, the waypoints are used to obtain the path, and the gates' information help establish the virtual tube [20]. The process of establishing the virtual tube includes two parts: the *generator curve* and the *cross section*[21]. Intuitively, a virtual tube is generated by a cross section perpendicularly moving along the generator curve, and the surface is the boundary of the virtual tube [22]. The following related definitions of the virtual tube must be drawn to propose the controller.

1. **Generator curve**, denoted by  $\gamma$ , represents the race path of the racer.
2. **Virtual tube**, denoted by  $\mathcal{T}_V$ .
3. **The boundary of virtual tube**, denoted by  $\partial \mathcal{T}_V$ .
4. **Unit tangent vector**, denoted by  $\mathbf{t}_c$ , represents the unit tangent vector along the path  $\gamma$ .
5. **The radius of virtual tube**, denoted by  $r_t(l)$ , represents the distance from the generator curve to the boundary.
6. **The generator curve projection**, denoted by  $\mathbf{m}$ . For any given point  $\mathbf{p} \in \mathcal{T}_V$ , the generator curve projection is defined as  $\mathbf{m}(\mathbf{p}) = \tilde{\mathbf{p}}$ , where  $\tilde{\mathbf{p}} \in \gamma$ .

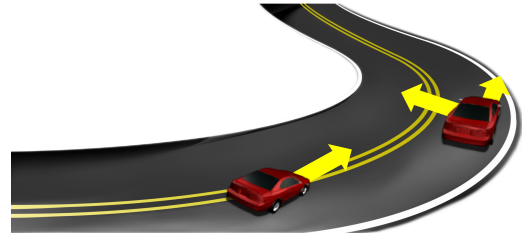


Fig. 3. The left car on the center is safe to accelerate and the right car near the boundary needs to decelerate to ensure safety.

There is a relationship shown as

$$\mathbf{t}_c^T (\mathbf{m}(\mathbf{p})) (\mathbf{p} - \mathbf{m}(\mathbf{p})) = 0, \quad (8)$$

where  $\mathbf{p} \in \mathcal{T}_V$ . Besides, an assumption is made on the proposed virtual tube.

**Assumption 2.** The drone can obtain the virtual tube information through measurement or prior knowledge.

For a drone located at any position within a virtual tube, there is a moving direction  $\mathbf{t}_c$  for it to race in the virtual tube. Because of the limitation of the space, how to generate a virtual tube is not presented here. A brief introduction will be given in the simulations.

### B. Spatial ILC Design

1) *Human Racer Strategy*: By referring to the training strategies of top human racing drivers in racing games, a spatial ILC scheme is designed based on the virtual tube established above. Racers are customarily required to know the path information in advance to plan a reasonable velocity distribution strategy for the actual racing scene. This speed distribution is designed based on location, taking into account factors such as curve radius, number of corners, and how the curves in the track are made up.

As described in [23], as the corner radius increases, the racer tends to take a faster forward speed, while when the radius decreases, the racer tends to slow down through the corner. If the car takes a large speed when cornering, it is prone to deviate from the track [24], even failing the race. As shown in Fig. 3, when the car is in the center of the road, it is safe to accelerate. On the other hand, when the car is close to the boundary, it is better to decelerate. Otherwise, it may cause the car to run out of the track. According to the principle above, the time-optimal spatial ILC is designed.

2) *Spatial Iterative Learning Control*: Based on the established virtual tube, the controller is designed as

$$\mathbf{v}'_c(l) = \mathbf{v}_h(l) + \mathbf{v}_p(l), \quad (9)$$

with two terms shown in Fig. 4.  $\mathbf{v}_h(l)$  is used for tangential pace control, and  $\mathbf{v}_p(l)$  is used for path convergence, namely helping the drone track the path  $\gamma$ .

(i) **Path Convergence**. Since the drone has a closet point on the virtual tube center curve, the error between the drone and the path is defined as

$$\mathbf{e}_p(l) = \mathbf{m}(\mathbf{p}(l)) - \mathbf{p}(l), \quad (10)$$

in which  $\mathbf{p}(l)$  represents the current position of the drone. Convergence control is based on the error. According to the analysis about the human racer strategy, the path convergence controller here is designed as

$$\mathbf{v}_p(l) = -k_0(l)v(l) \left( \frac{\partial \mathbf{m}}{\partial \mathbf{p}} - \mathbf{I}_2 \right)^T \mathbf{e}_p(l), \quad (11)$$

ensuring that the drone will not deviate from the virtual tube when the curvature of the center path is large or the tube is narrow. In (11),  $k_0(l)$  depends on the curvature and the width of the virtual tube, which is shown as

$$k_0(l) = k_2 + k_3 K(l) + k_4 \frac{1}{r_t(l)}, \quad (12)$$

where  $K(l)$  indicates the curvature,  $r_t(l)$  indicates the width of virtual tube and  $k_2, k_3, k_4 > 0$ . The above design implies that the path convergence control should be strengthened at a narrower or more curved virtual tube section. There exists a relationship [1]

$$\left( \frac{\partial \mathbf{m}}{\partial \mathbf{p}} \right)^T \mathbf{e}_p(l) \equiv 0, \quad (13)$$

so (11) is equivalent to

$$\mathbf{v}_p(l) = k_0(l)v(l) \mathbf{e}_p(l). \quad (14)$$

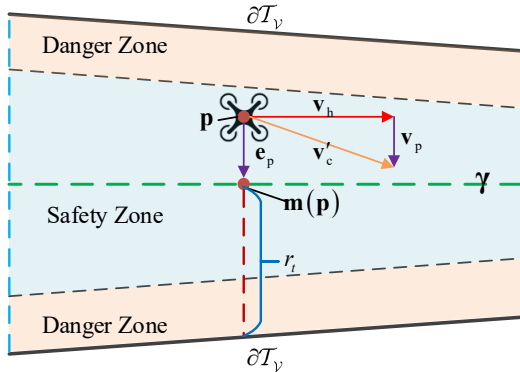


Fig. 4. The control command is divided into path convergence control and tangential pace control.

(ii) **Pace Controlling.** The pace controller is designed as

$$\mathbf{v}_h(l) = v(l) v^*(l) \mathbf{t}_c(l), \quad (15)$$

where  $v^*(l)$  is used for adjusting the tangential pace and  $\mathbf{t}_c(l)$  indicates the unit tangent direction at the location  $l$ .

Consistent with the idea of ILC, from the slow start to the continuous acceleration, each iteration will “summarize” the previous experience and constantly make adjustment dynamically at each location in the virtual tube to achieve the best. According to the training strategy of the racer, when  $\|\mathbf{e}_p(l)\|$  is large, it often implies that the race car is currently in a dangerous zone, a deceleration strategy has to be adopted to ensure safety. Otherwise, an acceleration strategy has to be performed. Based on the ideas above, a PD-type learning law for  $v^*$  is designed in the form of the spatial ILC,

$$v_{k+1}^*(l) = v_k^*(l) - \chi(k_p \|\mathbf{e}_{p,k}(l)\| + k_d \nabla \|\mathbf{e}_{p,k}(l)\|), \quad (16)$$

especially used for adjusting the contribution of  $\mathbf{v}_h$  in (9), where  $\|\mathbf{e}_{p,k}(0)\| = 0, k_p, k_d > 0, k$  indicates the iteration number and  $\chi$  is a nonlinear activation function shown in Fig. 5 which satisfies

$$\chi(x) = k_\chi(x)(x - x_{th}), 0 < \alpha \leq k_\chi(x) \leq \beta. \quad (17)$$

In Fig. 5, the threshold value  $x_{th} > 0$  represents the

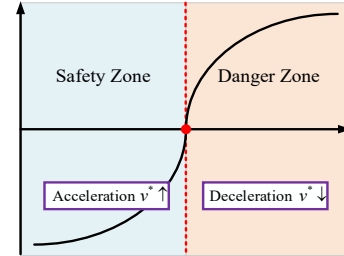


Fig. 5. The form of function  $\chi$  related to the error which divides the danger and safety zone with threshold  $x_{th}$ .

demarcation between the safety and the danger zone. When the tracking error is large, the drone is more likely to fly out of the boundary, corresponding to the right side of  $x_{th}$  in the function; on the contrary, it is safer, corresponding to the left.

With the error (10) and controller designed as (9), (11), (15) and (16), one has

$$\begin{aligned} \mathbf{e}_{p,k}^T \nabla \mathbf{e}_{p,k} &= \mathbf{e}_{p,k}^T \left( \frac{\partial \mathbf{m}}{\partial \mathbf{p}} - \mathbf{I}_2 \right) \nabla \mathbf{p} \\ &= \mathbf{e}_{p,k}^T \left( \frac{\partial \mathbf{m}}{\partial \mathbf{p}} - \mathbf{I}_2 \right) \frac{1}{v} (\mathbf{v}_c + \Delta \mathbf{v}) \\ &= -\mathbf{e}_{p,k}^T \frac{\kappa_{v_m}}{v} (\mathbf{v}_p + \mathbf{v}_h + \Delta \mathbf{v}) \\ &= -\kappa_{v_m} k_0 \mathbf{e}_{p,k}^T \mathbf{e}_{p,k} - \frac{\kappa_{v_m}}{v} \mathbf{e}_{p,k}^T \Delta \mathbf{v}. \end{aligned} \quad (18)$$

So

$$\nabla \|\mathbf{e}_{p,k}\| \leq -\gamma_1 \|\mathbf{e}_{p,k}\| + \gamma_2 \quad (19)$$

is obtained, where  $\gamma_1 = \kappa_{v_m}(v^*, \|\mathbf{e}_{p,k}\|) k_0, \gamma_2 \leq \varepsilon$  and (3), (4), (6), (8), (13) are utilized. It is easy to obtain

$$\frac{\partial (\gamma_1 \|\mathbf{e}_{p,k}\|)}{\partial \|\mathbf{e}_{p,k}\|} \geq \gamma_3 > 0, \gamma_4 \geq \frac{\partial (-\gamma_1 \|\mathbf{e}_{p,k}\|)}{\partial v^*} \geq 0, \quad (20)$$

when  $v^*$  and  $\|\mathbf{e}_{p,k}\|$  are bounded. So using the *mean value theorem*, (19) is written as

$$\nabla \|\mathbf{e}_{p,k}(l)\| \leq -\gamma_3 \|\mathbf{e}_{p,k}(l)\| + \gamma_4 v_k^*(l) + \gamma_5, \quad (21)$$

where  $\gamma_5 > 0$  is bounded.

**Theorem 1.** Under *Assumptions 1,2*, suppose a drone model satisfies (5), and the controller is designed as (9), with (11), (15) and (16). If  $|1 - k_\chi k_d \gamma_4| < 1$ , then the term  $v_k^*$  is uniformly ultimately bounded, as  $k \rightarrow \infty$ .

*Proof.* Since  $\|\mathbf{e}_{p,k}(0)\| = 0$ , (21) becomes

$$\|\mathbf{e}_{p,k}(l)\| \leq \int_0^l e^{-\gamma_3(l-s)} (\gamma_4 v_k^*(s) + \gamma_5) ds. \quad (22)$$

Further, with (17) and (18), the learning law (16) becomes

$$v_{k+1}^*(l) \leq (1 - k_\chi k_d \gamma_4) v_k^*(l) + (k_\chi k_d \gamma_3 - k_\chi k_p) \int_0^l e^{-\gamma_3(l-s)} (\gamma_4 v_k^*(s) + \gamma_5) ds - k_\chi k_d \gamma_5 + k_\chi x_{th}. \quad (23)$$

According to [25], if  $|1 - k_\chi k_d \gamma_4| < 1$ , then the term  $v_k^*$  is uniformly ultimately bounded, as  $k \rightarrow \infty$ . The proof of *Theorem 1* is completed.  $\square$

3) *Optimality Analysis*: In the following, optimality and parameter insensitive analysis is made on an arc tracked line with radius  $r > 0$  without loss of generality. Because an arc has the same curvature,  $\|e_p(l)\|$  and  $v(l)$  converge to constant after a sufficient number of learning trials, namely  $\|e_p(l)\| = \bar{e}_p, v(l) = \bar{v}, l \in [0, L]$ . This further implies  $\nabla \|e_p(l)\| = 0, l \in [0, L]$ . In this case,  $\bar{e}_p = x_{th}/k_p$  according to (16). The error is decomposed into two components

$$\nabla \|e_p(l)\| = \nabla \|e_p'(l)\| + \nabla \|e_p''(l)\|, \quad (24)$$

where the term  $\nabla \|e_p'(l)\|$  is determined by the pace speed control and  $\nabla \|e_p''(l)\|$  by the path convergence control. Intuitively, the term  $\nabla \|e_p'(l)\|$  has the form  $\nabla \|e_p'(l)\| = \frac{k'v(l)}{r+\bar{e}_p} = \frac{k'\bar{v}}{r+x_{th}/k_p}, k' > 0$ , which means a faster angle speed leading to a larger error. On the other hand, the term  $\nabla \|e_p''(l)\|$  has the form  $\nabla \|e_p''(l)\| = -k''\bar{e}_p = -k''x_{th}/k_p, k'' > 0$ . The fact  $\nabla \|e_p(l)\| = 0$  further

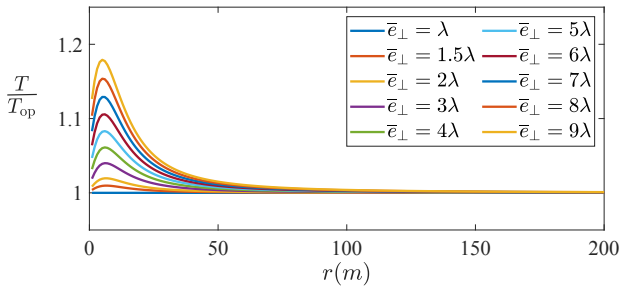


Fig. 6. Insensitivity of  $T/T_{op}$  to parameters.

implies  $k'\bar{v}/(r + x_{th}/k_p) = k''x_{th}/k_p$ . In the time optimal control, control saturation is often required to satisfy *Pontryagin Maximum Principle* [26]. Then,  $k'' = k'k_p v_{max} \left( (k')^2 + (r + x_{th}/k_p)^2 \right)^{-\frac{1}{2}}/x_{th}$  and  $x_{th}/k_p \geq \lambda > 0$  are obtained with *Assumption 1*, (3) and (4). Therefore, the lap time

$$T = \frac{L}{\bar{v}} = \frac{L(r + x_{th}/k_p) \left( (k')^2 + (r + x_{th}/k_p)^2 \right)^{\frac{1}{2}}}{rv_{max}}. \quad (25)$$

In particular, the tracked line is a straight line, namely  $r \rightarrow \infty$ , one has  $T \rightarrow L/v_{max}$ . Obviously, the lap time has achieved the optimal time. In order to examine the optimality of the obtained lap time, we set the optimal time is

$$T_{op} = \frac{L(r + \lambda) \left( (k')^2 + (r + \lambda)^2 \right)^{\frac{1}{2}}}{rv_{max}}, \quad (26)$$

when  $x_{th}/k_p$  is set to its minimum. As shown in Fig. 6,  $T$  and  $T_{op}$  are closer when  $r$  increases, and  $T/T_{op} \approx 1$  is insensitive to the parameters of the proposed spatial ILC, namely  $x_{th}, k_p$ .

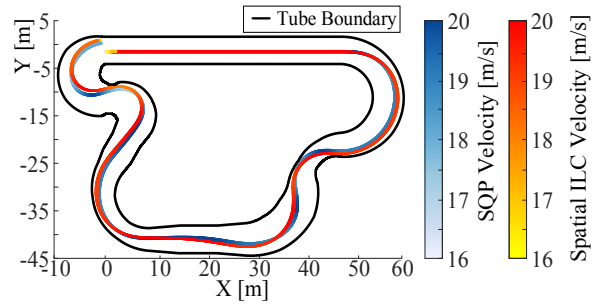
#### IV. SIMULATIONS AND EXPERIMENTS

In this section, simulations and experiments demonstrate that the proposed spatial ILC is model-free and has characteristics of online training, low computation, and optimization. Video is available at <https://youtu.be/qGTPGCLu2UQ> and <https://rflly.buaa.edu.cn>.

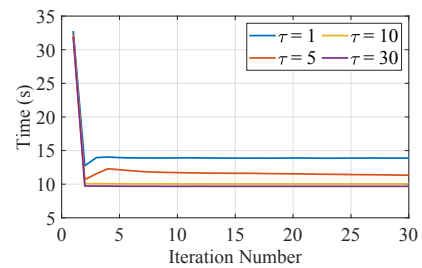
##### A. Comparison with an Optimization Method

For model (5), different values  $\tau$  for drones indicate different flight ability, which is unavailable to the proposed spatial ILC during the following simulations. For example, on the same virtual tube track shown in Fig. 7, the sequential quadratic programming (SQP) algorithm, a common optimization method, is used to solve the time-optimal problem, and its training time and lap time are compared with those obtained from the proposed spatial ILC.

As observed from Table 1, for different maneuvering parameters  $\tau$ , the lap time obtained from the proposed spatial ILC is very close to those obtained using the optimization method. But the training time of the proposed spatial ILC is only about 0.5%~1.1% of the SQP with 1500 path points. The *Optimality Tolerance* is set to  $1e^{-5}$  to judge the terminates.



(a) The comparison result using  $\tau = 5$ .



(b) The lap time varied with training iterations.

Fig. 7. The detailed simulation results of *Section V. A*.

The final pace along the path by the two methods are shown in Fig. 7 (a). As shown in Fig. 7 (b), the lap time decreases rapidly and finally reaches convergence through learning. The changing trend of the two methods is the same at the same position. This implies that the search direction of the proposed spatial ILC tends to be an optimal set.

TABLE I  
LAP TIME AND TRAINING TIME OF DIFFERENT ALGORITHM

$\tau$	Lap Time			Training Time		
	SQP	Spatial ILC	Spatial ILC/SQP	SQP	Spatial ILC	Spatial ILC/SQP
1	13.64s	<b>13.88s</b>	101.8%	15576.6s	<b>131.1s</b>	0.8%
5	11.26s	<b>11.34s</b>	100.7%	13166.1s	<b>147.3s</b>	1.1%
10	10.12s	<b>10.14s</b>	100.2%	15484.4s	<b>135.7s</b>	0.9%
30	9.82s	<b>9.85s</b>	100.3%	21159.5s	<b>110.9s</b>	0.5%

### B. Comparison in Race Competition

The proposed spatial ILC is compared with [6] in the same racing environment to verify the proposed approach's model-free and online fast iterative features. The racing environment, namely Soccer Field, is from [4] shown as Fig. 8. The benchmark includes the official baseline [4] Move-On-Spline API (MOS-ADRL) and the algorithms mentioned in [6], Hamilton Jacobi Bellman (HJB), Hamilton Jacobi Bellman-Reinforcement Learning (HJB-RL, the best algorithm in this paper), Move-on-Spline (MOS, MOS-ADRL's method improvement), and Supervised Learning (SL).

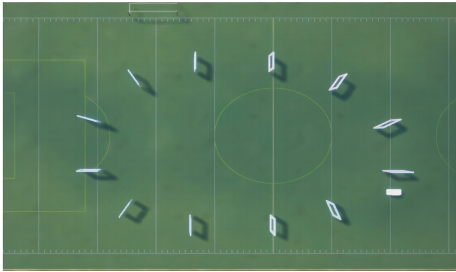


Fig. 8. The racing scene is shown in AirSim. It is a soccer field with 12 gates in a whole race.

Before performing the spatial ILC, a virtual tube must be established first. However, since the path optimization is not within the scope of this paper, a simple path which is generated by connecting several points is adopted. As shown in Table. II, the spatial ILC performs better in both the shortest lap time and the average lap time in the whole race. Besides, unlike the offline training of the commonly-used learning methods, the proposed spatial ILC can obtain the results within only 20 iterations (the drone races a dozen laps) through iterative online training. The advantage of online training is that the controller is obtained directly without modeling.

TABLE II  
LAP TIME OF DIFFERENT ALGORITHM

Algorithm	Shortest	Average
HJB	39.67s	68.10s
HJB-RL( $\alpha = 5e^{-5}$ )	over 28.99s	30.36s
HJB-RL( $\alpha = 1e^{-5}$ )	28.99s	34.89s
SL	30.13s	36.14s
MOS	47.10s	47.88s
MOS-ADRL	over 50s	58.02s
<b>Spatial ILC</b>	<b>24.02s</b>	<b>24.32s</b>

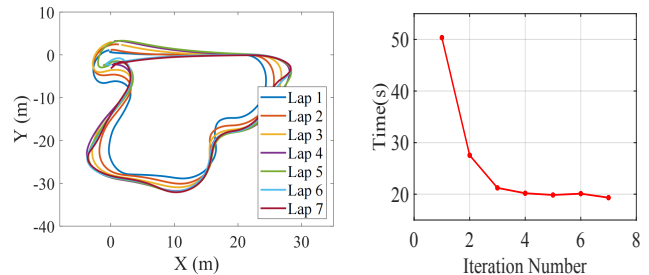
### C. Real-world Flight Experiment

The proposed scheme is implemented in a 30m\*30m outdoor area on a real quadcopter without modeling. The online training iteration number is set to 7; the initial speed is set to 2m/s and the saturation speed  $v_{\max} = 8\text{m/s}$  to keep safe. As



Fig. 9. Testing on a real quadcopter.

shown in Fig 10, the training path shows convergence path of the proposed scheme. The lap time shown gradually reaches convergence. Finally, the quadcopter reaches the minimum 20.19s only in 4 iterations from 50.35s in the 1st iteration. The total online training time is 178.71s.



(a) The GPS log from iteration 1 to 7. (b) The lap time varied with training iterations in real flight.

Fig. 10. The real flight log record.

## V. CONCLUSIONS

This paper proposes a time-optimal spatial ILC approach for drone racing competitions where the scene is modeled as a virtual tube as a constraint. This method is iterative and model-free, inspired by human racing drivers' acceleration and deceleration control. Comparison simulations and a real experiment are performed to show the effectiveness of the proposed method. Some conclusions are drawn: 1) the proposed approach can achieve near-optimal results with significantly lower computation, about 0.5%-1.1% of the optimal control; 2) this approach surpasses some state-of-the-art methods in the racing time with 17% improvement on the shortest time and 20% on average time in a benchmark drone racing platform; 3) the proposed approach can quickly converge to a stable state in real flight without modeling. In the future, more relationships between the proposed learning method and optimal control, and drone racing in 3D space are worth studying.

## REFERENCES

- [1] Q. Quan, *Introduction to Multicopter Design and Control*, Singapore: Springer, 2017.
- [2] E. Kaufmann, M. Gehrig, P. Foehn and R. Ranftl. "Beauty and the beast: Optimal methods meet learning for drone racing," *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp.690-696.
- [3] P. Foehn, D. Brescianini, E Kaufmann, et al, "Alphapilot: Autonomous drone racing," *Autonomous Robots*, 2022, vol.46, pp. 307-320.
- [4] R. Madaan, N. Gyde, S. Vemprala, et al, "Airsim drone racing lab", *NeurIPS 2019 Competition and Demonstration Track*, PMLR, 2022, pp.177-191.
- [5] M. W. Mueller, M. Hehn, R. D'Andrea. "A computationally efficient algorithm for state-to-state quadcopter trajectory generation and feasibility verification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp.3480-3486.
- [6] K. Nagami, M Schwager, "HJB-RL: Initializing reinforcement learning with optimal control policies applied to autonomous drone racing," in *Robotics: Science and Systems*. 2021, pp.1-9.
- [7] Y. Song, M. Steinweg, E. Kaufmann and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp.1205-1212.
- [8] D. J. Webb, J. Van Den Berg, "Kinodynamic RRT\*: Asymptotically optimal motion planning for robots with linear dynamics," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp.5054-5061.
- [9] C. R. Hargraves, S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, 1987, vol.10, pp.338-342.
- [10] P. Foehn, A. Romero, D Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Science Robotics*, 2021, vol.6, eabh1221.
- [11] X. Dai, C. Ke, Q. Quan and K.-Y. Cai, "RFlySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations," *Aerospace Science and Technology*, 2021, vol.114, pp.106727.
- [12] D. A. Bristow, M. Tharayil, A. G. Alleyne. "A survey of iterative learning control," *IEEE Control Systems Magazine*, 2006 vol.26, pp.96-114.
- [13] Y. Wang, F. Gao, F. J. Doyle III. "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, 2009, vol.19, pp.1589-1600.
- [14] J.-X. Xu, "A survey on iterative learning control for nonlinear systems," *International Journal of Control*, 2011, vol.84, pp.1275-1294.
- [15] D. Shen, Y. Wang. "Survey on stochastic iterative learning control," *Journal of Process Control*, 2014, vol.24, pp.64-77.
- [16] S. Riaz, L. Hui, M. S. Aldemir and F. Afzal, "A future concern of iterative learning control: A survey," *Journal of Statistics and Management Systems*, 2021, vol.24, pp.1301-1322.
- [17] S. Shah, D. Dey, C. Lovett and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *Field and Service Robotics*, Springer, Cham, 2018, pp.621-635.
- [18] J.-X Xu, D. Huang. "Spatial periodic adaptive control for rotary machine systems," *IEEE Transactions on Automatic Control*, 2008, vol.53, pp.2402-2408.
- [19] Z. Li, C. Yin, H. Ji and Z. Hou, "Constrained spatial adaptive iterative learning control for trajectory tracking of high speed train," *IEEE Transactions on Intelligent Transportation Systems*, 2021. DOI: 10.1109/TITS.2021.3106653.
- [20] Q. Quan, R. Fu, M. Li, et al, "Practical distributed control for VTOL UAVs to pass a virtual tube," *IEEE Transactions on Intelligent Vehicles*, 2021. DOI: 10.1109/TIV.2021.3123110.
- [21] Y. Gao, C. Bai, L. Zhang and Q. Quan. "Multi-UAV Cooperative Target Encirclement within an Annular Virtual Tube," *Aerospace Science and Technology*, 2022, pp.107800.
- [22] P. Mao and Q. Quan, "Making robotics swarm flow more smoothly: a regular virtual tube model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp.4498-4504.
- [23] S. Barendswaard, L. Van Breugel, B. Schelfaut, et al, "Effect of velocity and curve radius on driver steering behaviour before curve entry," *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp.3866-3871.
- [24] J. L. Campbell, M. G. Lichty, J. L. Brown, et al, *NCHRP Report 600: Human Factors Guidelines for Road Systems*. Transportation Research Board of the National Academies, Washington, DC, 2012.
- [25] Q. Quan, K.-Y. Cai, "Saturated repetitive control for a class of nonlinear systems: A contraction mapping method," *Systems & Control Letters*, 2018, vol.122, pp.93-100.
- [26] M. Athans, P. L. Falb, *Optimal control: an introduction to the theory and its applications*, Courier Corporation, 2013.