Bioinspiration & Biomimetics

PAPER

Optimal trajectory generation for time-to-contact based aerial robotic perching

IOP Publishing Bioinspir. Biomim. 14 (2019) 016008 https://doi.org/10.1088/1748-3190/aaeb13

Bioinspiration & Biomimetics

RECEIVED
28 December 2017

REVISED
10 October 2018

ACCEPTED FOR PUBLICATION
24 October 2018

PUBLISHED
27 November 2018

PAPER

Optimal trajectory generation for time-to-contact based aerial robotic perching

Haijie Zhang[1], Bo Cheng[2] and Jianguo Zhao[1]

[1] Department of Mechanical Engineering, Colorado State University, Fort Collins, CO 80523, United States of America
[2] Department of Mechanical and Nuclear Engineering, Pennsylvania State University, University Park, PA 16802, United States of America

E-mail: Jianguo.Zhao@colostate.edu

Keywords: aerial robot perching, optimal trajectory planning, time-to-contact

Abstract

Many biological organisms (e.g. insects, birds, and mammals) rely on the perception of an informational variable called time-to-contact (TTC) to control their motion for various tasks such as avoiding obstacles, landing, or interception. TTC, defined as the required time to contact an object if the current velocity is maintained, has been recently leveraged for robot motion control in various tasks. However, most existing robotic applications of TTC simply control the TTC to be constant or constantly decreasing, without fully exploring the applicability for TTC. In this paper, we propose two-stage TTC based strategies and apply them to aerial robotic perching. With the proposed strategies, we can generate reference trajectories for TTC to realize the non-zero contact velocity required by perching, which is impossible for constant or constantly decreasing TTC strategy, but of critical importance for robust perching performance. We conduct simulations to verify the superiority of the proposed strategies in terms of shorter time for perching and satisfying more constraints. Further, with properly designed controllers, we perform experiments on a palm-size quadcopter to track the planned reference trajectories and realize aerial robotic perching. The research presented in this paper can be readily applied to the control of flying robot for perching with visual feedback, and can inspire more alternative forms of TTC based planning and control for robotic applications.

1. Introduction

Although aerial robots are widely used for various civilian and military applications (e.g. surveillance, search, and monitoring, etc.), they suffer universally from limited airborne time (less than one hour) due to the low aerodynamic efficiency and high energy consumption [1]. To address this problem, perching onto objects (e.g. walls, power lines, or ceilings) will significantly extend aerial robots' functioning time as they can save or even harvest energy after perching, while also maintaining a desired altitude and orientation for surveillance or monitoring [2]. Successful perching for aerial robots, however, is quite challenging as it requires not only intelligent mechanical mechanisms to robustly engage the robot to the perching objects but also fast and accurate estimation, planning, and control of the robot motion so that the robot can progressively reduce its speed and adjust its orientation to perch on the objects with a desired velocity and orientation.

In recent years, researchers have investigated perching capabilities for aerial robots from both the mechanical and control aspects. A detailed review can be found in [2], and here we will only review some representative work. For mechanical investigations, the focus is on how to design robust perching mechanisms to ensure successful perching. Doyle *et al* developed an integrated, compliant, and underactuated gripping foot as well as a collapsing leg mechanism to enable a quadcopter to passively perch on the surface with moderate disturbances [3]. Daler *et al* designed a new perching mechanism based on a compliant deployable pad and a passive self-alignment system. With this mechanism, active control during final touch down is not needed [4]. Pope *et al* designed a mechanism to make a quadcopter fly, perch passively onto outdoor surfaces, climb, and take off again [5]. Graule *et al*

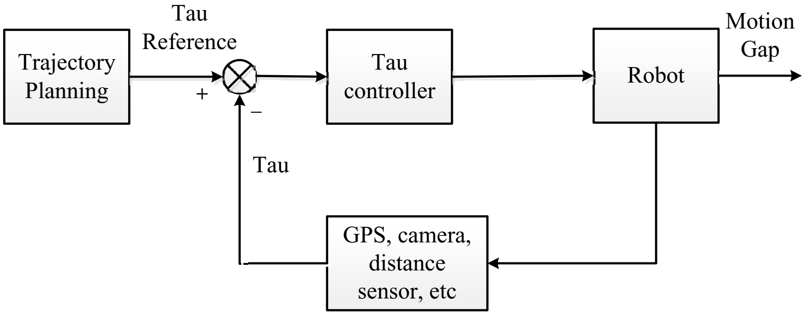

Figure 1. General idea for robot perching based on tau theory. This framework contains three major problems: 1) estimation problem: accurately estimate τ using sensors; 2) control problem: design a τ controller; 3) planning problem: plan the reference τ trajectory in τ space.

utilized controllable electrostatic adhesion to make a robotic insect perch and take off from surfaces made of various materials [6]. Kovac *et al* designed a 4.6 g perching mechanism which allows UAVs to perch on vertical surface of natural and man-made materials [7]. For investigations from the control and planning aspect, researchers have focused on how to generate and track flight trajectories for perching. Moore *et al* utilized linear quadratic regulator trees to plan and track trajectory for fixed-wing aircrafts to perch on power lines [8]. Mellinger *et al* designed a trajectory for quadcopter aggressive maneuvers to realize flights through narrow gaps and perching on inverted surfaces [9]. They also controlled quadcopters to perch on inclined surfaces with a downward-facing gripper [10]. Mohta *et al* leveraged visual servoing with two known points on the target surface to achieve perching using feedback from a monocular camera and an inertial measurement unit (IMU) [11]. In [12], a laser sensor is used to detect the perching initiation distance, and a pitch up process is used to assist the deceleration. Further, in [13], they applied thrust after pitch up to reduce the sensing requirement for triggering the perching process.

However, almost all of the existing investigations on perching are position-based, i.e. they require the precise position feedback either through global positioning systems (GPS) or motion tracking system, making them unsuitable for autonomous perching in situations where positions cannot be obtained (e.g. GPS-denied environments). In this paper, we leverage the concept of time-to-contact (TTC), defined as the projected time to contact a surface with the current velocity, for the planning and control of aerial perching. Compared with position-based perching methods, TTC-based methods can utilize simple but effective strategies to achieve autonomous perching without complex planning and control. Further, it can be potentially realized with onboard lightweight vision sensors to estimate TTC, which is ideal for miniature aerial robots as they cannot carry heavy sensors (e.g. LIDAR).

TTC is a part of the more general tau theory originated from Gibson's research on the relationship between animals' visual information and their locomotion [14]. Based on Gibson's work, Lee first proposed the concept of TTC by pointing out that rather than distance or speed, drivers leverage TTC to determine when to accelerate or decelerate to drive safely [15]. Later, Lee introduced tau coupling [16] to guide motions in three-dimensional space simultaneously. The basic concepts of tau theory are as follows [17, 18].

- A motion gap, denoted as $X(t)$, is the changing gap with respect to time t between the current state and the goal state. Motion gaps can be distance, force, angle, etc.
- Tau of a motion gap is the time to close this gap at its current closure rate $\dot{X}(t)$: $\tau = X(t)/\dot{X}(t)$. In the case with the gap being the distance, tau is the same to TTC. *In this paper, we will use tau or TTC interchangeably since only the distance as a motion gap will be considered.*
- Tau-dot is the time derivative of tau. By maintaining a constant tau-dot, animals and insects can land or perch on surfaces with a full stop.

TTC or tau has been widely found in controlling the motion for humans, animals, and insects. By estimating TTC from visual feedback, drivers can determine how to avoid collisions [15]. Bees keep a constant rate of image expansion (equivalent to TTC) to land on various vertical surfaces [19]. Pigeons are discovered to adopt TTC to safely perch on branches [20]. Seabirds can leverage TTC to adjust the timing to close wings before diving into water for fish [21].

With biological inspirations, tau theory has also been recently applied to various robotic applications such as avoiding obstacles or landing on ground [22]. For existing tau-theory based planning and control, the general architecture is illustrated in figure 1 and can be described as follows. First, a reference trajectory for tau or TTC is planned off-line based on the desired task (e.g. perching, docking, or landing). Then by comparing the reference tau with the estimated tau, which can be obtained from image feedback of cameras, GPS, or distance sensors, a controller is designed to control the robot's motion so that the reference tau can be tracked to accomplish the desired task.

Substantial work has been performed to address the estimation and control problem shown in figure 1. Mobile robot docking task is realized by estimating the TTC using the divergence of optic flow when the focus of expansion effects are considered [23]. Similarly, TTC is estimated from optical flow divergence on a drone and used for autonomous landing [24]. Robot chasing is also realized by controlling TTC [25]. Izzo and Croon propose different TTC-based controllers for landing [26]. Recently, Kendoul has conducted comprehensive research by proposing several TTC controllers and accomplished several tasks such as docking, landing, and navigation using a quadcopter based on tau theory [22]. We have also leveraged an image based featureless estimation method to calculate TTC on a quadcopter in landing scenarios [27]. Further, we have also applied the featureless TTC estimation method to realize mobile robot docking [28, 29]. Recently, based on TTC, several papers propose the distance and velocity estimation algorithm using a monocular camera for flying robots [30–33]. In [30, 33], with the estimated TTC or divergence as the extended Kalman filter input, the estimation can converge to the ground truth value fast.

Although extensive research has been carried out for estimation and control, the trajectory planning problem in tau theory is underexplored. Indeed, most of existing research simply utilizes the constant tau dot strategy to generate the reference trajectory of tau [22]. However, directly applying constant tau dot strategy to perching can only control the contact velocity to be zero [22, 23], which is not always desired for robotic perching. In fact, most perching mechanisms require a substantial velocity in the direction perpendicular to the perching object to ensure that gripping mechanisms can robustly attach to the object [10]. For example, the mechanism in [10] requires a minimum normal velocity of 0.8 m s^{-1} for successful perching. In such cases with non-zero contact velocity, existing tau theory is unlikely to work. To address this problem, we have extended the tau theory by proposing a two-stage strategy to control the contact speed to a specific value and validated the theory using a mobile robot [34]. However, the strategy can only satisfy two constraints (contact velocity and maximum deceleration), making robust perching not feasible as they normally require several constraints to be satisfied [10, 35]. In this paper, we propose a new planning strategy for TTC-based robotic perching and validate the proposed strategy using a palm-size quadcopter.

Our major contribution in this paper is to leverage TTC or tau to accomplish robotic perching, which requires simpler planning and control compared with position-based approaches. Specifically, there are two contributions. First, we propose a new two-stage planning method to generate the reference trajectory for TTC. Such a method can generate optimal trajectories satisfying multiple constraints required for robust perching. Second, we validate the proposed planning strategy using a palm-size quadcopter by mapping the planned trajectory in tau space into the commands acceptable by the quadcopter. Note that in this paper, although TTC is estimated from motion tracking system in the experiments, we plan to integrate our vision-based estimation algorithm [28, 29] with the proposed strategy for vision-based perching using onboard cameras in the future using a larger quadcopter currently under developments.

The rest of this paper is organized as follows. Section 2 describes existing planning strategies for tau, including the widely used constant tau dot strategy (CTDS) [22] and our recently proposed constant tau dot based two-stage strategy (CTDTS) [34]. The newly proposed two-stage strategy will also be discussed in detail. Based on the planned reference, section 3 presents a controller design to track the reference for perching with a quadcopter. To verify the performance of the proposed planning strategies and control methods, section 4 discusses and compares the simulation and experiment results.

2. Tau based trajectory generation

In this section, we first introduce two trajectory generation methods for tau (CTDS and CTDTS), discuss the need for new strategies for robust aerial perching, and detail the new inverse of polynomial based two-stage strategy (IPTS).

As shown in figure 2, the perching problem aims to control the motion of an aerial robot flying towards a surface to contact the surface with a perching speed in a specific range so that the gripping mechanism can robustly attach to the surface [36]. Generally, the orientations should also be adjusted appropriately before the final touchdown; however, in this paper, we will not consider the attitude maneuvering as our main goal is to investigate the control of contact velocity through the use of tau. If we establish a coordinate frame attached to the surface with X along the perching direction, then the gap $X(t) < 0$ and velocity $\dot{X}(t) > 0$. The desired contact velocity is $V_c \in (V_l, V_u)$, where V_l and V_u are the lower and upper bound for successful perching velocity, respectively. For different perching mechanisms, V_l and V_u vary and can be obtained experimentally. Without loss of generality, we assume the initial velocity is larger than the required contact velocity $\dot{X}(0) > V_c$ since biological organisms generally decelerate to perch [2]. Let t_c be the time when contact occurs, then the non-zero contact velocity requirement is $\dot{X}(t_c) = V_c \neq 0$.

2.1. Constant tau dot strategy (CTDS)

For braking or landing problem, the CTDS has been shown to guide animals or humans to smoothly decelerate to zero contact velocity [37, 38]. Also, the applications of CTDS in robotic field effectively

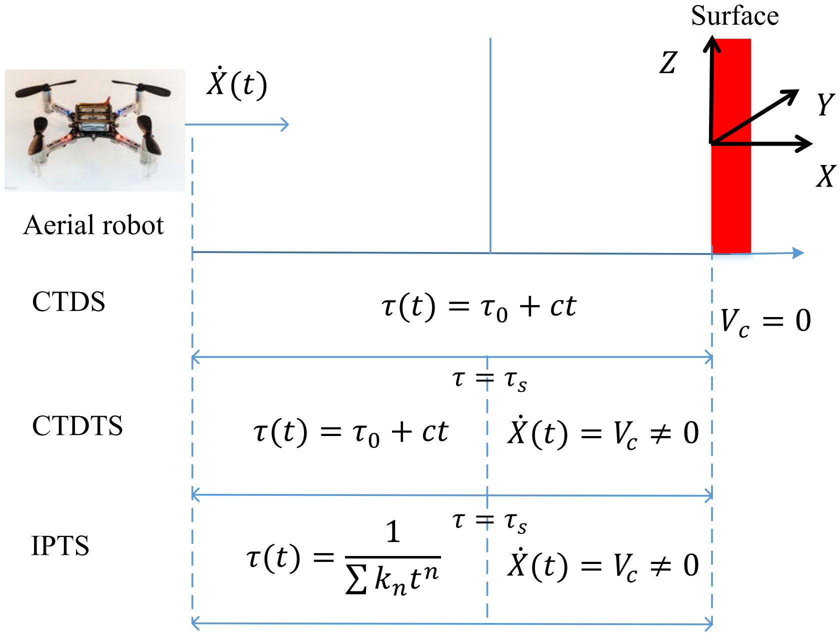

Figure 2. General idea for three tau based strategies for perching, constant tau dot strategy (CTDS), constant tau dot based two-stage strategy (CTDTS) and inverse of polynomial based two-stage strategy (IPTS).

demonstrate its potential for robot motion control [22].

To better understand the other two strategies, we briefly summarize the CTDS as follows. First, as mentioned in the introduction, TTC is defined as:

$$\tau(t) = \frac{X(t)}{\dot{X}(t)}. \tag{1}$$

For CTDS, suppose the rate of change of τ, i.e. the tau-dot, is a constant c, then we have:

$$\tau(t) = ct + \tau_0, \quad \tau_0 = \frac{X(0)}{\dot{X}(0)} < 0 \tag{2}$$

where $X(0) < 0$ is the initial distance, and $\dot{X}(0) > 0$ is the initial velocity. Combining equations (1) and (2), we can solve for $X(t)$, $\dot{X}(t)$, and $\ddot{X}(t)$:

$$X(t) = X_0 \left(1 + c\frac{\dot{X}_0}{X_0}t\right)^{1/c} \tag{3}$$

$$\dot{X}(t) = \dot{X}_0 \left(1 + c\frac{\dot{X}_0}{X_0}t\right)^{1/c-1} \tag{4}$$

$$\ddot{X}(t) = \frac{\dot{X}_0^2}{X_0}(1-c)\left(1 + c\frac{\dot{X}_0}{X_0}t\right)^{1/c-2} \tag{5}$$

where $X_0 = X(0)$ and $\dot{X}_0 = \dot{X}(0)$. Based on the above equations, we cannot realize non-zero contact velocities with c having different values. In fact, we can discuss it based on the ranges of c according to [22]:

- when $c \leqslant 0$, $X(t), \dot{X}(t), \ddot{X}(t)$ converge asymptotically to 0 for $t \to \infty$. Therefore, perching is impossible since the time cannot be infinite when the contact occurs;
- when $0 < c \leqslant 0.5$, $X(t), \dot{X}(t), \ddot{X}(t)$ become 0 at the same finite time. Although perching can be accomplished in finite time, the contact velocity will always be zero;
- when $0.5 < c < 1$, $X(t)$ and $\dot{X}(t)$ become 0 at the same finite time, but $\ddot{X}(t)$ becomes ∞. Perching is infeasible since robots cannot have infinitely large accelerations;
- when $c = 1$, a robot will move towards the surface with a constant velocity—the initial velocity. Perching is feasible in this case, but the contact velocity is fixed;
- when $c > 1$, $\dot{X} \to \infty$, $\ddot{X} \to \infty$ as $X \to 0$. Perching is again infeasible since robots cannot have infinitely large velocities and accelerations.

For the general CTDS, c is chosen to be in $(0, 0.5]$ so that the distance, velocity, and acceleration can be zero in the same finite time to realize contact with a surface for landing or docking applications [22]. However, we can see that, with $c \in (0, 0.5]$, the desired non-zero contact velocity cannot be achieved since when the contact occurs, i.e. $X(t) = 0$, we have $1 + c\frac{\dot{X}_0}{X_0}t = 0$, so $\dot{X}(t)$ and $\ddot{X}(t)$ would also be zero [22]. This means when the contact or perching occurs, both the velocity and acceleration must be zero. In this case, the perching may fail if a non-zero contact velocity is required. To address this problem, we have proposed the CTDTS [34].

2.2. Constant tau dot based two-stage strategy (CTDTS)

As shown in figure 2, CTDTS uses a constant tau dot strategy in the first stage to decelerate to the desired contact velocity V_c. The second stage initiates when tau is larger than a prescribed threshold τ_s (Note that $\tau \leqslant 0$ owing to the choice of frame setup as shown in figure 2). In the second stage, the same forward velocity is maintained at V_c until perching occurs. Note that the initiation of the second stage is different from our previous CTDTS presented in [34], where a prescribed distance is used. Using tau is better than distance as tau specifies how soon will the robot contact the surface so that the robot can initiate an attitude maneuvering if necessary. Tau-based threshold is also adopted in biological systems (e.g. flies [39] or hawks [40]) for attitude maneuvering or leg extensions before the final touchdown. In general, the magnitude of τ_s is rather small and we assume that $\tau_0 < \tau_s < 0$, where τ_0 is the initial tau.

Assume the second stage starts at time t_s with velocity V_c, then the reference trajectory for tau can be represented in two stages

$$\tau_{ref}(t) = \begin{cases} ct + \tau_0, & \text{if } 0 \leqslant t \leqslant t_s \\ \tau_s + t - t_s, & \text{if } t > t_s \end{cases}. \quad (6)$$

With a specified threshold τ_s, desired contact velocity V_c, and τ_0 depending on the initial conditions, we need to solve the constant tau dot c and the switching time t_s to obtain $\tau_{ref}(t)$. A unique solution can be found for c and t_s by using the desired tau (τ_s) and velocity (V_c) at the stage transition: $\tau_{ref}(t_s) = \tau_s$ and $\dot{X}(t_s) = V_c$. In fact, the solution can be found as:

$$c = \frac{\log(\frac{\tau_s}{\tau_0})}{\log(\frac{V_c}{\dot{X}_0}) + \log(\frac{\tau_s}{\tau_0})} \quad (7)$$

$$t_s = \frac{\tau_s - \tau_0}{c}. \quad (8)$$

With $\dot{X}_0 > V_c > 0$ and $\tau_0 < \tau_s < 0$, we can easily show that c will be in $(0, 1)$. However, the unique solution for c and t_s should satisfy the constraints for the acceleration capability of a flying robot which is limited by its motors. Note that for the first stage with a constant tau dot, the robot always decreases its speed when $c \in (0, 1)$, i.e. $\ddot{X}(t)$ is always negative. For the deceleration of the whole process, after analysis, we know that when $t \in [0, t_s]$ [34]:

- if $0 < c \leqslant 0.5$, $\ddot{X}(t)$ will monotonically increase. Since $\ddot{X}_0 < 0$, the maximum deceleration should be achieved at $t = 0$;
- if $0.5 < c < 1$, $\ddot{X}(t)$ will monotonically decrease. Since $\ddot{X}_0 < 0$, the maximum deceleration should be achieved at $t = t_s$.

Therefore, the solution of c should satisfy the following deceleration constraint:

$$\ddot{X}(0) = (1-c)\frac{\dot{X}_0^2}{X_0} < a_{max}, \quad \text{if } 0 < c \leqslant 0.5 \quad (9)$$

$$\ddot{X}(t_s) = (1-c)\frac{\dot{X}_0^2}{X_0}\left(\frac{V_c}{\dot{X}_0}\right)^{\frac{1-2c}{1-c}} < a_{max}, \quad \text{if } 0.5 < c < 1 \quad (10)$$

where a_{max} is the maximum acceleration/deceleration of the robot. With such constraints, we can see that a major limitation for CTDTS is that the unique solution c may not satisfy the constraints specified in equations (9) or (10), leading to no feasible solution for the reference tau. Therefore, new strategies should be developed to address this issue.

To compare the time required for whole perching process with the proposed planning strategy to be discussed in the next subsection, we obtain the total time for the two stages as [34]:

$$t = \frac{X_0}{c\dot{X}_0}\left[\left(\frac{V_c}{\dot{X}_0}\right)^{\frac{c}{1-c}} - 1\right] - \left(\frac{X_0}{\dot{X}_0}\right)^{\frac{1}{1-c}}. \quad (11)$$

2.3. Inverse of polynomial based two-stage strategy (IPTS)

The shortcomings of CTDTS can be addressed by proposing new strategies with more parameters in the first stage while keeping the second stage the same. In this subsection, we discuss a new mathematical form of tau reference in the first stage based on the inverse of a polynomial:

$$\tau(t) = \frac{1}{\sum_{i=0}^{n} k_n t^n} \quad (12)$$

where $k_i \quad (i = 0, 1, \ldots, n)$ are parameters to be determined. $k_0 = 1/\tau_0$ can be determined by initial conditions, while $k_1, k_2, \ldots, k_n$ can be determined from multiple constraints or optimizations for minimizing time, control effort, or energy, etc. As shown in figure 2, the second stage for IPTS is still constant velocity after tau reaches a specified threshold τ_s.

With the proposed trajectory in equation (12), the distance, velocity, and acceleration in the first stage can be solved analytically from integration:

$$X(t) = X_0 \exp\left(\sum_{i=0}^{n} \frac{k_i}{i+1} t^{i+1}\right) \quad (13)$$

$$\dot{X}(t) = X_0 \sum_{i=0}^{n} k_i t^i \exp\left(\sum_{i=0}^{n} \frac{k_i}{i+1} t^{i+1}\right) \quad (14)$$

$$\ddot{X}(t) = X_0 \left[\sum_{i=1}^{n} i k_i t^{i-1} + \left(\sum_{i=0}^{n} k_i t^i\right)^2\right] \exp\left(\sum_{i=0}^{n} \frac{k_i}{i+1} t^{i+1}\right). \quad (15)$$

The inverse of polynomial approach has three major advantages compared with the constant tau-dot. First, with a larger n, more constraints can be satisfied since we need to solve for more parameters. Therefore, the limitation for CTDTS can be eliminated. Second, as will be shown in simulations and experi-

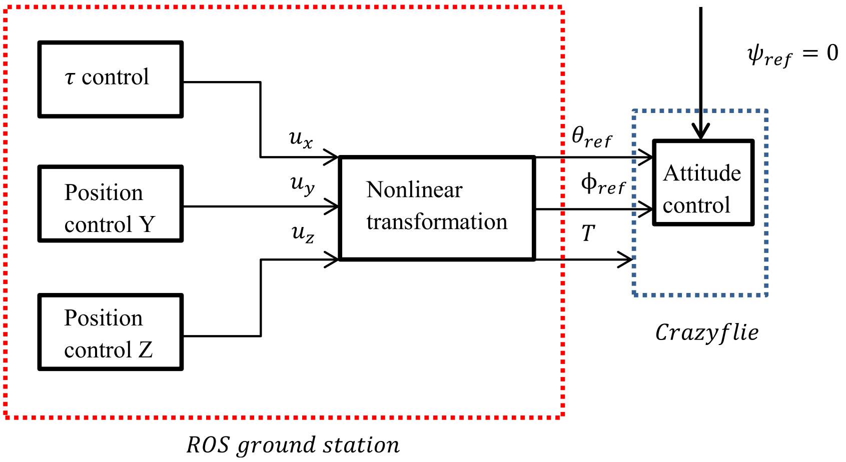

Figure 3. General control diagram for experiments. Three controllers are combined to generate the control command for Crazyflie.

ments, IPTS can generate a reference trajectory with a shorter perching time, since the robot can accelerate first and then decelerate to the desired velocity. On the contrary, as discussed in section 2.2, CTDTS can only decelerate to the desired velocity. Third, the resulting reference trajectories for both CTDS and CTDTS rely on the initial conditions (distance and velocity) as can be seen from equations (3)–(5) and (8). However, as we will numerically show later, a fixed set of parameters for IPTS will work if the initial conditions are in a given range, greatly facilitating implementations since estimating distance and velocity from tau is a difficult problem [30–33].

Since the number of parameters n in IPTS is flexible, we generally cannot obtain a unique solution for them. In this case, we will take the minimum time optimization as an example to obtain a unique solution and realize perching with the non-zero contact velocity requirement. For this optimization, several constraints are set: (1) Similar to CTDTS, the second stage initiates at time t_s while the tau reaches a specified threshold τ_s. (2) The velocity V_c at time t_s, i.e. the contact velocity, should be in a reasonable range (V_l, V_u). (3) The velocity for the whole process should be less than the maximum velocity V_{max}. (4) The acceleration/deceleration should be less than the maximum value limited by the capabilities of the motors. With such constraints, if we want to minimize the total time t for perching, the optimization can be formulated as:

$$\min_{k_1,k_2,\ldots,k_n} \quad t = f(k_1, k_2 \ldots k_n) \tag{16a}$$

$$\text{subject to} \quad \tau(t_s) = \tau_s \tag{16b}$$

$$V_l \leqslant \dot{X}(t_s) \leqslant V_u \tag{16c}$$

$$\dot{X}(t) \leqslant V_{max} \quad \text{for } 0 \leqslant t \leqslant t_s \tag{16d}$$

$$\left|\ddot{X}(t)\right| \leqslant a_{max} \quad \text{for } 0 \leqslant t \leqslant t_s \tag{16e}$$

where $t = t_s + |\tau_s|$ is the total time for the two stages with t_s the time for the first stage, which can be numerically solved from $\tau(t_s) = 1/(\sum_{i=0}^{n} k_n t^n) = \tau_s$. It should be noted that if we want to minimize the total perching time as formulated in equation (16), then the resulting parameters k_i will uniquely depend on the initial conditions, i.e. given a set of initial conditions, we can solve for a set of k_i. However, if we do not minimize the time, then it is possible that we can find a set of k_i that will only loosely depend on the initial conditions, i.e. as long as the initial conditions are in a range, a fixed set of k_i will make sure the perching constraints are satisfied.

3. Tau controller for aerial robots

We will verify the proposed trajectory generation methods in tau space using a palm-size and open source quadcopter (Crazyflie 2.0, Bitcraze). To this end, we need to design a controller to control the motion of the quadcopter to track the planned reference trajectories for successful perching. The modeling and control for quadcopters have been investigated intensively in literature [9, 22, 41–45]. In general, the control inputs for quadcopters are a combination of a thrust force and a torque vector, of which the directions are along or around the axis of a body frame attached to a quadcopter.

The Crazyflie quadcopter has an onboard and pre-tuned attitude controller to stabilize the orientations around a reference orientation $[\theta_{ref}, \phi_{ref}, \psi_{ref}]$ with θ, ϕ, and ψ the roll, pitch, yaw angle, respectively. The attitude controller is implemented with the onboard IMU and the autopilot. In order to control the robot to track the desired reference tau, we need to generate $[\theta_{ref}, \phi_{ref}, \psi_{ref}]$ and the thrust force T. ψ_{ref} is set to be zero which ensures the Crazyflie always faces the perching surface so that the perching mechanism can work properly.

As shown in figure 3, θ_{ref}, ϕ_{ref}, and T are obtained as follows. First, we implement a tau controller in the desired perching direction X.

$$u_x = k_{px}(1 - \frac{\tau_{ref}}{\tau}) + k_{ix}\int(1 - \frac{\tau_{ref}}{\tau}) + k_{dx}\frac{d}{dt}(1 - \frac{\tau_{ref}}{\tau}). \tag{17}$$

In this controller, instead of directly using $\tau_{ref} - \tau$ as the error item, we leverage $1 - \tau_{ref}/\tau$ since such an error item has shown to have better performance [22]. Theoretically, we only need the tau controller in equation (17) to control the motion along perching direction (in the world frame shown in figure 2, it is along the X direction). However, for our experiments later, we need to control the motion in Y and Z direction since the field of view (FOV) of our motion tracking system is limited. If the motion along Y and Z direction is not controlled, then the robot may fly outside the FOV. The position control in Y and Z is implemented as a PID controller:

$$\begin{cases} u_y = k_{py}(y_{ref} - y) + k_{iy}\int(y_{ref} - y) + k_{dy}\frac{d}{dt}(y_{ref} - y) \\ u_z = k_{pz}(z_{ref} - z) + k_{iz}\int(z_{ref} - z) + k_{dz}\frac{d}{dt}(z_{ref} - z) \end{cases} \tag{18}$$

where y_{ref} and z_{ref} are set to be zero. With the computed (u_x, u_y, u_z) expressed in world frame, we can obtain θ_{ref}, ϕ_{ref}, and T by mapping (u_x, u_y, u_z) into the body frame using the following nonlinear transformation [41, 46]:

$$\begin{cases} \phi_{ref} = \sigma_\phi[\arcsin(\frac{u_x S_{\psi_{ref}} - u_y C_{\psi_{ref}}}{\sqrt{{u_x}^2 + {u_y}^2 + (u_z + g)^2}})] \\ \theta_{ref} = \sigma_\theta[\arctan(\frac{u_x C_{\psi_{ref}} + u_y S_{\psi_{ref}}}{u_z + g})] \\ T = \sigma_T[m(u_x(S_\theta C_{\psi_{ref}} C_\phi + S_{\psi_{ref}} S_\phi) + u_y(S_\theta S_{\psi_{ref}} C_\phi - C_{\psi_{ref}} S_\phi) + (u_z + g)C_\theta C_\phi)] \end{cases} \tag{19}$$

where m and g are the mass of Crazyflie and gravitational acceleration, respectively. S and C corresponding to sin and cos respectively, and $\sigma()$ is a saturation function to ensure the computed roll, pitch and thrust are in a reasonable range. We choose the nonlinear transformation in equation (19) due to its simplicity compared with other similar transformations [47, 48].

To gain better control performance, we adopt gain scheduling in τ controller for k_{px} [28]:

$$k_{px} = \begin{cases} k_{px1} & \text{for } \frac{\tau_{ref}}{\tau} > 1 \\ k_{px2} & \text{for } \frac{\tau_{ref}}{\tau} \leqslant 1 \end{cases} \tag{20}$$

where $k_{px1} > k_{px2} > 0$. As explained in [28], the advantage of using different proportional gains is as follows. If $\tau_{ref}/\tau > 1$, we can get $\dot{X} > X/\tau_{ref}$, we need to decelerate to make $\dot{X} = X/\tau_{ref}$ to achieve $\tau_{ref}/\tau = 1$. However, the absolute value of X is also decreasing as the robot flying towards the perching surface. Therefore, we need larger decelerations compared with the case when $\tau_{ref}/\tau \leqslant 1$. We set $k_{px1} = 2.5k_{px2}$ in our experiments, which will be discussed in the next section.

4. Simulation and experiment results

To validate the effectiveness of the proposed planning and control algorithms, we perform both simulation and experiments to show that both CTDTS and IPTS can be leveraged for perching with a non-zero contact velocity with the IPTS being able to satisfy more constraints with a shorter perching time. We also demonstrate through simulations that IPTS will work when initial conditions are not exactly known.

4.1. Simulation results

For simulation, we only simulate the trajectory planning for tau to compare the performance of different strategies discussed in section 2, leaving the controller implementation to experiments. The initial conditions and constraints are selected based on the experimental setup to be discussed in the next subsection. Based on the motion tracking system's field of view, the initial conditions are chosen as $X_0 = -3$ m and $\dot{X}_0 = 1.5$ m s^{-1}. Based on the perching mechanism (needle) and the perching surface (foam board), the bounds for the contact velocity are set as $V_u = 1$ m s^{-1}, $V_l = 0.7$ m s^{-1}. Considering the delay of wireless communication and capabilities of Crazyflie, the maximum velocity and acceleration/deceleration are selected as $V_{max} = 2.5$ m s^{-1} and $|a_{max}| = 1.4$ m s^{-2}, respectively.

For the CTDS, in order to accomplish perching in finite time, $c \in (0, 0.5]$. Since the larger the tau dot is, the faster the perching would be completed. Therefore, we set the tau dot to be $c = 0.5$. With such a selection and the initial conditions, the tau reference for the whole perching process is:

$$\tau_{ref_{\mathrm{CTDS}}}(t) = 0.5t - 2. \tag{21}$$

With such a tau reference, we simulate the CTDS and plot the distance, velocity, acceleration and TTC with the blue lines in figures 4(a), (c), (e) and (g). As shown in the figure, it takes 4 s to finish the perching process with *a zero contact velocity*.

For the CTDTS, based on the initial conditions and constraints, we choose the desired contact velocity $V_c = V_u = 1$ m s^{-1} so that the perching can be achieved faster. With both the desired switching tau $\tau_s = -0.5s$ and contact velocity V_c, a unique solution exists for the constant c for tau dot and time duration for the first stage t_s. Considering the initial conditions and constraints, we can solve them as $c = 0.7337$ and

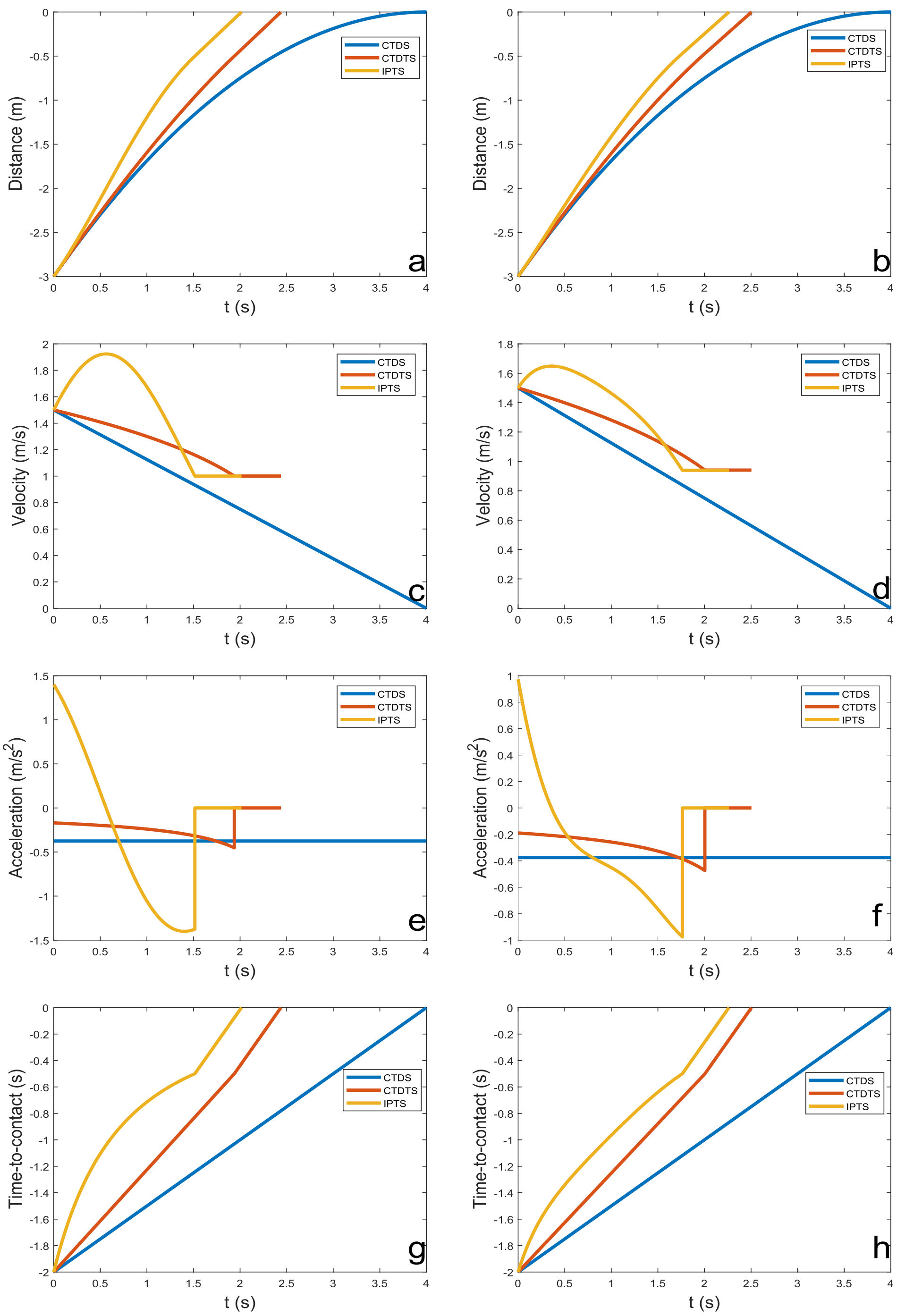

Figure 4. Simulation results with different constraints. The left column shows simulations with $V_{max} = 2.5$ m s^{-1}, $a_{max} = 1.4$ m s^{-2}. The right column shows simulations with $V_{max} = 2.0$ m s^{-1}, $a_{max} = 1.0$ m s^{-2}. From top to bottom are the distance, velocity, acceleration, and tau for CTDS, CTDTS, IPTS, respectively. The CTDTS and IPTS can both realize the nonzero contact velocity and IPTS generates the faster perching.

$t_s = 1.92$ s. The solution of c is found to satisfy the constraint of the acceleration specified in equation (10). The resulting tau reference for the whole perching process is thus:

$$\tau_{ref_{CTDTS}}(t) = \begin{cases} 0.7337t - 2, & \text{if } t < 1.92 \\ t - 2.42, & \text{if } t \geqslant 1.92 \end{cases}. \qquad (22)$$

With such a tau reference, we simulate the CTDTS and plot the distance, velocity, acceleration and TTC with the orange lines in figures 4(a), (c), (e) and (g). As shown in the figure, it takes 2.42 s to finish the perching process with a contact velocity the maximum allowable one.

For the IPTS, we use a third order polynomial ($n = 3$) to solve this problem since it is the smallest order of polynomial that can satisfy the constraints in equation (16). With the initial conditions and constraints, the optimization is performed with fmincon, a Matlab built in optimization function. To avoid being trapped at local minima, we perform the optimization with different initial conditions for k_1, k_2, and k_3. The optimization algorithm generates the final results

Table 1. Simulation results comparison.

Constraints		Strategy	Contact velocity (m s^{-1})	Perching time (s)
$V_{max} = 2.5$ m s^{-1}	$a_{max} = 1.4$ m s^{-2}	CTDS	0	4
		CTDTS	1	2.42
		IPTS	1	2.014
$V_{max} = 2.0$ m s^{-1}	$a_{max} = 1.0$ m s^{-2}	CTDS	0	4
		CTDTS	0.94	2.504
		IPTS	0.94	2.2628

Table 2. Feasible initial conditions for different IPTS strategies.

IPTS order	k	V_0 range	Z_0 range
3rd	$k_1 = -0.60, k_2 = -0.057, k_3 = 0.054$	$[1.89, 1.99]$	$[-3.88, -3.73]$
4th	$k_1 = -0.60, k_2 = 0.15, k_3 = -0.31, k_4 = 0.14$	$[1, 1.65]$	$[-2.92, -2]$
5th	$k_1 = -0.57, k_2 = 0.13, k_3 = -0.57, k_4 = 0.53, k_5 = -0.14$	$[1.4, 2.15]$	$[-3.43, -2]$

for the constants as $k_1 = -0.7166$, $k_2 = -0.1907$, $k_3 = 0.0067$. With such constants, the switching time $t_s = 1.5148$ s. As a result, the tau reference for the whole perching process is:

$$\tau_{ref_{IPTS}}(t) = \begin{cases} \frac{1}{0.0067t^3 - 0.1907t^2 - 0.7166t - 0.5}, & \text{if } t < 1.5148 \\ t - 2.014, & \text{if } t \geqslant 1.5148 \end{cases}. \tag{23}$$

With such a tau reference, we plot the distance, velocity, acceleration and TTC with the yellow lines in figures 4(a), (c), (e) and (g). Comparing the simulation results for the three different trajectories, we can conclude that IPTS can take the advantage of motor capabilities to decelerate and accelerate, while CTDTS can only allow for deceleration. Thus, the velocity of the first stage for IPTS is increased first and decreased, which induces the faster perching compared with the CTDTS. Further, CTDTS can only satisfy the acceleration constraint with a fixed contact velocity V_c, while IPTS can satisfy additional velocity constraints with a flexible contact velocity in a range (V_l, V_u). For CTDTS, the acceleration figure verifies the maximum deceleration occurs at t_s when $c > 0.5$ as shown in equation (10).

From the simulation results, we can see that even the desired contact velocity V_c for IPTS is specified in a range (V_l, V_u), the optimization will make $V_c = V_u$ to realize the fastest perching. To investigate whether V_c will be always equal to V_u or not, we change the constraints to $V_{max} = 2$ m s^{-1}, $|a_{max}| = 1$ m s^{-2}. The resulting contact velocity from the optimization in this case is 0.94 m s^{-1}. Therefore, for different constraints, the contact velocity may not be V_u, and the constraint $V_c \in (V_l, V_u)$ is necessary for the optimization of minimum time perching. In this case the tau reference for the whole perching process for IPTS is:

$$\tau_{ref_{IPTS}} = \begin{cases} \frac{1}{-0.2563t^3 + 0.295t^2 - 0.5745t - 0.5}, & \text{if } t < 1.7628 \\ t - 2.2628, & \text{if } t \geqslant 1.7628 \end{cases}. \tag{24}$$

The simulation results for all the three cases under the new constraints are shown in figures 4(b), (d), (f) and (h). The reference trajectory for CTDS and CTDTS are plotted in the figure with blue and oranges lines, respectively. Note that for the new constraints, we set $V_c = 0.94$ m s^{-1} for CTDTS to compare with the IPTS. The constant c for CTDTS can be solved as 0.7479 and the perching time is 2.504 s. The IPTS switches stage at $t_s = 1.7628$ s and perches at $t = 2.2628$ s. From the figure, we can see that the trends of the two simulations with different constraints (left and right column of figure 4) are exactly the same, but the contact velocities are different.

To better visualize the results, we have listed all the constraints and the corresponding perching time in table 1. As can be seen from the table, IPTS can achieve the best performance under the given constraints with a perching time of 2.014 s or 2.2628 s compared with 2.42 s or 2.504 s for CTDTS and 4 s for CTDS.

In addition, we also conduct several simulations to show the advantage of IPTS strategy for safe perching when the initial conditions are not exactly known. Under this situation, we do not aim to minimize the total perching time, but only to satisfy the contact velocity, maximum velocity and maximum acceleration constraints. Thus the goal of safe perching can still be fulfilled. In this case, a fixed set of coefficients k_i can be found for a given range of initial conditions. Further, the range will expand if the order of the polynomial in IPTS increases. We have tested several examples for polynomials of different orders as shown in table 2. From the simulations, we can find many sets of ks to satisfy the constraints. For a 3rd order polynomial, one example set is $k_1 = -0.60$, $k_2 = -0.057$, $k_3 = 0.054$. With this set of k, as long as the initial condition is $Z_0 \in [-3.88, -3.73]$ and $V_0 \in [1.89, 1.99]$, the constraints we used in the experiments can be satisfied. A 4th order polynomial is also investigated, and the allowable range for Z_0 and V_0 will increase to $[-2.92, -2]$ and $[1, 1.65]$, respectively. A 5th order polynomial will further increase the range for Z_0 and V_0 to $[-3.43, -2]$ and $[1.4, 2.15]$, respectively. Note that the ranges for V_0 and Z_0 shown in table 2 are generated from a given set of ks. We can also find other sets of ks to generate dif-

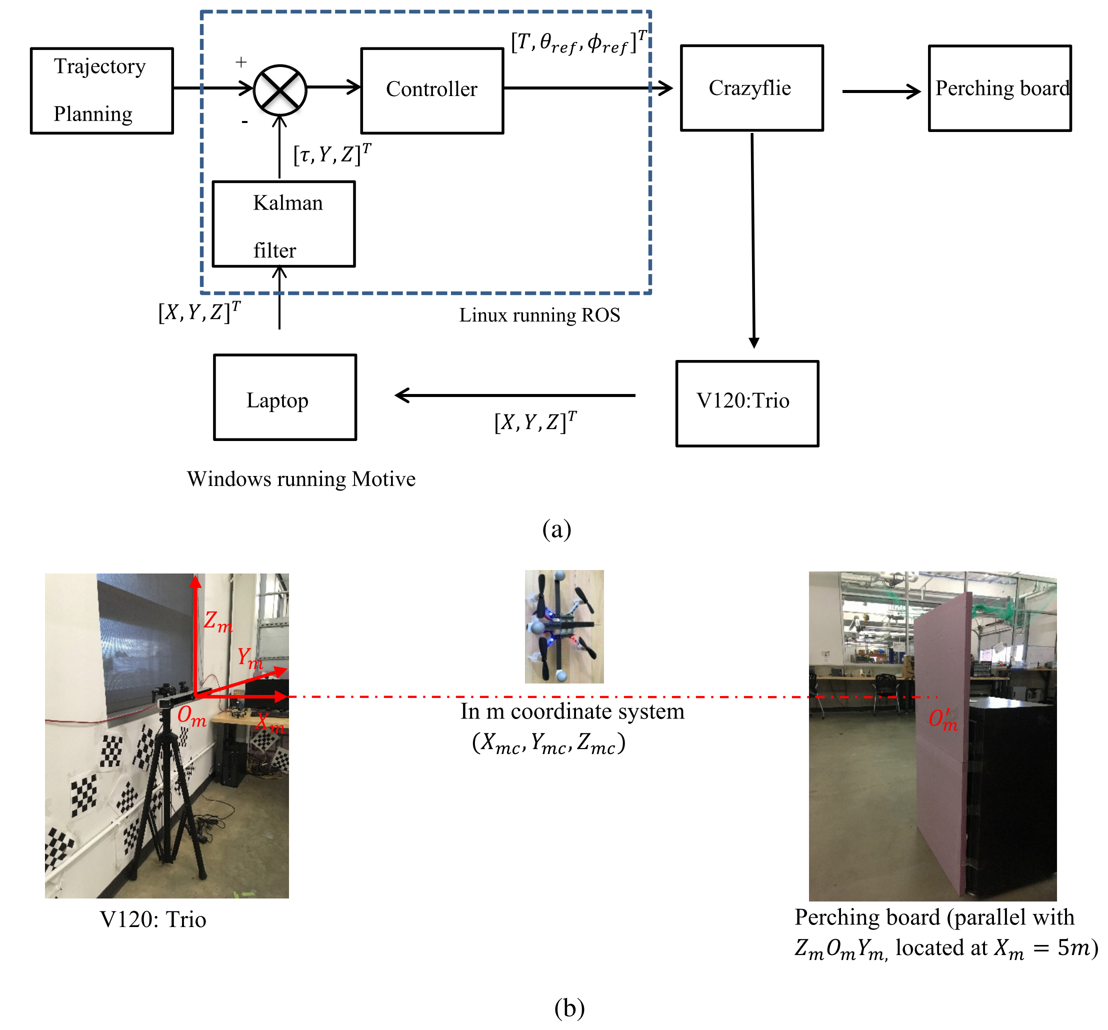

Figure 5. Experiment scheme and motion tracking system coordinate setup. The coordinate system origin O_m is projected as O'_m at the center of the perching board along X_m axis. The position and orientation of Crazyflie are measured by motion tracking system. The distance X between Crazyflie and the perching board is calculated by $X = X_{mc} - 5$. Then τ is calculated by definition and control command is generated by the τ controller and position controller. (a) Experiment scheme. (b) Coordinate system setup.

ferent ranges. Therefore, we believe that if the initial conditions can be roughly estimated, then we can find a fixed set of *k*s to satisfy the constraints as long as the estimated initial conditions fall inside the range for the set of *k*s. Fortunately, estimating initial distance and velocity from TTC is highly possible as demonstrated in some recent research results [30, 33].

4.2. Experimental results

After simulations, we experimentally test the tau based trajectory planning and control to enable perching for aerial robots. Although our ultimate goal is to leverage vision as feedback to estimate tau [27–29, 34], the hardware cannot provide accurate estimation with demanded computation time. In our initial step, we use a motion tracking system (V120: Trio, Optitrack) which can provide more accurate and faster feedback compared with monocular camera to track the position and orientation of the Crazyflie in the world frame. The V120: Trio is a pre-calibrated motion tracking system with three high speed cameras, of which the estimation error can be less than 1 mm. The largest distance it can track with larger markers is about 5.2 m. The quadcopter, Crazyflie, only weighs about 27 g (without markers for motion tracking) with a size of 92 × 92 × 29 mm. Due to the small size and light weight, it is very safe for indoor experiments. Currently, the perching mechanism is realized by a needle placed in front of Crazyflie, although more sophisticated mechanisms will be investigated in the future. A vertical foam board serves as the perching surface.

The experiment scheme is shown in figure 5(a). A motion tracking coordinate system is defined as shown in figure 5(b). In this system, origin is set at the optical center of the middle camera, X axis is defined along the optical axis of the middle camera, Z axis is defined upward, and Y axis is defined based on the former axis. The motion tracking system feedback the robot's position, which is sampled by a laptop running windows with Motive—a software provided by Optitrack. The same position is then transmitted to a desktop running

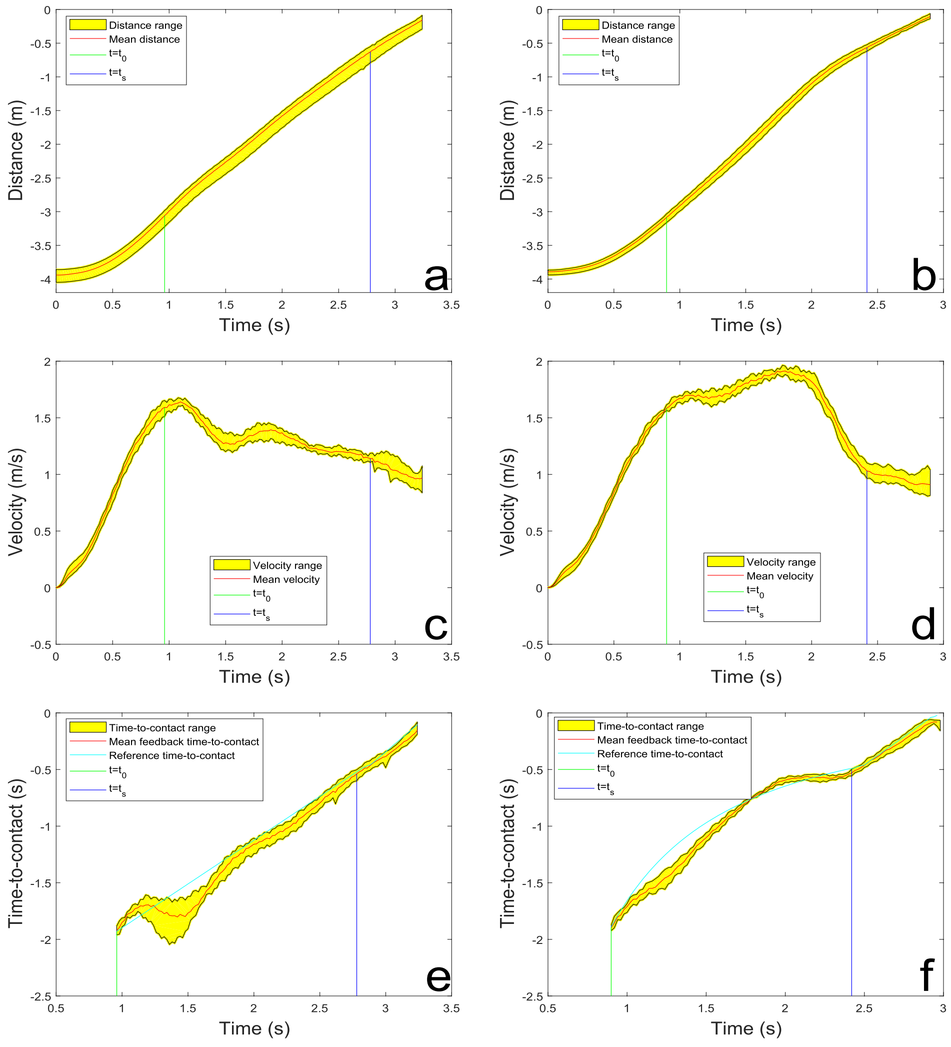

Figure 6. Experiment results for CTDTS (left column) and IPTS (right column). From top to the bottom are the distance, velocity, and tau for CTDTS and IPTS, respectively. The CTDTS and IPTS can realize the nonzero contact velocity, and IPTS requires shorter time for perching.

Linux with Robot Operating System. On this desktop, we first estimate the current τ using $\tau = X/\dot{X}$, where X is the distance along X_m axis between the Crazyflie and the perching board placed at 5 m away from the motion tracking system. And the velocity $\dot{X}$ is obtained from the position X using a Kalman filter. Based on the estimated τ and position in Y and Z, we compute the control command based on the controllers in equations (17) and (18), which will then be mapped to $[T, \theta_{ref}, \phi_{ref}]^T$ through the nonlinear transformation. The computed $[T, \theta_{ref}, \phi_{ref}]^T$ is then wirelessly transmitted to the Crazyflie through the Crazyradio to control its motion.

For the perching experiments, we use the same initial conditions with the simulation: $X_0 = -3$ m, $\dot{X}_0 = 1.5$ m s^{-1}, $V_u = 1$ m s^{-1}, and $V_l = 0.7$ m s^{-1}. The constraints are also the same based on the Crazyflie's capability: $V_{max} = 2.5$ m s^{-1}, and $|a_{max}| = 1.4$ m s^{-2}. Note that even the maximum acceleration of the Crazyflie and velocity are larger than the chosen constraints, the delay from wireless transmitting the control command confines the control performance. The constraints and perching velocity range are obtained after several tests with a manual remote controller. To make the Crazyflie have the specific initial conditions, we first accelerate the initially hovering Crazyflie for a while. When $X_0 \approx -3$ m and $\dot{X}_0 \approx 1.5$ m s^{-1} are satisfied, we initiate the first stage of the two-stage tau based strategy at time t_0. When the feedback tau of Crazyflie is $\tau = \tau_s = -0.5$ s, the second stage is initiated at time t_s. After that, the Crazyflie is controlled to perch on the foam board with a constant speed.

The PID parameters for the controllers are selected as follows: $k_{px2} = 0.4k_{px1} = 31\,000$, $k_{ix} = k_{dx} = 2500$, $k_{py} = 14\,500$, $k_{iy} = 2000$, $k_{dy} = 4500$, $k_{pz} = 20\,000$, $k_{iz} = 1500$, $k_{dz} = 3500$, which are tuned based on the thrust signal range which is in $(0, 65\,535)$. Similarly, the saturation function is selected as:

$$\sigma_\phi(\omega) = \sigma_\theta(\omega) = \begin{cases} -25°, & \text{if } \omega \leqslant -25° \\ \omega, & \text{if } -25° < \omega < 25° \\ 25°, & \text{if } \omega \geqslant 25° \end{cases} \tag{25}$$

$$\sigma_T(T) = \begin{cases} 10\,000, & \text{if } T \leqslant 10\,000 \\ T, & \text{if } 10\,000 < T < 65\,000\ . \\ 65\,000, & \text{if } T \geqslant 65\,000 \end{cases} \quad (26)$$

We conduct five experiments for both the CTDTS and the IPTS, respectively. CTDS is not tested since it cannot achieve non-zero contact velocity required for perching. We fill all the five experimental results area with yellow and plot the mean value of the five experimental results in red. The results of CTDTS, with a total perching time of 3.24 s, are shown in figures 6(a), (c) and (e), while the results of IPTS, with a total perching time of 2.90 s, are shown in figures 6(b), (d) and (f).

From the figures, we can see that both CTDTS and IPTS can be used for non-zero contact velocity perching. IPTS can generate faster perching while satisfy more constraints. For CTDTS, after t_0, there is still a small period that the Crazyflie keeps increasing speed. It is because from $t = 0$ s to t_0, the Crazyflie is always accelerating and it cannot respond fast enough to decelerate immediately. Despite this small period, it is almost decreasing by controlling the tau to follow the reference. Note that there is a period of increasing speed because of the tracking performance of the tau controller. When $\tau_{ref} > \tau$, the Crazyflie increases its speed to increase the actual τ and vice versa. Finally, it follows τ_{ref} very well and the contact speed is about 0.97 m s^{-1}. On the other hand, for the IPTS, similar to the simulation results, the speed first increases then decreases and finally contact the surface with a speed of about 0.91 m s^{-1}. Also the τ tracks the reference value quite well, although there exists some discrepancy after t_0 which might due to the delay of the Crazyradio. For both CTDTS and IPTS, the tau reference in the second stage can be used for controlling the aerial robot to fly with almost a constant speed even though the speed slightly decreases which is again caused by Crazyradio delay.

5. Conclusion and future work

Tau theory has been widely applied for robot motion control for tasks such as landing and docking. In this paper, we propose two TTC or tau based two-stage strategies to realize perching for aerial robots with a non-zero contact velocity. Specifically, we design CTDTS and IPTS as the tau reference and develop the corresponding control laws. Simulation results have shown that both CTDTS and IPTS can accomplish a non-zero contact velocity with IPTS being able to satisfy more constraints and generate a shorter perching time. Furthermore, perching experiments with a palm-size quadcopter also validate the faster perching of the IPTS. In the future, we will implement the vision based TTC estimation algorithm to replace the current motion tracking system. Also, we are working on a bistable perching mechanism for UAV to perch on different surfaces with different materials. In addition, simply controlling a UAV's velocity for perching onto surfaces might not be sufficient since a surface might have different orientations. To address this problem, we will also investigate the estimation of surface orientation as well as the attitude control of UAVs. Eventually, we aim to accomplish vision-based perching onto arbitrary surfaces.

Acknowledgments

This work is partially supported by the National Science Foundation under Grant IIS-1815476 and IIS-1815519.

ORCID iDs

Haijie Zhang https://orcid.org/0000-0002-3990-9225
Bo Cheng https://orcid.org/0000-0002-6982-0811
Jianguo Zhao https://orcid.org/0000-0003-0305-2597

References

[1] Floreano D and Wood R J 2015 Science, technology and the future of small autonomous drones *Nature* **521** 460–6

[2] Roderick W R, Cutkosky M R and Lentink D 2017 Touchdown to take-off: at the interface of flight and surface locomotion *Interface Focus* **7** 20160094

[3] Doyle C E, Bird J J, Isom T A, Kallman J C, Bareiss D F, Dunlop D J, King R J, Abbott J J and Minor M A 2013 An avian-inspired passive mechanism for quadrotor perching *IEEE/ASME Trans. Mechatronics* **18** 506–17

[4] Daler L, Klaptocz A, Briod A, Sitti M and Floreano D 2013 A perching mechanism for flying robots using a fibre-based adhesive *IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 4433–8

[5] Pope M T, Kimes C W, Jiang H, Hawkes E W, Estrada M A, Kerst C F, Roderick W R, Han A K, Christensen D L and Cutkosky M R 2017 A multimodal robot for perching and climbing on vertical outdoor surfaces *IEEE Trans. Robot.* **33** 38–48

[6] Graule M, Chirarattananon P, Fuller S, Jafferis N, Ma K, Spenko M, Kornbluh R and Wood R 2016 Perching and takeoff of a robotic insect on overhangs using switchable electrostatic adhesion *Science* **352** 978–82

[7] Kovač M, Germann J, Hürzeler C, Siegwart R Y and Floreano D 2009 A perching mechanism for micro aerial vehicles *J. Micro-Nano Mechatronics* **5** 77–91

[8] Moore J and Tedrake R 2009 Powerline perching with a fixed-wing uav *Proc. of the AIAA Infotech@Aerospace Conf. (Seattle, WA)*

[9] Mellinger D, Michael N and Kumar V 2012 Trajectory generation and control for precise aggressive maneuvers with quadrotors *Int. J. Robot. Res.* **31** 664–74

[10] Thomas J, Pope M, Loianno G, Hawkes E W, Estrada M A, Jiang H, Cutkosky M R and Kumar V 2016 Aggressive flight with quadrotors for perching on inclined surfaces *J. Mech. Robot.* **8** 051007

[11] Mohta K, Kumar V and Daniilidis K 2014 Vision-based control of a quadrotor for perching on lines *IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 3130–6

[12] Lussier Desbiens A, Asbeck A T and Cutkosky M R 2011 Landing, perching and taking off from vertical surfaces *Int. J. Robot. Res.* **30** 355–70

[13] Mehanovic D, Bass J, Courteau T, Rancourt D and Desbiens A L 2017 Autonomous thrust-assisted perching of

a fixed-wing uav on vertical surfaces *Conf. on Biomimetic and Biohybrid Systems* (Springer) pp 302–14
[14] Gibson J J 1958 Visually controlled locomotion and visual orientation in animals *Br. J. Psychol.* **49** 182–94
[15] Lee D N 1976 A theory of visual control of braking based on information about time-to-collision *Perception* **5** 437–59
[16] Lee D N 1998 Guiding movement by coupling taus *Ecological Psychol.* **10** 221–50
[17] Lee D N 2009 General tau theory: evolution to date *Perception* **38** 837
[18] Lee D N 2016 Tau in action in development *Action as an Organizer of Learning and Development* (*The Minnesota Symp. on Child Psychology Series* vol 33) (New York: Psychology Press) p 1
[19] Baird E, Boeddeker N, Ibbotson M R and Srinivasan M V 2013 A universal strategy for visually guided landing *Proc. Natl Acad. Sci.* **110** 18686–91
[20] Lee D N *et al* 1993 Visual control of velocity of approach by pigeons when landing *J. Exp. Biol.* **180** 85–104
[21] Lee D N and Reddish P E 1981 Plummeting gannets: a paradigm of ecological optics *Nature* **293** 5830
[22] Kendoul F 2014 Four-dimensional guidance and control of movement using time-to-contact: application to automated docking and landing of unmanned rotorcraft systems *Int. J. Robot. Res.* **33** 237–67
[23] McCarthy C, Barnes N and Mahony R 2008 A robust docking strategy for a mobile robot using flow field divergence *IEEE Trans. Robot.* **24** 832–42
[24] De Croon G, Ho H, De Wagter C, Van Kampen E, Remes B and Chu Q 2013 Optic-flow based slope estimation for autonomous landing *Int. J. Micro Air Veh.* **5** 287–98
[25] Kaneta Y, Hagisaka Y and Ito K 2010 Determination of time to contact and application to timing control of mobile robot *IEEE Robotics and Biomimetics* pp 161–6
[26] Izzo D and Croon G D 2012 Landing with time-to-contact and ventral optic flow estimates *J. Guid. Control Dyn.* **35** 1362–7
[27] Zhang H and Zhao J 2017 An integrated unmanned aerial vehicle system for vision based control *ASME 2017 Dynamic Systems and Control Conf.* (American Society of Mechanical Engineers) p V003T39A011
[28] Zhang H and Zhao J 2016 Biologically inspired vision based control using featureless time-to-contact estimations *IEEE Int. Conf. on Advanced Intelligent Mechatronics* (IEEE) pp 1133–8
[29] Zhang H and Zhao J 2017 Bio-inspired vision based robot control using featureless estimations of time-to-contact *Bioinspiration Biomimetics* **12** 025001
[30] Chirarattananon P 2018 A direct optic flow-based strategy for inverse flight altitude estimation with monocular vision and imu measurements *Bioinspiration Biomimetics* **13** 036004
[31] Van Breugel F, Morgansen K and Dickinson M H 2014 Monocular distance estimation from optic flow during active landing maneuvers *Bioinspiration Biomimetics* **9** 025002
[32] de Croon G C 2016 Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy *Bioinspiration Biomimetics* **11** 016004
[33] Ho H W, de Croon G C and Chu Q 2017 Distance and velocity estimation using optical flow from a monocular camera *Int. J. Micro Air Veh.* **9** 198–208
[34] Zhang H, Cheng B and Zhao J 2017 Extended tau theory for robot motion control *IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 5321–6
[35] Moore J, Cory R and Tedrake R 2014 Robust post-stall perching with a simple fixed-wing glider using lqr-trees *Bioinspiration Biomimetics* **9** 025013
[36] Jiang H, Pope M T, Hawkes E W, Christensen D L, Estrada M A, Parlier A, Tran R and Cutkosky M R 2014 Modeling the dynamics of perching with opposed-grip mechanisms *IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 3102–8
[37] Lee D N, Reddish P E and Rand D 1991 Aerial docking by hummingbirds *Naturwissenschaften* **78** 526–7
[38] Lee D N, Simmons J A, Saillant P A and Bouffard F 1995 Steering by echolocation: a paradigm of ecological acoustics *J. Comparative Physiol.* A **176** 347–54
[39] Wagner H 1982 Flow-field variables trigger landing in flies *Nature* **297** 147–8
[40] Davies M and Green P 1990 Optic flow-field variables trigger landing in hawk but not in pigeons *Naturwissenschaften* **77** 142–4
[41] Luukkonen T 2011 Modelling and control of quadcopter *Independent Research Project in Applied Mathematics* (Helsinki: Aalto University)
[42] Mellinger D and Kumar V 2011 Minimum snap trajectory generation and control for quadrotors *IEEE Int. Conf. on Robotics and Automation* (IEEE) pp 2520–5
[43] Preiss J A, Hönig W, Sukhatme G S and Ayanian N 2017 Crazyswarm: a large nano-quadcopter swarm *Proc. IEEE Int. Conf. on Robotics and Automation*
[44] Hoenig W, Milanes C, Scaria L, Phan T, Bolas M and Ayanian N 2015 Mixed reality for robotics *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (Hamburg, Germany)* pp 5382–7
[45] Kendoul F 2012 Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems *J. Field Robot.* **29** 315–78
[46] Zuo Z 2010 Trajectory tracking control design with command-filtered compensation for a quadrotor *IET Control Theory Appl.* **4** 2343–55
[47] Kendoul F, Yu Z and Nonami K 2010 Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles *J. Field Robot.* **27** 311–34
[48] Mahony R, Kumar V and Corke P 2012 Multirotor aerial vehicles *IEEE Robot. Autom. Mag.* **20** 20–32