

An Autonomous Quadrotor Avoiding a Helicopter in Low-Altitude Flights

Zhilong Liu, Aislan G. Foina

Abstract

Drones are the first robots to arrive in smart cities, and collision avoidance with helicopters is among the first barriers to the widespread drone use. In this paper, we develop control designs to analyze collision avoidance between quadrotors and helicopters. Communication radios capable of broadcasting traffic information is regarded as a promising solution to the Sense and Avoid (SAA) problem for small Unmanned Aircraft Systems (sUAS), especially for high speed flights where long-range sensing is not possible. A good example is Automatic dependent surveillance-broadcast (ADS-B). In this paper, a hybrid controller with ADS-B is introduced to enable a small quadrotor to perform optimal collision avoidance with a high-speed helicopter during autonomous navigation. First, a position controller and a safety controller are formulated independently. Then, they are combined together to form a hybrid automaton where transitions between the controllers are performed. Simulation shows the effectiveness of the hybrid controller.

I. INTRODUCTION

Recent advances in sensing and computing technology has made unmanned aerial vehicles/systems (UAV/UAS) low-cost but still increasingly capable of executing complex missions in challenging environments. They have gained popularity in a vast range of civilian applications, including search and rescue, disaster relief, and filming. Recently, the Federal Aviation Administration (FAA) has issued the Notice of Proposed Rulemaking (NPRM) on UAS certifications [1], indicating that a large number of UAS will present in the National Airspace System (NAS) in the near future. Our research aims at drones that travel in class G airspace. Big value of this sort envisaged by Amazon Prime Air [2] happens only when the drone is able to fly itself tens of miles from the distribution center to people's homes autonomously. One prerequisite for such flights is collision avoidance. If thousands of drones were flying freely in urban areas without any safety control, they would become a huge danger to existing aircraft. This paper is an exploration of the drone

collision avoidance problem in urban areas. Our main product is a hybrid controller that enables a UAV to perform collision avoidance with manned aircraft during autonomous navigation.

To formulate the collision avoidance problem, we first identify the manned and unmanned aircraft of interest. Airspace management proposals from the FAA [1] and corporations such as Google [3] and Amazon [2, 4] envisage drones flying below $500ft$, or $150m$ in class G airspace, i.e., below all the aircraft carrying people. At this altitude, news, police, and EMS helicopters are the manned aircraft of interest [3]. On the other hand, UAS flying in urban areas need vertical take-off, landing, hovering ability, and omni-directional maneuverability. This makes most urban UAS multirotors. Quadrotors are the simplest type of multirotors. Hence we focus on collision avoidance between quadrotors and helicopters. This is one of the first barriers to smart planes flying in smart cities [3].

In general, sense-and-avoid (SAA) technology could be divided into two categories, namely collaborative and non-collaborative. Non-collaborative SAA relies on remote sensing technologies, such as radars, laser range finders, and cameras, to detect obstacles. Aircraft with these sensors could detect a wide range of obstacles. However, for small UAS (sUAS) less than $50lb$ [1], we have limited payload capacity and limited detection range, typically less than $100m$ [5,6]. On the other hand, collaborative SAA relies on vehicle-to-vehicle (V2V) long-range communication, broadcasting and subscribing GPS-based traffic data of aircraft nearby. Google provides a discussion on collaborative SAA systems in [3]. The transceiver candidates are Dedicated Short-Range Communications (DSRC) and Automatic Dependent Surveillance-Broadcast (ADS-B). The communication range goes from $900m$ (DSRC) [7] to $24km$ (ADS-B) [8]. However, collaborative SAA also has several short-comings. First, the communication delay can be up to $1s$ for ADS-B. Second, the position information is obtained from GPS, typically accurate to $5m$ near ground [9]. In addition, obstacles outside the network are not detected.

The collision avoidance problem of interest involves high-speed vehicles. Typical quadrotors and helicopters can travel at $30m/s$ and $70m/s$, respectively [10, 11]. In the worst-case head-on collision, assuming an avoidance reaction takes $5s$, then a range requirement of $500m$ is implied. This result completely excludes the possibility of non-collaborative SAA. In addition, the $500m$ range requirement also indicates its insensitivity to GPS position errors ($5m$, or 1%). Therefore, collaborative SAA is the only feasible choice. Indeed, one can observe that the emergence of the small UAS industry today stimulates the rapid growth of small-size ADS-B technology. Examples are XPG-TR from Sagetech [8] and ADS-B ONE from NextGen UAS Transponders [12]. In this

paper, we assume that the vehicles of interest are equipped with collaborative SAA.

Given the collision detection mechanism, what remains is the navigation and collision avoidance algorithms, which constitute the contribution of this paper. The position controller is a set of concisely formulated sliding mode controllers augmented with dynamic surface control techniques. The safety controller generates optimal control actions when collisions are possible. We want to address the following scenario: A manned helicopter ambulance equipped with (ADS-B)-Out is flying at a low altitude to pick up a patient, when a quadrotor UAV with (ADS-B)-In is delivering packages autonomously. The quadrotor can receive real-time traffic information of the helicopter. If collision is possible, the quadrotor would avoid the manned helicopter. The avoidance maneuver is performed exclusively by the quadrotor UAV.

II. LITERATURE REVIEW

A. Position Controller

The quadrotor control literature can be divided into two classes. The first class focuses on attitude and altitude stabilization [13]–[17], while the second class controls the horizontal position as well [18]–[20]. Controlling the first class is straight forward because attitude and altitude are directly controlled by four independent control inputs. However, when horizontal position tracking is required, the problem becomes more difficult. The challenge is on resolving the underactuation issue, or using only four independent actuation inputs to handle all six degrees of freedom. Specifically, the tilting in roll and pitch are coupled with horizontal motions. To derive feasible horizontal control laws, the dynamic model is usually greatly simplified [18], [19] to avoid the explosion of terms when taking time derivatives.

In [18], Lee tried feedback linearization to fully control a quadrotor. The system model was reduced to include only thrust forces and gravity, and yet the derived control laws were quite lengthy. In addition, it augmented the state-space model with control inputs as additional states, yielding complicated dynamical input control laws. In [20], another similar feedback linearization method with dynamic inputs was presented. The model in [20] was more complete, and the term explosion problem became huge! In summary, it is practically impossible to derive concise control laws from a full quadrotor model by directly applying nonlinear control methods.

In this paper, we resolve the term explosion problem by combining nonlinear control methods with dynamic surface control (DSC) techniques. Specifically, we augment the quadrotor model with two first-order filters from the DSC literature [21]. With the DSC filters, we were able to

access slightly lagged versions of the required time derivatives, and indirectly formulate concise control laws, to fully control a quadrotor, including horizontal positions.

B. Avoidance Controller

Collision avoidance designs entail assumptions on cooperation between aircraft. The most conservative strategy is to set up the problem as a non-cooperative pursuit evasion game [22], in which a helicopter is chasing a quadrotor. It would be ideal to permit the helicopter to do whatever it likes, but the difference in vehicle capability shows this to be infeasible. From [11], a typical maximum speed of commercial helicopters is 160mph , or 70m/s . For a quadrotor, the maximum cruise speed is about 65km/h , or 30m/s [10]. If a helicopter is deliberately chasing a quadrotor, the quadrotor would eventually be caught, regardless of its avoidance efforts.

We focus instead on a semi-cooperative strategy, in which the responsibility for avoidance is placed on the quadrotor, and it gives way to the helicopter. This is also Google's current proposal [3]. We capture this in two specific assumptions.

- the quadrotor initiates the collision avoidance maneuvers;
- the helicopter maintains constant heading, altitude, and does not increase speed once the drone initiates collision avoidance.

The first attempt on addressing the collision avoidance problem in an optimal manner is presented in [23, 24]. The avoidance problem is formulated as a path planning problem, in which the optimal path could be found by solving a mixed integer programming problem. The forbidden zone is set by rectangular constraints for simplicity. However, this is a centralized approach, which requires avoidance actions from both vehicles. In addition, solving MIQP becomes computationally expensive when the number of binary variables increases.

In [25], the collision avoidance problem between multiple quadrotors is solved by a decentralized optimal control approach. The optimal control is defined as the maximum acceleration with a constant optimal heading. Consequently, a real-time avoid set can be computed. The avoidance is activated only when collision becomes inevitable. However, the safety controller is derived from a simplified two-dimensional model. Once the controller is applied to a full 3D dynamic model, the optimality is not longer guaranteed because of rotational inertia and aerodynamic drag. Although the model difference is almost negligible in the low-acceleration ($\sim 1.5\text{m/s}^2$) low-speed ($\sim 2\text{m/s}$) drone-drone avoidance scenario [25], it is no longer valid in high-acceleration ($\sim 10\text{m/s}^2$) high-speed ($\sim 100\text{m/s}$) drone-helicopter scenario.

In this paper, we modified the safety controller in [25] to fit our drone-helicopter scenario. First, we only use the safety set from [25] as an initial guess, and enlarge the safety set via an iterative algorithm to account for 3D dynamics. Second, the optimal control is only applied to the quadrotor, and the helicopter flies freely as if the quadrotor does not exist.

III. POSITION CONTROLLER

A. Dynamic Model

First, we need a quadrotor dynamic model. We adopt the same model introduced in [17], with a different notation. Define a fixed north-east-down (NED) inertial world frame \mathcal{W} and a non-inertial body frame \mathcal{B} attached to the center of gravity of the quadrotor (Figure 1a). The following is a list of variables used to describe the dynamics of a quadrotor.

- $\mathbf{X} = [X \ Y \ Z]^T$: quadrotor position in \mathcal{W} ;
- $\mathbf{V} = [V_X \ V_Y \ V_Z]^T$: quadrotor velocity in \mathcal{W} ;
- $\boldsymbol{\Theta} = [\phi \ \theta \ \psi]^T$: Euler angles roll, pitch, and yaw in \mathcal{B} , respectively;
- $\boldsymbol{\omega} = [\omega_x \ \omega_y \ \omega_z]^T$: quadrotor angular velocity in \mathcal{B} ;
- m : quadrotor mass;
- $\mathbf{I} = \text{diag}(I_x, I_y, I_z)$: mass moment of inertia in \mathcal{B} ;
- ω_{r_i} : motor speeds, $i = 1, 2, 3, 4$;
- $\Omega = -\omega_{r_1} + \omega_{r_2} - \omega_{r_3} + \omega_{r_4}$: sum of motor speeds;
- k_f, k_m : motor thrust and torque coefficients, respectively;
- c_t, c_r : translational and rotational friction coefficients, respectively;
- l : moment arm from the origin of \mathcal{B} to each motor.
- \mathbf{g} : gravitational acceleration, $[0 \ 0 \ g]^T$ in NED frame with $g = 9.81m/s^2$.

Assume that the motor forces are proportional to $\omega_{r_i}^2$, and the control inputs \mathbf{U} satisfy (1) with input constraints. Physically, the control inputs U_1, U_2l, U_3l, U_4 represent the total thrust and the total motor torques along the roll, pitch, and yaw axes, respectively. These are the control inputs for altitude and Euler angles. The constraints in (1) represent the physical capacity of the motors.

Other forces are gravity $m\mathbf{g}$, translation drag $c_t\mathbf{V}$, rotational drag $c_r\boldsymbol{\omega}^2$, and Coriolis forces from quadrotor body rotation and motor rotations. Define $R_{\mathcal{B} \rightarrow \mathcal{W}}$ to be the rotation matrix from \mathcal{B} to \mathcal{W} , and R_v to be a linear transformation from $\dot{\boldsymbol{\Theta}}$ to $\boldsymbol{\omega}$. The state-space equations could be written compactly as (2). We only gives a brief summary of the model to introduce enough notations for controller conversions. For details, refer to [17].

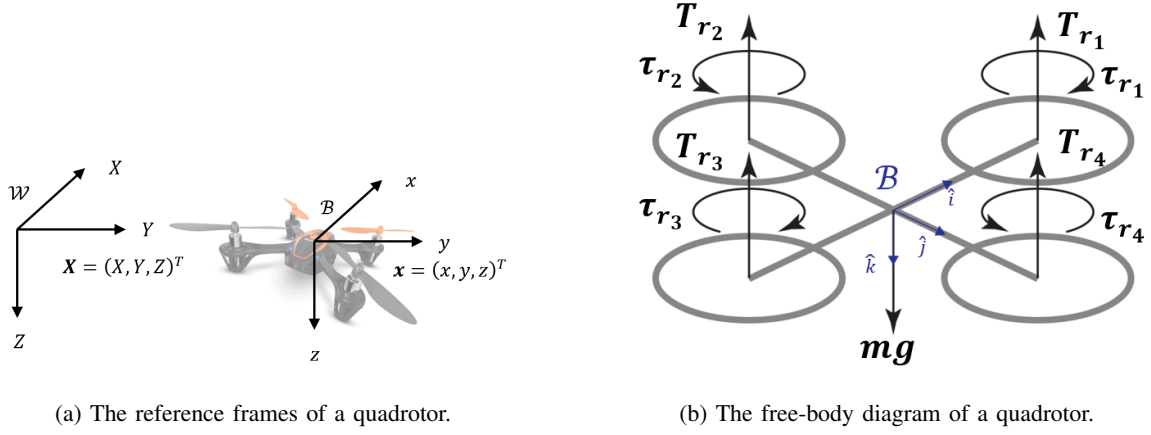


Fig. 1: Formulation of the quadrotor model

$$\begin{aligned}
 \mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} &= \begin{bmatrix} k_f & k_f & k_f & k_f \\ 0 & k_f & 0 & -k_f \\ k_f & 0 & -k_f & 0 \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} \omega_{r_1}^2 \\ \omega_{r_2}^2 \\ \omega_{r_3}^2 \\ \omega_{r_4}^2 \end{bmatrix} \quad \begin{aligned} 0 &< U_1 \leq 4k_f\omega_{r,max}^2 \\ -k_f\omega_{r,max}^2 &\leq U_2 \leq k_f\omega_{r,max}^2 \\ -k_f\omega_{r,max}^2 &\leq U_3 \leq k_f\omega_{r,max}^2 \\ -2k_m\omega_{r,max}^2 &\leq U_4 \leq 2k_m\omega_{r,max}^2 \end{aligned} \quad (1)
 \end{aligned}$$

$$\dot{\mathbf{X}} = \mathbf{V}$$

$$\dot{\boldsymbol{\Theta}} = \mathbf{R}_v^{-1}\boldsymbol{\omega}$$

$$\dot{\mathbf{V}} = \mathbf{g} + \frac{1}{m} \left(-c_t \mathbf{V} + \mathbf{R}_{B \rightarrow W} \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix} \right) \quad (2)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} \left(\boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) - \boldsymbol{\omega} \times \left(\mathbf{I}_r \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix} \right) - c_r \boldsymbol{\omega}^2 + \begin{bmatrix} U_2 l \\ U_3 l \\ U_4 \end{bmatrix} \right)$$

B. Sliding Mode Controllers with DSC Filters

Given the quadrotor model in Section III-A, we can follow the standard procedure of designing sliding mode controllers, and formulate four independent control laws governing the altitude and attitudes (Euler angles) [17], listed below in equation (3) and (4).

$$U_1 = \frac{m}{c\dot{\phi}c\theta} \left[\left(-g + \frac{c_t}{m} V_Z + \ddot{Z}_d \right) - \lambda_Z \dot{e}_Z - \eta_Z \text{sgn} S_Z \right] \quad (3)$$

$$U_4 = I_z \left[\left(-\frac{I_x - I_y}{I_z} \omega_x \omega_y + \frac{c_r}{I_z} \omega_z^2 + \ddot{\psi}_d \right) - \lambda_\psi \dot{e}_\psi - \eta_\psi \text{sgn} S_\psi \right]$$

$$U_3 = \frac{I_y}{l} \left[\left(-\frac{I_z - I_x}{I_y} \omega_x \omega_z + \frac{c_r}{I_y} \omega_y^2 + \frac{I_r}{I_y} \Omega \omega_x \right) - \lambda_\theta \dot{e}_\theta - \eta_\theta \text{sgn} S_\theta \right] \quad (4)$$

$$U_2 = \frac{I_x}{l} \left[\left(-\frac{I_y - I_z}{I_x} \omega_y \omega_z + \frac{c_r}{I_x} \omega_x^2 + \frac{I_r}{I_x} \Omega \omega_y \right) - \lambda_\phi \dot{e}_\phi - \eta_\phi \text{sgn} S_\phi \right]$$

with tracking errors and sliding surfaces defined as in equation (5), and λ 's and η 's as tuning parameters.

$$\begin{aligned} e_X &\triangleq X - X_d & e_\theta &\triangleq \theta - \theta_d & S_X &\triangleq \dot{e}_X + \lambda_X e_X & S_\theta &\triangleq \dot{e}_\theta + \lambda_\theta e_\theta \\ e_Y &\triangleq Y - Y_d & e_\phi &\triangleq \phi - \phi_d & S_Y &\triangleq \dot{e}_Y + \lambda_Y e_Y & S_\phi &\triangleq \dot{e}_\phi + \lambda_\phi e_\phi \\ e_Z &\triangleq Z - Z_d & e_\psi &\triangleq \psi - \psi_d & S_Z &\triangleq \dot{e}_Z + \lambda_Z e_Z & S_\psi &\triangleq \dot{e}_\psi + \lambda_\psi e_\psi \end{aligned} \quad (5)$$

Now, we can derive the horizontal control laws by augmenting our system with DSC filters. The overall horizontal motion control architecture is shown in Figure 2. It is a 3-step process. In the following, we show the design process in details.

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} \xleftarrow{SM} \begin{bmatrix} \bar{\theta} \\ \bar{\phi} \end{bmatrix} \xleftarrow{DSC} \begin{bmatrix} \theta_d \\ \phi_d \end{bmatrix} \xleftarrow{SM} \begin{bmatrix} U_3 \\ U_2 \end{bmatrix}$$

Fig. 2: The horizontal motion control architecture.

1) *derive synthetic inputs $\bar{\theta}$ and $\bar{\phi}$ via sliding mode (SM)*: First, take the time derivative of the sliding surfaces S_X and S_Y in (5) and set them equal to the robustness terms with parameters η_X and η_Y , respectively. Then, substitute \dot{V}_X and \dot{V}_Y from (2) and simplify with small angle approximations: $c\phi s\theta \approx \theta$ and $s\phi \approx \phi$, we have two new sliding mode control laws given by (6).

$$\begin{bmatrix} \bar{\theta} \\ \bar{\phi} \end{bmatrix} = R(\psi) \begin{bmatrix} \frac{m}{U_1} \left(\left(\frac{c_t}{m} V_X + \ddot{X}_d \right) - \lambda_X \dot{e}_X - \eta_X \text{sgn} S_X \right) \\ \frac{m}{U_1} \left(\left(\frac{c_t}{m} V_Y + \ddot{Y}_d \right) - \lambda_Y \dot{e}_Y - \eta_Y \text{sgn} S_Y \right) \end{bmatrix} \quad (6)$$

with rotation matrix

$$R(\psi) \triangleq \begin{bmatrix} c\psi & s\psi \\ s\psi & -c\psi \end{bmatrix}$$

Note that the roll $\bar{\theta}$ and $\bar{\phi}$ on the left-hand side of equation (6) represent the synthetic controls for the desired horizontal motions. We distinguish them from the actual pitch θ and roll ϕ .

2) *augment system with DSC filters*: To proceed, we need to somehow use equation (6) in the time derivative of S_θ and S_ϕ , so that the controls U_3 and U_2 can appear. However, as was mentioned in Section II-A, it requires taking time derivatives of (6), which leads to explosion of terms. To avoid that, we instead augment the dynamic system (2) with two first-order DSC filters shown in (7), to indirectly converge the desired pitch θ_d and roll ϕ_d to the synthetic controls $\bar{\theta}$ and $\bar{\phi}$, respectively [21]. The time constants τ_θ and τ_ϕ determine the convergence rates. The initial conditions are set to zeros in both filters.

$$\begin{aligned} \tau_\theta \dot{\theta}_d + \theta_d &= \bar{\theta}, & \theta_d(0) &= 0 \\ \tau_\phi \dot{\phi}_d + \phi_d &= \bar{\phi}, & \phi_d(0) &= 0 \end{aligned} \quad (7)$$

The DSC filters (7) allows us to compute $\dot{\theta}_d$ and $\dot{\phi}_d$ required in \dot{S}_θ and \dot{S}_ϕ , without taking time derivatives of $\bar{\theta}$ and roll $\bar{\phi}$. Therefore, no simplifications in the dynamics are needed to accommodate the explosion of terms. A semi-global stability proof via Lyapunov analysis is lengthy because it involves the error dynamics $(\dot{\theta}_d - \dot{\bar{\theta}})$ and $(\dot{\phi}_d - \dot{\bar{\phi}})$, which again involve explosion of terms. A standard proof is provided in [21]. Here, we simply adopt the results.

3) *derive control inputs U_3 and U_2 via sliding mode (SM)*: Now we can take the time derivatives of S_θ and S_ϕ , and set them equal to their robustness terms with parameters η_θ and η_ϕ , respectively, with $\ddot{\theta} \approx \dot{\omega}_y$ and $\ddot{\phi} \approx \dot{\omega}_x$.

$$\begin{aligned} \dot{S}_\theta &\approx \dot{\omega}_y + \lambda_\theta(\dot{\theta} - \dot{\theta}_d) \triangleq -\eta_\theta \text{sgn} S_\theta \\ \dot{S}_\phi &\approx \dot{\omega}_x + \lambda_\phi(\dot{\phi} - \dot{\phi}_d) \triangleq -\eta_\phi \text{sgn} S_\phi \end{aligned} \quad (8)$$

In equation (8), $\dot{\omega}_y$ and $\dot{\omega}_x$ include the control inputs U_3 and U_2 . They ($\dot{\omega}_y$ and $\dot{\omega}_x$), as well as $\dot{\theta}$ and $\dot{\phi}$, are given by (2), and $\dot{\theta}_d$ and $\dot{\phi}_d$ are given by re-expressing the DSC filters as (9).

$$\begin{aligned} \dot{\theta}_d &= \frac{1}{\tau_\theta} (\bar{\theta} - \theta_d) \\ \dot{\phi}_d &= \frac{1}{\tau_\phi} (\bar{\phi} - \phi_d) \end{aligned} \quad (9)$$

In equation (9), $\bar{\theta}$ and $\bar{\phi}$ are given by (6), and θ_d and ϕ_d are given by the DSC filter updates in (7). Then, the horizontal control laws are given by (4).

Lastly, the signum function ($sgn(\cdot)$) should be replaced by a smoothed function to avoid chattering. In this paper, we use a function with logistic and linear smoothing (10).

$$\eta sgn(S) \leftarrow \left[\eta_1 \left(\frac{2}{1 + e^{-cS}} - 1 \right) + \eta_2 S \right] \quad (10)$$

Throughout the derivation, we did not make any simplification to the full dynamic model (2) other than small angle approximations. The innovation lies in the adoption of DSC filters.

IV. SAFETY CONTROLLER

With the position controller, the quadrotor is capable of autonomous navigation to any given destinations. However, it is still prone to collision avoidance. In this section, the safety controller in [25] is modified to accommodate the drone-helicopter avoidance scenario.

A. Horizontal Safety Controller from Kinematic Model

To keep the system implementable in real time, the safety controller is derived from a simplified horizontal kinematic model [25]. The avoidance maneuver, $\mathbf{u}_h = [a_h \ \theta_h]^T$, is a vector representing the quadrotor's acceleration \mathbf{u}_h with magnitude a_h and direction θ_h , where θ_h is measured from the positive x-direction in the relative frame centered at the helicopter (Figure 3).

Figure 3 highlights some position trajectories in solid red lines under optimal control. The helicopter is at the origin, and the quadrotor has a relative position $\mathbf{x}(t) = (x(t), y(t))$ and is heading toward the helicopter with relative velocity $\dot{\mathbf{x}}(t)$. The quadrotor starts the avoidance maneuver at some initial time $t = t_0$ in the past when touching the avoid set ∂K_h . At the present time $t = 0$, the quadrotor flies tangent to the boundary of the safety set ∂S_h and does not enter S_h .

The safety set S_h is defined by the solid gray circle with radius r_{min} around the quadrotor. The set of all $\mathbf{x}(0)$ lies on the boundary of the safety set ∂S_h . The quadrotor should never enter S_h . The set of all possible $\mathbf{x}(t_0)$ that come as close as possible defines the boundary of the avoid set K_h , indicated by the dashed red curve in Figure 3. The complement of K_h is the maximal controlled invariant set [22], in which there is no restrictions on control. But within the avoid set K_h , optimal control has to be applied. Note that the relative frame is rotated such that $\dot{\mathbf{x}}(t_0) = 0$

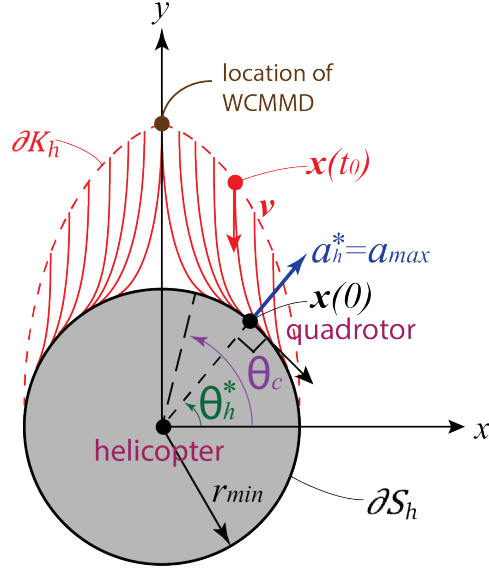


Fig. 3: The safety set S_h is in gray and the avoid set K_h is in red.

and $\dot{y}(t_0) < 0$ on ∂K_h . Therefore, points on ∂K_h has purely downward velocities. A backward rotation should be made after the optimal control angle θ_h^* is computed [25].

Define state variables $\mathbf{x}_h(t) = [\mathbf{x}(t); \dot{\mathbf{x}}(t)]$, the relative kinematics is:

$$\begin{aligned} \dot{\mathbf{x}}_h(t) &= f(\mathbf{x}_h(t), \mathbf{u}_h(t)) \\ \frac{d}{dt} \begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} &= \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ a_h(t) \cos \theta_h(t) \\ a_h(t) \sin \theta_h(t) \end{bmatrix} \end{aligned} \quad (11)$$

Follow the derivation in [25], we obtain the optimal acceleration for $t \in [t_0, 0]$ in (12).

$$a_h^*(t) = a_{max}; \quad \theta_h^*(t) = \text{atan2} \left(\frac{y(0)}{x(0)} \right) \quad (12)$$

Then, we need to find the avoid set $\partial K_h = \{(x(t_0), y(t_0))\}$. It could be calculated by simple geometry. Define the relative velocity $v = |\dot{y}(t_0)|$. Given a particular θ_h^* and r_{min} on ∂S_h , we have

$$\begin{aligned} x(t_0) &= \left(r_{min} - \frac{v \sin^2 \theta_h^*}{2a_{max}} \right) \cos \theta_h^* \\ y(t_0) &= \left(r_{min} + \frac{v}{2a_{max}} (1 + \cos^2 \theta_h^*) \right) \sin \theta_h^* \end{aligned} \quad (13)$$

From Figure 3, we can see that θ_h^* is only defined on certain parts of ∂S_h : $\theta_h^* \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$ for some $\theta_c \triangleq \theta_h^* \Big|_{x(t_0)=0}$ shown in (14). At $\theta_h^* = \theta_c$ (or $x(t_0) = 0$) and maximum v , we have the worst-case scenario, giving us the WCMMD to be $y(t_0) \Big|_{\theta_h^*=\theta_c}$ in (13).

$$\theta_c = \sin^{-1} \frac{\sqrt{2a_{max}r_{min}}}{v}, \quad 2a_{max}r_{min} < v^2 \quad (14)$$

Now we can answer the question on when to apply the optimal control. Except at $x(t_0) = 0$, each point on ∂K_h maps uniquely to a θ_h^* on ∂S_h . Therefore, we first generate ∂K_h using (13) for a dense set of $\theta_h^* \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$, then compare the quadrotor's relative position $\mathbf{x}(t)$ to points on ∂K_h . If $\mathbf{x}(t)$ equals to a particular point on ∂K_h , then we can find the corresponding θ_h^* and apply optimal acceleration $\mathbf{u}_h^* = [a_h^* \ \theta_h^*]^T$.

B. Iterative Algorithm

With dynamic model (2), we can apply control (12) at ∂K_h defined by (13) and (14). However, the quadrotor will intrude the safety set S_h due to rotational inertia and drag. To resolve the intrusion, we need to compute the worst-case earlier reaction time (WCERT), or the time the quadrotor has to react earlier before touching the safety set S_h , defined in Section IV-A, to prevent collision in the worst-case scenario.

Algorithm 1 computes WCERT for the horizontal safety controller. The function $\mathbf{x}(t) = \text{worst_case}(wcert)$ simulates the worst-case avoidance scenario when the kinematic controller reacts $wcert$ earlier, and return the relative horizontal position $\mathbf{x}(t)$ for the whole simulation period.

We start the simulation from the safety sets (13). If the minimum relative distance $\min_t |\mathbf{x}(t)|$ is less than r_{min} , then we increment $wcert$ and simulate again. In this case, we set dt to be $0.02s$ without drag and $0.05s$ with drag, which gives a spacial resolution of $2m$ and $5m$, respectively, when maximum speed is $v = 100m/s$.

Algorithm 1 is applied to a range of a_{max} and v defined in Section IV-A. The result is a look-up table $\text{WCERT}(a_{max}, v)$. The generation of WCERT values is time-consuming but performed

Data: $dt, wcert$

Result: $wcert$

```

dt;                                % time increment
wcert = 0;                         % current WCERT
x(t) = worst_case(wcert);
while min_t|x(t)| < r_min do
    | wcert = wcert + dt;
    | x(t) = worst_case(wcert);
end

```

Algorithm 1: Iterative algorithm to compute WCERT.

off line. In real flights, we can just run the light-weight kinematic safety controller in real-time but react WCERT(a_{max}, v) earlier. If the dynamic model in the simulation is close to the real dynamics, then we are safe.

C. Horizontal Control Conversion

To convert the horizontal kinematic control u_h^* to dynamic controls U_2 and U_3 , we assume that the quadrotor is in vertical equilibrium when performing the avoidance. Then the inputs that could affect the horizontal motions are the desired roll (ϕ_d), pitch (θ_d), and yaw (ψ_d). It turns out that yaw is not effective, so we choose roll and pitch. The conversion is straight forward. First, decompose the acceleration into components in the world frame \mathcal{W} , then rotate it to the body frame \mathcal{B} , and finally convert it to the desired pitch (θ_d) and roll (ϕ_d). Equation (15) summarizes the process.

$$\begin{aligned}
 a_X^* &= a_{max} \cos(\theta_h^*) & a_Y^* &= a_{max} \sin(\theta_h^*) \\
 a_x^* &= \cos\psi a_X^* + \sin\psi a_Y^* & a_y^* &= -\sin\psi a_X^* + \cos\psi a_Y^* \\
 \theta_d &= -\text{atan2}(a_x^*, g) & \phi_d &= \text{atan2}(a_y^*, g)
 \end{aligned} \tag{15}$$

D. The Hybrid Controller

The position controller and the safety controller together constitute a hybrid controller. The safety controller decides when the safety maneuver starts and terminates. Figure 4 shows the hybrid automaton of the controller switching mechanism. At the beginning, the quadrotor receives

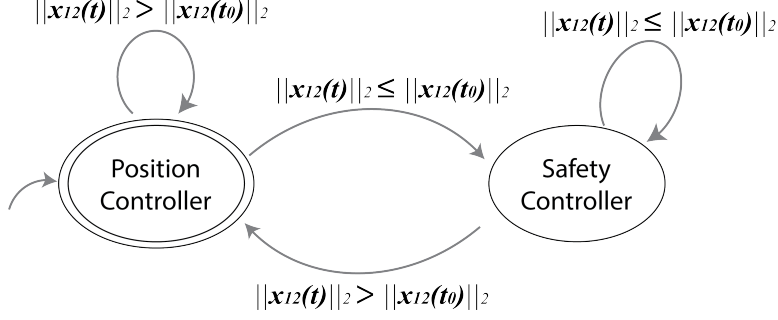


Fig. 4: The hybrid automaton indicating the controller switching mechanism.

a waypoint command, and the position controller starts to drive the quadrotor to the destination. When a helicopter is detected, an avoidance set calculation is performed using equation (13). The switching occurs when $\|\mathbf{x}_{12}(t)\|_2 \leq \|\mathbf{x}_{12}(t_0)\|_2$. Once the safety controller is activated, the quadrotor avoids the helicopter by applying a_{max} in the optimal direction. After $\|\mathbf{x}_{12}(t)\|_2 > \|\mathbf{x}_{12}(t_0)\|_2$, the optimal controller is deactivated, and the quadrotor switches back to position control. If another helicopter is passing by, the optimal control is enabled again.

V. SIMULATION

In this paper, we use the parameters in equation (16) to mimic a realistic model (3DR Solo [26]). We could not guarantee the parameters to cover all possible quadrotors, but we do choose conservative parameter values carefully. There are three uncertain key parameters which dominates the simulation results, namely the rotational inertia (RI) I , the translation drag coefficient c_t , and the communication delay Δt_{delay} .

$$\begin{aligned}
 m &= 1.5 \text{ kg} & I &= \text{diag}([5 \ 5 \ 8]e-3) \text{ kg} \cdot \text{m}^2 \\
 k_m &= 10^{-6} \text{ Nm} \cdot \text{s}^2 & k_f &= 5.0 \times 10^{-6} \text{ N} \cdot \text{s}^2 \\
 c_t &= 0.50 \text{ Ns/m} & c_r &= 0.10 \text{ Nm} \cdot \text{s}^2 \\
 l &= 0.2 \text{ m} & \mathbf{g} &= [0 \ 0 \ 9.81]^T \text{ m/s}^2
 \end{aligned} \tag{16}$$

- $\mathbf{I} = \text{diag}(I_x, I_y, I_z)$: We approximate the rotational inertia (RI) by approximating the 3DR Solo as a solid cylinder of radius 0.1 m and height 0.1 m . The calculation follows from [27], and the system identification results in [17] validates our calculation.
- c_t : Instead of adopting results from [17], which yields an unrealistically large terminal velocity, we picked a more conservative value of 0.5 Ns/m converted from [28, 29]. The

number is validated from static equilibrium between maximum horizontal thrust of $\sqrt{3}mg$ and translation drag $c_t v_{max}$, where $v_{max} = 50m/s$ is the terminal speed of the quadrotor [10].

- Δt_{delay} : The communication delay is set to $1.0s$ throughout the simulations, to approximate the worst-case behavior of ADS-B [30].

The simulation is expected to be realistic, although a hardware implementation is necessary to confirm the effectiveness of the hybrid controller. However, the hardware implementation is too involved, and thus not presented at this stage. It will be left as future work.

A. Safety Controller

In this section, two set of simulation results are presented. First, we examine how the 2D safety controller behaves when applied to the full 3D quadrotor model. Then, we present the WCMMD and WCERT results covering all practical cases in heatmaps. All safety sets are produced by gridding 120 points evenly for $\theta_h^* \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$.

1) *safety controller behavior*: Figure 5 shows the behavior of the safety controller. Two factors, namely the magnitude of the maximum acceleration and drag force, are investigated. The dashed red lines are the reference signals generated from the high-level safety controller, while the black solid lines are the tracking signals from the low-level sliding mode angle controllers [17]. The quadrotor is originally at rest. The parameters are

$$a_{max} = [2 \ 5 \ 10] m/s^2 \quad r_{min} = 20m \quad v_{12} = 70m/s$$

In Figure 5a, when a_{max} is low, say $2m/s^2$, the constant acceleration assumption holds very well, but in sacrifice it takes a long time ($4.9s$) to finish the avoidance maneuver. At the beginning of the avoidance maneuver, the tracking delay between the dashed red line and the solid black line is $0.28s$, which results in a ERT of $0.1s$.

The avoidance duration is shortened to half as much ($2.6s$) when $a_{max} = 10m/s^2$, but the desired pitch θ_d has some variation at the beginning and the end of the avoidance. The variation exists because the quadrotor could not achieve the desired acceleration instantaneously due to rotational inertia (RI). Nonetheless, the assumption of applying constant maximum acceleration remains valid. The tracking delay is $0.56s$, which gives a ERT of $0.2s$. When $a_{max} = 5m/s^2$, the result is somewhere in between. As a_{max} increases, the tracking delay increases, and ERT

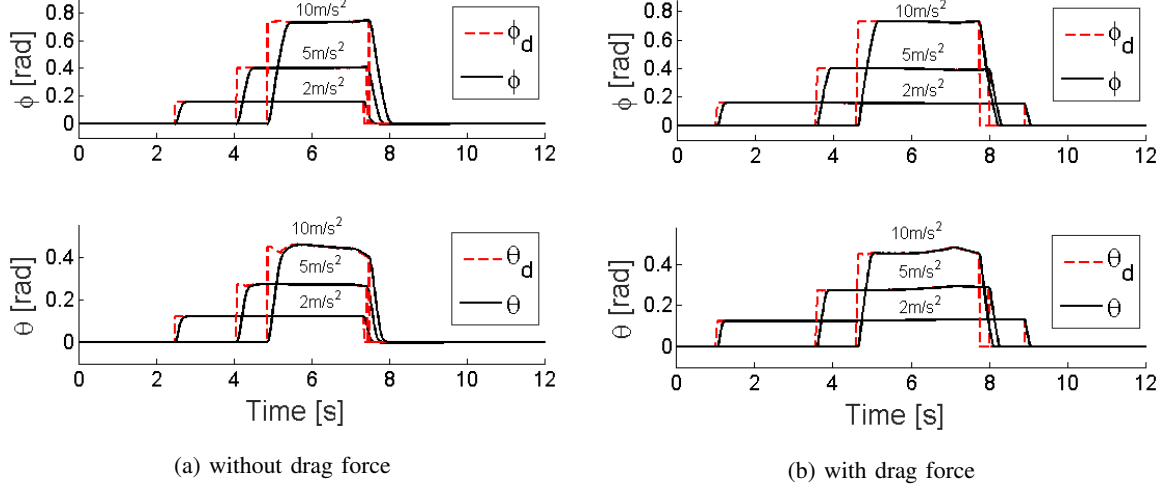


Fig. 5: The 2D optimal safety control is applied to a 3D quadrotor model.

increases almost proportional to the tracking delay. Therefore, without drag, ERT is solely determined by the tracking delay at the beginning of the avoidance maneuver.

In the normal case with drag force (Figure 5b), the variation in desired pitch (θ_d) is worse, although the desired roll (ϕ_d) stays almost constant. θ_d is generally increasing during the avoidance maneuver, especially at $a_{max} = 10m/s^2$. It makes sense because as the velocity goes up, the drag force goes up, and to generate an equivalent a_{max} , a larger pitch angle is needed. The avoidance duration is longer, range from $7.9s$ (when $a_{max} = 2m/s^2$) to $3.1s$ (when $a_{max} = 10m/s^2$).

However, with drag, ERT is not a strong function of tracking delay anymore. With similar tracking delays, the ERT is $1.54s$ when $a_{max} = 2m/s^2$ and $0.42s$ when $a_{max} = 10m/s^2$. This result is the complete opposite to the no-drag case. As a_{max} increases, the drag force increases faster, and to counteract the drag, a larger ERT is required.

2) *worst-case scenarios:* In this section, we discuss the simulation results of the safety controller considered in Section IV-A. The WCMMD and WCERT are observed when the quadrotor and helicopter are heading directly to each other at maximum horizontal relative velocity at the same altitude. The effect of rotational inertia (RI), drag forces, and communication delay described in Section V are evaluated.

Figure 6a shows the WCMMD result by using the horizontal safety controller on a kinematic quadrotor. It serves as the base of comparison. The ranges for a_{max} and v are chosen for reasons. A quadrotor can typically produce a maximum thrust of twice as its weight, or $2mg$ [31], which can generate a maximum horizontal acceleration of $17m/s^2$ when tilting at 60° at a constant

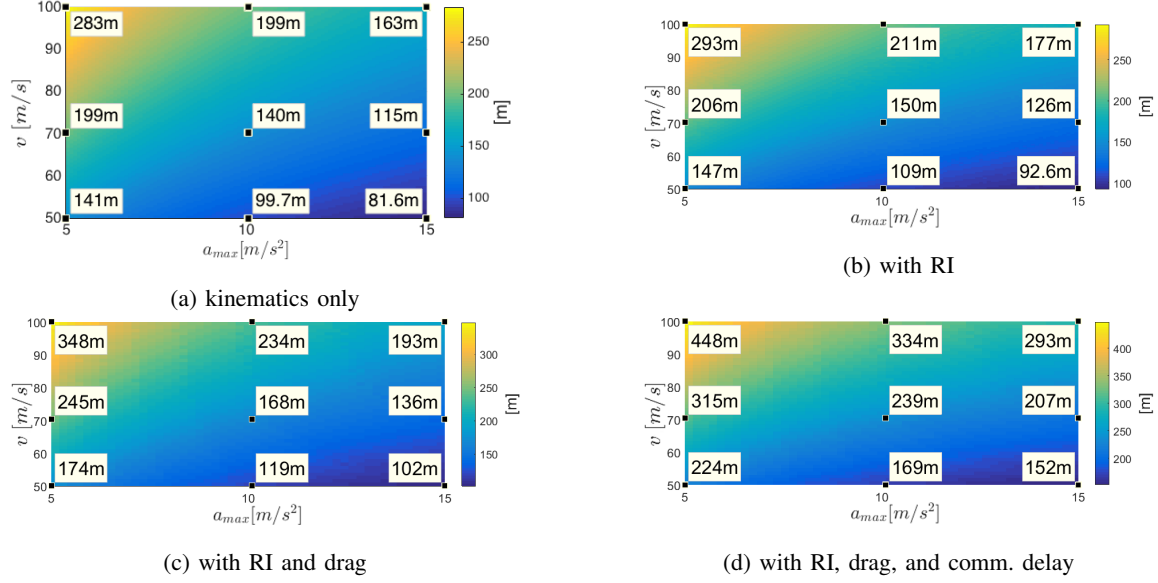


Fig. 6: Summary of WCMMD for the horizontal controller ($r_{min} = 20m$).

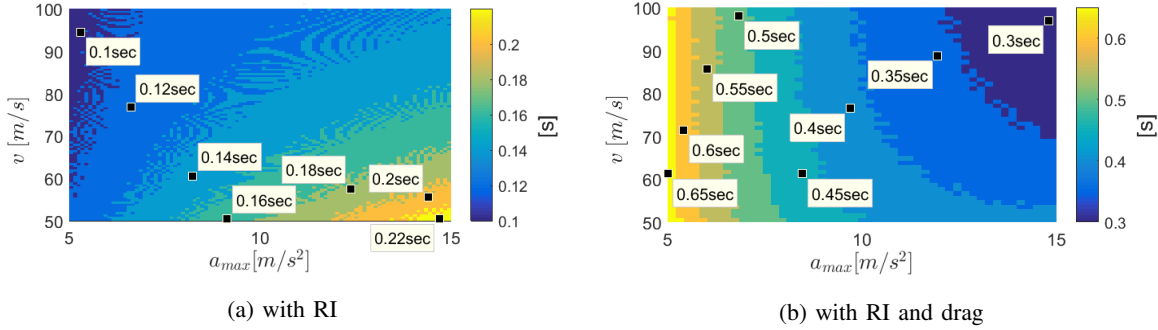


Fig. 7: Summary for the WCERT as defined in Section IV-A ($r_{min} = 20m$).

altitude. Therefore, we picked a_{max} to be between 5 and $15m/s^2$. A commercial helicopter typically has a maximum horizontal speed of $70m/s$ [11], while a quadrotor has a maximum cruise speed of $30m/s$ [10]. Therefore, we set the maximum horizontal relative velocity v to be between 50 and $100m/s$.

Figure 6b, 6c, and 6d give the WCMMD results by using the horizontal safety controller on a quadrotor with RI, drag, and communication delay, respectively. The axis ranges are the same as Figure 6a for comparison purpose.

The WCMMD increases faster and faster from the lower-right corner to the upper-left corner. In the lower-right corner, when a_{max} is high and v is low, we get small WCMMD ranging from

93m (no drag) to 152m (with drag and communication delay). In the upper-left corner, with small a_{max} and large v , we get large WCMMD ranging from 293m to 448m. Compared with Figure 6a, RI gives a maximum of 10m, or 3%, increase in the WCMMD. The WCMMD increments due to drag are 55m, or 18%, in the upper left corner and 10m, or 10%, in the lower right corner. Lastly, depending on the relative velocity v , the WCMMD increment of communication delay range from 50m (15%) to 100m (30%).

The more interesting result comes in the WCERT maps. First, the WCERT values without drag is always smaller than the ones with drag. The WCERT ranges from 0.10s to 0.22s without drag and from 0.30s to 0.65s with drag. Second, the two WCERT maps in Figure 7 show opposite patterns with respect to a_{max} . With RI only (Figure 7a), the WCERT increase as a_{max} increases. It is because a larger a_{max} implies a larger tilting angle (ϕ_d and θ_d), which takes more time for the sliding mode controllers to rotate the quadrotor. With both RI and drag (Figure 7b), the WCERT decreases as a_{max} increases. It says that as a_{max} gets larger, it becomes easier to overcome the drag, and the WCERT becomes smaller. To take into account the communication delay, one can simply add the delay (1s for ADS-B) on top of Figure 7.

B. Hybrid Controller

We tested the hybrid controller extensively in a wide range of scenarios to prove its effectiveness. Here we only present a typical package-delivery scenario. A quadrotor is typically designed to produce a maximum thrust of $2mg$ [31], which can give a maximum horizontal acceleration of $14m/s^2$ when hovering. Figure 8 shows a high-speed high-acceleration scenario. In Figure 8a, the quadrotor takes off from the origin and is trying to get to the point $(X, Y, Z) = (500m, 500m, -100m)$ when a helicopter is flying toward it in a straight line. The horizontal position signals of both the helicopter and the quadrotor are contaminated with Gaussian white noise with a standard deviation of 5m. In addition, the quadrotor is experiencing a horizontal wind disturbance of V_{wind} similar to [15]. In this scenario, the parameters are

$$\begin{aligned} r_{min} &= 20m & a_{max} &= 14m/s^2 & V_{wind} &= [-10, -10, 0]m/s \\ v_{heli} &= 70m/s & v_{quad} &= 28.3m/s & v_{12} &= 98.3m/s \end{aligned}$$

By employing the safety controller, the quadrotor successfully avoids the helicopter and gets to the final destination (Figure 8a). Assume that there is no communication delay, when the WCERT is set to 0.1s, the minimum relative distance between the two vehicles is $r_{12,min} = 21.8m$, which

is barely larger than the nominal threshold $r_{min} = 20m$, indicating that the safety controller performs almost optimally. Note that if the vehicles are equipped with ADS-B, the WCERT should take into account the $1s$ communication delay, giving a WCERT value of $1.1s$.

Figure 8b shows the time history of the horizontal command and control signals. In all subplots, the black solid lines give the actual signal trajectory. In the X , Y , and Z time history plots, the red dashed lines indicate desired destination. In the θ and ϕ plots, the blue dashed lines show the command generated from the angle controller, and the red dashed lines give command from the position controller. At the beginning, position control is enabled, and the actual trajectory (solid black) follow the position control command. At time $t = 6.7s$, the quadrotor switches to the safety (angle) control until $t = 8.3s$, which gives an avoidance duration of $1.6s$. The tracking delay in roll (ϕ) is long ($\sim 0.44s$) since the error between the desired roll and the actual roll is large at the beginning of the avoidance. During the avoidance, the quadrotor is under angle control, and the position control signal X_d and Y_d follow the actual position X and Y , instead. After the safety maneuver, the quadrotor switches back to position control smoothly. After arriving at the destination ($t \geq 32s$), the quadrotor still has nonzero roll and pitch to counteract the wind disturbance. Note that full thrust (U_1) is applied during the avoidance, indicating the choice of a_{max} is the highest achievable acceleration. Nonetheless, the quadrotor follows the generated path closely in position control mode, indicating that the generated path works well with the sliding mode controllers, under wind disturbance and measurement noise.

The efficiency of the safety controller could also be evaluated by the total flight time. In this scenario, the total flight time without avoidance is $27.5s$, and that with avoidance is $31.8s$, which gives an overall navigation delay of $4.3s$. After deducting the $1.6s$ avoidance maneuver, the navigation time is increased by $2.7s$. If the destination is 10 miles away, the total flight time would be around 10 minutes. The optimality ensures that the extra time spent on avoidance is not significant compared to the whole trip.

VI. CONCLUSION

In this paper, a hybrid controller with collision avoidance capability is presented in the context of high-speed heterogeneous flights. A concise position controller is formulated by adopting DSC techniques. In addition, a computationally feasible 2D optimal safety controller minimizing the avoidance duration is modified to be suitable for a 3D quadrotor model with rotational inertia. The combined hybrid controller is capable of performing optimal avoidance when flying to a destination autonomously.

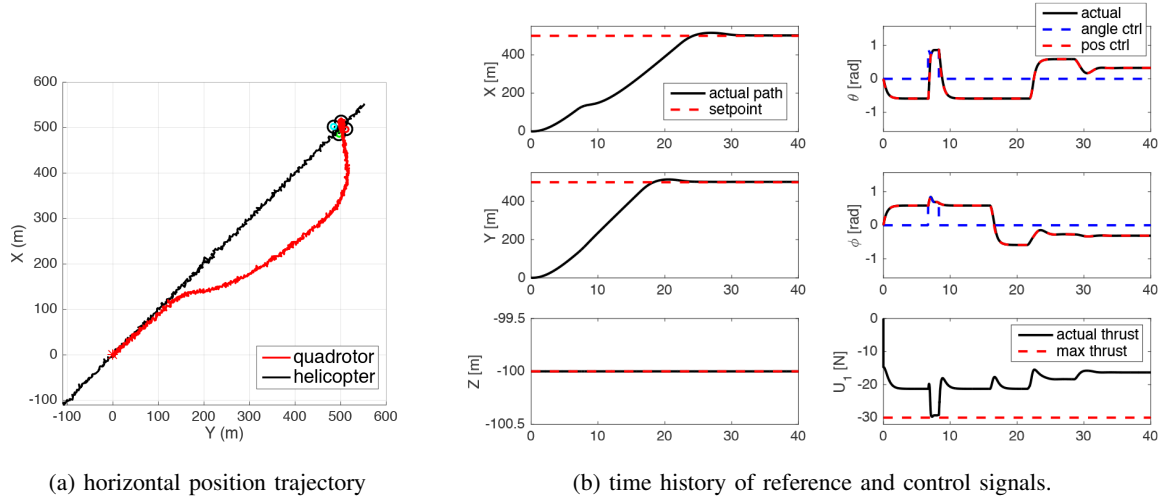


Fig. 8: The quadrotor avoids a helicopter on the way to a destination.

Some future work remains. First, a more sophisticated 2D model could be deployed. Second, the safety controller should be improved to handle a small multi-helicopter avoidance scenario. Third, the hybrid controller should be integrated with a path following algorithm, which is necessary to avoid multi-drone-to-drone collision in air highways. Lastly, a hardware implementation is necessary to confirm its effectiveness in real life.

REFERENCES

- [1] Operation and certification of small unmanned aircraft systems. [Online]. Available: https://www.faa.gov/regulations_policies/rulemaking/recently_published/media/2120-AJ60_NPRM_2-15-2015_joint_signature.pdf
- [2] Determining Safe Access with a Best Equipped, Best-Served Model for Small Unmanned Aircraft Systems.
- [3] Google uas airspace system overview.
- [4] Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems.
- [5] ART SYS 360. [Online]. Available: <http://www.artsys360.com/>
- [6] Hokuyo UTM-30LX. [Online]. Available: <http://www.autonomoustuff.com/hokuyo-utm-30lx.html>
- [7] I. C. Msadaa, P. Cataldi, and F. Filali, "A comparative study between 802.11 p and mobile wimax-based v2i communication networks," in *Next Generation Mobile Applications, Services and Technologies (NGMAST), 2010 Fourth International Conference on*. IEEE, 2010, pp. 186–191.
- [8] Sagetech. [Online]. Available: <http://www.sagetechnic.com/>
- [9] Gps accuracy. [Online]. Available: <http://www.gps.gov/systems/gps/performance/accuracy/>
- [10] 101kmph Quadcopter OFM Seeker 450 V2 Special Edition. [Online]. Available: https://www.youtube.com/watch?v=CjgMhO_qZT0
- [11] 11 of the Most Expensive Private Helicopters Money Can Buy. [Online]. Available: <http://www.refinedguy.com/2012/08/02/11-of-the-most-expensive-private-helicopters-money-can-buy/>

- [12] ADS-B ONE. [Online]. Available: <http://www.nextgenuastransponders.com/ads-b-one/>
- [13] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2451–2456.
- [14] J. Li and Y. Li, "Dynamic analysis and pid control for a quadrotor," in *Mechatronics and Automation (ICMA), 2011 International Conference on*. IEEE, 2011, pp. 573–578.
- [15] T. Madani and A. Benallegue, "Backstepping sliding mode control applied to a miniature quadrotor flying robot," in *IEEE Industrial Electronics, IECON 2006-32nd Annual Conference on*. IEEE, 2006, pp. 700–705.
- [16] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 2247–2252.
- [17] H. Bouadi, S. Simoes Cunha, A. Drouin, and F. Mora-Camino, "Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking," in *Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on*. IEEE, 2011, pp. 449–455.
- [18] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of control, Automation and systems*, vol. 7, no. 3, pp. 419–428, 2009.
- [19] R. Xu and Ü. Özgüner, "Sliding mode control of a quadrotor helicopter," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2006, pp. 4957–4962.
- [20] T. Taniguchi, L. Eciolaza, and M. Sugeno, "Quadrotor control using dynamic feedback linearization based on piecewise bilinear models," in *Computational Intelligence in Control and Automation (CICA), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–7.
- [21] B. Song and J. K. Hedrick, *Dynamic surface control of uncertain nonlinear systems: an LMI approach*. Springer Science & Business Media, 2011.
- [22] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.
- [23] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *European control conference*, vol. 1, 2001, pp. 2603–2608.
- [24] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 477–483.
- [25] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 4357–4363.
- [26] 3DR Solo Smart Drone. [Online]. Available: <http://3drobotics.com/solo-drone/>
- [27] F. P. Beer and E. Johnston, "Statics and dynamics," 1972.
- [28] AeroQuad Forum. [Online]. Available: <http://aeroquad.com/showthread.php?2714-Mathematical-modell-drag-coefficient>
- [29] B. R. Munson, D. F. Young, and T. H. Okiishi, *Fundamentals of fluid mechanics*. New York, 1990.
- [30] A. Helfrick, *Principles of avionics*. Avionics Communications, 2010.
- [31] How to Choose Motor and Propeller for Quadcopter and Multicopter. [Online]. Available: <http://blog.oscarliang.net/how-to-choose-motor-and-propeller-for-quadcopter/>