Quadrotors in Smart Cities Avoiding Helicopters

Zhilong Liu, Raja Sengupta, Aislan Foina

Abstract—Drones are the first robots to arrive in smart cities, and collision avoidance with helicopters is among the first barriers to widespread drone use. In this paper, we develop control designs and vehicle models to analyze collision avoidance between quadrotors and helicopters, and use them to derive the ranges required for sensing or communication radios used for collision avoidance. The models account for rotational inertia and drag. Our controllers consider both horizontal and vertical avoidance strategies and show that the vertical avoidance strategy can be better than the horizontal one, though it does entail a loss of altitude. Preliminary use of our models show that at one second delays, communication and sensing ranges need to be about 500 m, which is viable for communication radios on small drones but challenging for lightweight radar or lidar.

I. INTRODUCTION

The “smart city” concept can trace its origin back to the 1990s [1]. We envision cities smartened by automated planes and automated cars. While there are no automated cars on the market, drones are in the news everyday. The industry is selling upwards of 200,000 every month. Consumers, prosumers, and corporations are buying and flying them. The first wave of robots in our streets, fields, and backyards have arrived.

Drones, or Unmanned Aerial Systems (UAS), flying at low altitude are an essential part of smart cities. Our research is aimed at drones that travel, which is something only military drones do today. A photographer using a drone to photograph a site, drives the drone to the site, and flies it only upon reaching the site. 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. One prerequisite for such flight is collision avoidance - the focus of this paper.

This paper is an exploration of the drone collision avoidance problem in urban areas. Our main product is a methodology and software to assess the sensor or communication ranges and delays that will be required for collision avoidance systems. We also use the method to estimate preliminary range requirements.

II. PROBLEM FORMULATION

We first identify the manned and unmanned aircraft of interest. Airspace management proposals from the FAA [3] and corporations such as Google [4] and Amazon [2, 5] envisage drones flying below 500 ft, or 150 m, i.e., below all the aircraft carrying people. At this altitude, news, police, and EMS helicopters are the manned aircraft of interest [4]. 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 [4].

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 [6], 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 [7], a typical maximum speed of commercial helicopters is 160 mph, or 70 m/s. For a quadrotor, the maximum cruise speed is about 65 km/h, or 30 m/s [8]. 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 [4]. 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 second assumption is reasonable because we show the avoidance maneuver lasts for less than 4 seconds. This is a good starting point for deriving range requirements. An actual design would have to account for collision avoidance even when the helicopter is landing or taking off.

The control literature gives us two controller designs to perform collision avoidance, namely non-separated [9, 10] and separated designs [11]. The non-separated design in [10] combines avoidance with navigation and formulates a single optimization problem. The optimal path is found by solving a mixed integer programming problem. The separated design in [11], on the other hand, separates collision avoidance and navigation. The separated design entails some loss of optimality in the path, but it simplifies inter-aircraft coordination.

In this paper, we use collision avoidance controllers similar to [11] in combination with kinematic aircraft models, to produce the first estimates of the distance at which a drone needs to detect a helicopter and initiate collision avoidance maneuvers (Section III-A and III-B). The methods try to minimize this maneuver initiation distance so that the drone can go about its business, and switch to collision avoidance only when necessary, i.e., a shorter distance would make collision avoidance impossible. In other words, the methods give us minimum maneuver distances (MMD). The maneuver
distances depend on the relative velocity of the drone with respect to the helicopter and its acceleration capabilities. Larger relative velocities and smaller acceleration capabilities increase MMD. Derivations in [11] show that for a given acceleration capability, the largest MMD is realized when the quadrotor and helicopter are heading directly at each other with maximum velocity. We call this the worst-case MMD (WCMMMD) and it determines the sensing or communication range required for detection.

The collision avoidance controller in [11] executes constant altitude maneuvers. We call this a horizontal safety controller (section III-A). In section III-B we use similar ideas to design a vertical safety controller, which turns out to require shorter ranges than the horizontal one in some cases (see section IV-B). We then match these two controllers to quadrotor models in the literature accounting for rotational inertia and drag [12]. This is done through an iterative correction to the horizontal and vertical controllers explained in section III-C. The horizontal and vertical safety controllers output synthetic accelerations which must be translated into 4 motor speeds. For this we use a sliding mode controller in the literature [13] in section III-C. Thus section III describes the entire method for deriving the WCMMMD and thereby the range requirements.

Section IV applies the method to a range of parameter values to produce the WCMMMD with kinematic aircraft, and aircraft with rotational inertia and drag. The dynamics add about 18% to WCMMMD and thereby the detection ranges. We then add communication delays in a simple way to show that they result in another significant increase.

The paper concludes with the implications of the analyses for sensing and communication ranges. Collision avoidance by communication, now supported by Google, assumes the helicopter will transmit its GPS position, speed, and heading. The quadrotor will difference this received data from its own position, speed, and heading, to compute the relative values. These will then trigger the collision avoidance controller when necessary. We discuss the implication of our range findings for ADS-B [14] and DSRC [15].

III. Safety Controller Design

In this section, we first evaluate both the horizontal and vertical kinematic safety controllers in the quadrotor-helicopter collision avoidance scenario. Then, we developed an iterative algorithm to refine the kinematic avoid sets incrementally, to account for the quadrotor dynamics.

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 [11]. 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 \( \theta_h \). Where \( \theta_h \) is measured from the positive x-direction in the relative frame centered at the helicopter (Figure 1).

Figure 1 highlights some position trajectories in solid red lines under optimal control. The helicopter is at the origin, and the quadrotor has a relative position \( x(t) = (x(t), y(t)) \) and is heading toward the helicopter with relative velocity \( \dot{x}(t) \). The quadrotor starts the avoidance maneuver at some initial time \( t = t_0 \) in the past when touching the avoid set \( \partial K_h \). At the present time \( t = 0 \), the quadrotor flies tangent to the boundary of the safety set \( \partial S_h \) 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 \( x(0) \) that lie on the boundary of the safety set \( \partial S_h \). The quadrotor should never enter \( S_h \). The set of all possible \( x(t_0) \) that come as close as possible defines the boundary of the avoid set \( K_h \), indicated by the dashed red curve in Figure 1. The complement of \( K_h \) is the maximal controlled invariant set [6], 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 \( \dot{x}(t_0) = 0 \) and \( \dot{y}(t_0) < 0 \) on \( \partial K_h \). Therefore, points on \( \partial K_h \) have purely downward velocities. A backward rotation should be made after the optimal control angle \( \theta_h^* \) is computed [11].

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

\[
\begin{align*}
\dot{\mathbf{x}}_h(t) &= f(\mathbf{x}_h(t), \mathbf{u}_h(t)) \\
\frac{d}{dt} &= \begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ a_h(t)\cos\theta_h(t) \\ a_h(t)\sin\theta_h(t) \end{bmatrix} \\
&= \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ a_h(t)\cos\theta_h(t) \\ a_h(t)\sin\theta_h(t) \end{bmatrix}
\end{align*}
\]

Follow the derivation in [11], we obtain the optimal acceleration for \( t \in [t_0, 0] \) in (2).

\[
a^*_h(t) = a_{max}; \quad \theta^*_h(t) = \tan^{-1}\left(\frac{\dot{y}(0)}{\dot{x}(0)}\right)
\]
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 $\theta^*_h$ and $r_{\text{min}}$ on $\partial S_h$, we have

$$
\begin{align*}
x(t_0) &= \left( r_{\text{min}} - \frac{v \sin^2 \theta^*_h}{2a_{\text{max}}} \right) \cos \theta^*_h \\
y(t_0) &= \left( r_{\text{min}} + \frac{v}{2a_{\text{max}}} \left( 1 + \cos^2 \theta^*_h \right) \right) \sin \theta^*_h
\end{align*}
$$

(3)

From Figure 1, we can see that $\theta^*_h$ is only defined on certain parts of $\partial S_h$: $\theta^*_h \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$ for some $\theta_c \triangleq \theta^*_h \bigg|_{x(t_0) = 0}$ shown in (4). 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 $y(t_0) = \theta^*_h$ in (3).

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

(4)

Now we can answer the question on when to apply the optimal control. Except at $x(t_0) = 0$, each point on $\partial K_h$ maps uniquely to a $\theta^*_h$ on $\partial S_h$. Therefore, we first generate $\partial K_h$ using (3) for a dense set of $\theta^*_h \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$, then compare the quadrotor’s position $x(t)$ to points on $\partial K_h$. If $x(t)$ equals to a particular point on $\partial K_h$, then we can find the corresponding $\theta^*_h$ and apply optimal acceleration $u^*_h = [a^*_h \ 0]$. We define $x_v = [z(t); \dot{z}(t)]$. The new kinematics system is a combination of (1) and (5), with $a_h = 0$ in (1) and (5) being the vertical kinematics. The constraint on $a_v(t)$ is from [16].

$$
\begin{align*}
\dot{x}_v(t) &= f(x_v(t), a_v(t)) \\
\frac{d}{dt} [z(t); a_v(t)] &= [\dot{z}(t); -g \leq a_v(t) \leq g]
\end{align*}
$$

(5)

We define a new cylindrical safety set $\partial S_v$ by extruding the original safety set $\partial S_h$ in Figure 1 upward and downward by a minimum height $h_{\text{min}}$ (Figure 2). The goal is to ensure the quadrotor is outside of the cylinder.

To make a fair worst-case comparison with the horizontal safety controller, we assume that $x_v(t_0) = 0$, or the two vehicles are at the same altitude with zero vertical speed before avoidance. Formulate the optimal control problem according to [11], the optimal control is given by (6).

$$
a^*_v = g \text{sgn}(z(t_0))
$$

(6)

Ideally, (6) says that when the quadrotor is below the helicopter at $t_0$, then let it fall freely. Otherwise, when the quadrotor is above the helicopter, then apply the maximum thrust. In reality, the available downward acceleration is a little less than $g$ due to drag and the minimum thrust constraint in (8) to maintain the desired quadrotor pose.

The vertical avoid set is again derived from backward dynamics. For a dense set of $\theta_h \in [0, \pi]$ and $|z(t_0)| \in [-h_{\text{min}}, h_{\text{min}}]$, we calculate the new avoid set $\partial K_v = \{ (x(t_0), y(t_0)) \}$ satisfying (7).

$$
\begin{align*}
x(t_0) &= r_{\text{min}} \cos \theta_h \\
y(t_0) &= r_{\text{min}} \sin \theta_h + v \sqrt{2(h_{\text{min}} - |z(t_0)|)}
\end{align*}
$$

(7)

The shape of the avoid set $\partial K_v$ is a half circle offset by a certain distance away from the helicopter indicated by the dashed red lines in Figure 2. Once the quadrotor hits $\partial K_v$, the safety controller is activated. The WCMMD is obtained by substituting the relative altitude $z(t_0) = 0$, relative heading $\theta_h(t_0) = \pi/2$, and maximum relative speed $v$ into equation (7), or the quadrotor is at the same altitude as the helicopter and heading toward it right before the avoidance. The MMD drops to $r_{\text{min}}$ as $|z(t_0)|$ approaches $h_{\text{min}}$ either upward or downward.

In the vertical avoidance, we are concerned with not only MMD, but also the maximum vertical deviation (MVD), or the difference in altitude before and after the vertical avoidance. The worst-case MVD (WCMMD) in Figure 2 occurs together with the WCMMD. Right after a downward avoidance, the vertical speed of the quadrotor reaches a local maximum, and it will continue to go down for a while until it starts rising back to the original altitude. The downward MVD values are determined by $h_{\text{min}}$ when accelerating and the maximum thrust when decelerating.

C. Refine the Avoid Sets from Dynamic Model

The kinematic safety controllers is a good starting point. However, maximum accelerations $a^*_h$ and $a^*_v$ could not be achieved instantaneously and constantly in reality. For a quadrotor, rotational inertia and aerodynamic drag delays and reduces the maximum acceleration, respectively. Therefore, the safety controller have to be activated a little earlier before touching the avoid set $\partial K_h$, we define this time to be the earlier reaction time (ERT). In the worst-case, when the quadrotor and the helicopter are flying directly towards each other at maximum speed, we have the worst-case earlier reaction time (WCERT). In this section, we show an iterative algorithm to obtain the WCERT with a quadrotor dynamic
model. The WCERT values are then used to refine the avoid set to ensure safety.

1) Dynamic Model: First, we need a quadrotor dynamic model. 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 3). The following is a list of variables used to describe the dynamics of a quadrotor.

- $X = [X\ Y\ Z]^T$: quadrotor position in $\mathcal{W}$;
- $V = [V_X\ V_Y\ V_Z]^T$: quadrotor velocity in $\mathcal{W}$;
- $\Theta = [\phi\ \theta\ \psi]^T$: Euler angles of roll, pitch, and yaw in $\mathcal{B}$, respectively;
- $\omega = [\omega_x\ \omega_y\ \omega_z]^T$: quadrotor roll, pitch, and yaw rates in $\mathcal{B}$, respectively;
- $I = \text{diag}(I_x, I_y, I_z)$: mass moment of inertia in $\mathcal{B}$;
- $\omega_r$: 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.
- $g$: gravitational acceleration, $[0\ 0\ g]^T$ in NED frame with $g = 9.81\text{m/s}^2$.

Assume that the motor forces are proportional to $\omega_r^2$, and the control inputs $U$ satisfy (8) with input constraints. Physically, the control inputs $U_1, U_2, U_3, U_4$ 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 (8) represent the physical capacity of the motors.

Other forces are gravity $mg$, translation drag $c_tV$, rotational drag $c_r\omega^2$, and Coriolis forces from quadrotor body rotation and motor rotations. Define $R_{B\rightarrow W}$ to be the rotation matrix from $\mathcal{B}$ to $\mathcal{W}$, and $R_o$ to be a linear transformation from $\hat{\Theta}$ to $\Omega$. The state-space equations could be written compactly as (9). We only gives a brief summary of the model to introduce enough notations for controller conversions. For details, refer to an equivalent dynamic model provided in [12].

\[
U = \begin{bmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{bmatrix} = \begin{bmatrix}
k_f & k_f & k_f & k_f \\
0 & k_f & 0 & -k_f \\
k_f & 0 & -k_f & 0 \\
k_m & -k_m & k_m & -k_m
\end{bmatrix} \begin{bmatrix}
\omega_1^2 \\
\omega_2^2 \\
\omega_3^2 \\
\omega_4^2
\end{bmatrix}
\]

\begin{align}
4k_f\omega_{r,min}^2 & \leq U_1 \leq 4k_f\omega_{r,max}^2 \\
-4k_f\omega_{r,max}^2 & \leq U_2 \leq -4k_f\omega_{r,max}^2 \\
-2k_m\omega_{r,max}^2 & \leq U_3 \leq 2k_m\omega_{r,max}^2 \\
-2k_m\omega_{r,max}^2 & \leq U_4 \leq 2k_m\omega_{r,max}^2
\end{align}

\[
\dot{X} = V \\
\dot{\Theta} = R_{v}^{-1}\omega
\]

\[
V = g + \frac{1}{m} \begin{bmatrix}
-c_tV - R_{B\rightarrow W} \begin{bmatrix}
0 \\
0 \\
U_1
\end{bmatrix} \\
U_2
\end{bmatrix}
\]

\[
\omega = I^{-1} \left( \omega \times (I\omega) - \omega \times \left( \begin{bmatrix}
0 \\
I_r \\
0
\end{bmatrix} - c_r\omega^2 \right) + c_t\omega^2 \right) + \begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix}
\]

In this paper, we use the parameters in (10) to mimic a realistic model (3DR Solo [17]). We could not guarantee the parameters to cover all possible quadrotors, but we do choose conservative parameter values carefully to estimate sensing ranges. There are three uncertain key parameters which dominates the simulation results, namely, the rotational inertia $I$, the translation drag coefficient $c_t$, and the communication delay $\Delta t_{\text{delay}}$.

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

- $I = \text{diag}(I_x, I_y, I_z)$: We approximate the rotational inertia by approximating the 3DR Solo as a solid cylinder of radius $0.1m$ and height $0.1m$. The calculation follows from [18], and the system identification results in [12] validates our calculation.
- $c_t$: Instead of adopting results from [12], which yields an unrealistically large terminal velocity, we picked a more conservative value of $0.5\text{Ns/m}$ converted from [19, 20]. The number is validated from static equilibrium between maximum horizontal thrust of $\sqrt{3}mg$ and translation drag $c_t v_{\text{max}}$, where $v_{\text{max}} = 50\text{m/s}$ is the terminal speed of the quadrotor [8].
- $\Delta t_{\text{delay}}$: The communication delay is set to $1.0\text{sec}$ throughout the simulations, to approximate the worst-case behavior of ADS-B [21].

2) Horizontal Control Conversion: To convert the horizontal kinematic control $u^h_k$ 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 ($\phi_d$), pitch ($\theta_d$), and yaw ($\psi_d$). Any two of the three can completely specify the horizontal acceleration. In simulation, it turns out that yaw is not effective, so we use roll and pitch.

In vertical equilibrium, the conversion from $u_t^*$ to $\phi_d$ and $\theta_d$ is straightforward. Equation (11) shows the process. 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 ($\theta_d$) and roll ($\phi_d$).

\[
\begin{align*}
    a_x^* &= a_{max}\cos(\theta_h^*) \\
    a_y^* &= a_{max}\sin(\theta_h^*) \\
    a_z^* &= \cos\psi a_x + \sin\psi a_y \\
    a_{\theta}^* &= -\sin\psi a_x + \cos\psi a_y \\
    \theta_d &= -\arctan(2(a_x^*, g)) \\
    \phi_d &= \arctan(2(a_y^*, g))
\end{align*}
\]  

Lastly, a set of sliding mode controllers from equation (24) in [13] are adopted to drive the quadrotor to $\phi_d$ and $\theta_d$ using control inputs $U_2$ and $U_3$.

3) Vertical Control Conversion: To apply the vertical kinematic control to the 3D model, we use the following conversion by ignoring drag. In this case, we can control $U_1$ directly with constraints in (8), and no low-level controllers are needed.

\[
\begin{align*}
    \theta_d = \phi_d = 0; \\
    U_1^* \approx g - a_v^*
\end{align*}
\]

4) Iterative Algorithm: With dynamic model (9), we can either apply control (2) at $\partial K_v$ defined by (3), or apply (6) at $\partial K_v$ defined by (7). In both case, the quadrotor will intrude the safety set $S_h$ or $S_v$ as expected. To resolve the intrusion, we need to compute WCERT.

**Data:** $dt$, $wcert$  
**Result:** $wcert$

\[
\begin{align*}
    dt; & \quad \text{% time increment} \\
    wcert = 0; & \quad \text{% current WCERT} \\
    x(t) = \text{worst\_case}(wcert); & \\
    \text{while } \min_t|\dot{x}(t)| < r_{min} \text{ do} & \\
    \quad wcert = wcert + dt; & \\
    \quad x(t) = \text{worst\_case}(wcert); & \\
    \end{align*}
\]

**Algorithm 1:** Iterative algorithm to compute WCERT.

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

We start the simulation from the safety sets (3). If the minimum relative distance $\min_t|x(t)|$ is less than $r_{min}$, then we increment $wcert$ and simulate again. In this case, we set $dt$ to be 0.02 sec without drag and 0.05 sec with drag, which gives a spacial resolution of 2 m and 5 m, respectively, when maximum speed is $v = 100m/s$.

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

A similar algorithm with conditions on both $r_{min}$ and $h_{min}$ in the while loop is used to compute the WCERT for the vertical safety controller.

### IV. Simulation

In this section, we discuss the simulation results of the two controllers considered in Section III. 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 III-C are evaluated.

#### A. Horizontal Safety Controller

Figure 4a 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 $2mg$ [16], which can generate a maximum horizontal acceleration of $17m/s^2$ when tilting at $60^\circ$ at a constant altitude. Therefore, we picked $a_{max}$ to be between 5 and $15m/s^2$. A commercial helicopter typically has a maximum horizontal speed of $70m/s$ [7], while a quadrotor has a maximum cruise speed of $30m/s$ [8]. Therefore, we set the maximum horizontal relative velocity $v$ to be between 50 and $100m/s$.

Figure 4b, 4c, and 4d 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 4a 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 4a, 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 0.10sec to 0.22sec without drag and from 0.30sec to 0.65sec with drag. Second, the two WCERT maps in Figure 5 show opposite patterns with respect to $a_{max}$. With RI only (Figure 5a), the WCERT increase as $a_{max}$ increases. It is because a
larger $a_{\text{max}}$ implies a larger tilting angle ($\phi_d$ and $\theta_d$), which takes more time for the sliding mode controllers to rotate the quadrotor. With both RI and drag (Figure 5b), the WCERT decreases as $a_{\text{max}}$ increases. It says that as $a_{\text{max}}$ gets larger, it becomes easier to overcome the drag, and the WCERT becomes smaller.

B. Vertical Safety Controller

In this section, we present results for the vertical safety controller. For a cylindrical safety set with radius $r_{\text{min}}$ and height $h_{\text{min}}$ equal to 20m, the WCMMD result is shown in Figure 6. Unlike the horizontal controller, the result is purely a function of the relative velocity $v$, because we have a constant downward or upward acceleration of about $1g$. When $v = 100\text{m/s}$, we need a WCMMD of 220m with drag and 365m with drag. Drag gives WCMMD increments from 20m (17%) to 40m (18%). The 1sec communication delay adds the largest increments in WCMMD, and the drag adds about 40m. Rotational inertia has no effect because the quadrotor does not rotate in this scenario. Nonetheless, the WCMMD due to drag and delay are consistent with the results in Figure 4b, 4c, and 4d, respectively, at a constant acceleration of $10\text{m/s}^2$.

Table I shows the results for WCERT and WCMVD as defined in Section III-B. The WCERT is zero without drag and 0.42sec with drag. In this case, since the quadrotor does not rotate when performing vertical avoidance, there is no tracking delay, and thus no WCERT without drag. With drag, WCERT only depends on maximum acceleration $a_{\text{max}}$, which is a constant $g$, thus we have a constant WCERT.

<table>
<thead>
<tr>
<th>WCERT (sec)</th>
<th>WCMVD (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o drag</td>
<td>0.00</td>
</tr>
<tr>
<td>w/ drag</td>
<td>0.42</td>
</tr>
</tbody>
</table>

TABLE I: The WCERT and WCMVD results for the vertical safety controller.

Notice that the WCMVD with drag (26.6m) is smaller than without drag (39.3m). The difference is produced by drag. Without drag, the quadrotor in minimum thrust is
almost experiencing free fall, with only minimum thrust $4k_f \omega r_{min}^2$ to maintain the quadrotor pose. Constraint in (5) implies that the quadrotor stabilizes itself at an altitude deviation of about $2h_{min}$ by symmetry. With drag, we can actually stop the quadrotor faster because the vertical drag also contributes to the deceleration. Therefore, the conservative WCMVD is from the no-drag case.

C. Comparison

We see a trade-off when the practical constraints of the horizontal and vertical controllers are considered. Compared with the vertical controller, the horizontal controller has demanding requirements on the tilting angle, but it does not lose altitude given enough thrust. Therefore, both controllers have advantages in different applications. When the quadrotor is flying over a wide-open area or is heavily loaded, the vertical controller is preferred since losing $50m$ of altitude would not affect the navigation too much, but tilting the quadrotor by $45^\circ$ is difficult. On the other hand, if the quadrotor is flying in areas with high obstacle density without much payload, the horizontal controller becomes preferable. In this case, losing $10m$ of altitude may force the quadrotor to hit a tall building or tower.

D. Preliminary Sense-And-Avoid (SAA) Range Assessment

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 $50lb$ [3], we have limited payload capacity and limited detection range, typically less than $100m$ [22,23]. On the other hand, collaborative SAA relies on V2V long-range communication. Google provides a discussion on collaborative SAA systems in [4]. The transceiver candidates are Dedicated Short-Range Communications (DSRC) and Automatic Dependent Surveillance-Broadcast (ADS-B). The communication range goes from $900m$ (DSRC) [15] to $24km$ (ADS-B) [14]. However, communication delay could be up to $1sec$ (ADS-B), and obstacles outside the network are not detected.

From the estimated WCMMD results, we can conclude that the quadrotor needs to detect the helicopters as much as $350m$ (Figure 4c) and $265m$ (Figure 6) horizontally when using the horizontal and vertical safety controller, respectively, without communication delay. This result excludes the possibility of non-collaborative SAA, leaving collaborative SAA the only choice. ADS-B transceivers, including universal access transceiver (UAT) and 1090 MHz extended squitter, can achieve such ranges easily. A potential problem is the low transmission rate ($1Hz$) [14]. If the communication delay is taken into account, the conservative range would be up to $100m$ more. DSRC in Intelligent Transportation Systems is particularly suitable for medium range and delay-sensitive safety applications. In [15], the effective range of DSRC was evaluated to be about $900m$, with less than $60ms$ delays in low-data rate applications. Therefore, ADS-B and DSRC are both feasible SAA technology for drone-helicopter collision avoidance.

V. Conclusion

In this paper, we identified the low-altitude collision avoidance problem to be between an autonomous quadrotor and a manned helicopter. Horizontal and vertical avoidance controllers are derived from safety control theory and an iterative algorithm. The performance of both controllers are evaluated based on rotational inertia, drag, and communication delay. Numerical results are presented in terms of reaction distance and time. As an estimate, rotational inertia, drag, and communication delay give $3\%$, $18\%$, and $33\%$ increase in WCMMD, respectively. Compared with the horizontal safety controller, the vertical safety controller looks equally promising when the cylindrical safety set has an equal radius and height of $20m$. Depending on the constraints in applications, one may be more preferable than the other. Given the conservative WCMMD estimates, we also assess the feasibility of the existing SAA technologies. DSRC gives the best balance between range and communication delay.

Although we developed a method to assess the helicopter-quadrotor avoidance sensing range requirement, the numerical results are by no means perfect. They only give a preliminary assessment on sensing ranges. Future works remain to obtain better numerical results. First, the chosen values $r_{min}$ and $h_{min}$ only consider the size of the vehicles. If aerodynamic effects such as vortex are taken into account, the safety range may be much larger. Second, although the chosen quadrotor is realistic, it may not be a good vehicle for package delivery or filming. A comprehensive survey is needed to solidify the parameters. Third, we could refine the dynamic model by including wind in a more refined drag model.

REFERENCES