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 use of drones. 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 remote sensing is not possible. In this paper, we develop a hybrid safety controller to analyze collision avoidance between a quadrotor and a helicopter. The safety controller is capable of incorporating vehicle dynamics, wind distrubance, communication delay, and sensor uncertainty, to enable a small quadrotor to perform optimal collision avoidance with a high-speed helicopter, in autonomous navigations. Simulation shows the effectiveness of the 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. The NASA UTM project is an effort on enabling low-altitude UAS flights [2]. Big value envisaged by Amazon Prime Air [3] 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. Our research aims at drones that travel in class G airspace. This paper is an exploration of the drone collision avoidance problem in urban areas. The main contribution is a safety control framework that enables a UAS to perform collision avoidance with manned aircraft during autonomous navigation.

Research supported in part by NASA UARC Contract#: UCSCMCA-14-020 and NSF Contract#: CNS-1136141

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 [4] and Amazon [3, 5] envisage drones flying below $500ft$, or $150m$ in class G airspace, i.e., below all the aircraft carrying people. At this altitude, emergency flights involving news, police, and EMS helicopters occupy a majority of the possible flights [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, and quadrotors are the simplest type of multirotors. Hence we focus on collision avoidance between quadrotors and helicopters.

For safety critical type of application, it is important to capture the worst-case scenario. Although airspace design can efficiently separate helicopters from drones, the separation is not 100% guaranteed in urgent scenarios such as helicopter emergency flights, which accounts for a majority of low-altitude helicopter missions. In addition, it is not safe to assume exclusive avoidance responsibility to human pilots because pilot errors account for 85% of crashes in general aviation [6]. To ensure safety, the quadrotor must take avoidance actions in the last minute, before a collision becomes unavoidable.

In general, sense-and-avoid (SAA) technology could be divided into two categories, namely collaborative and non-collaborative [3]. Non-collaborative SAA relies on remote sensing technologies 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 detection range. Some state-of-the-art sensors with direct distance measurements are small radars [7], lidars [8], and 3D cameras [9], with ranges of $400m$, $60m$, and $30m$, respectively. Significant UAS SAA research has been conducted on them [10]–[14]. Monocular cameras may or may not give distance information, depending on the detection algorithms. For static obstacles, distance information is possible via monocular SLAM [15], [16]. However, for moving objects, only optical-flow algorithm is available [17]–[19], with no distance information. Other research efforts focus on fusion of both types of sensors [20]–[22].

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 [4]. The transceiver candidates are Dedicated Short-Range Communications (DSRC) and Automatic Dependent Surveillance-Broadcast (ADS-B). The communication range goes from $900m$ (DSRC) [23] to $24km$ (ADS-B) [24]. However, collaborative SAA also has several short-comings. The first one is communication delay. ADS-B

Out has a bounded transmission latency of up to $2s$ [25]. Other WiFi-based technologies such as WiMax and 4G LTE suffer from insufficient geo-spatial coverage and indefinite/unpredictable communication delays and are not suitable for safety critical communications [26]. Second, the position information is obtained from GPS, typically accurate to $5m$ near ground [27]. 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. The $400m$ detection range from non-collaborative SAA implies a $4s$ reaction time under perfect weather condition, which may not be realistic in worst-case collision with uncertainties. The large range requirement also indicates its insensitivity to GPS position errors ($5m$, or 1.3%). Therefore, collaborative SAA is a more 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 XPS-TR from Sagetech [24] ($100g$ and $12W$ max power) and PING2020 from uAvioni [28]. In this paper, we conservatively assume that the vehicles of interest are equipped with collaborative SAA. Therefore, the collision avoidance algorithm must be capable of handling communication delays.

Given the collision detection mechanism, what remains is the collision avoidance algorithm, which constitutes the contribution of this paper. The safety controller generates optimal control actions when collisions are possible. The algorithm should be able to incorporate bounded uncertainties such as communication delay, wind disturbance, sensor uncertainty, and vehicle dynamics. However, the uncertainties are vehicle and scenario specific, and their numerical values are not the focus of this paper. Our simulation only show a typical scenario, indicating our algorithm can handle these types of uncertainties. 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 UAS 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 UAS.

In the given drone-helicopter scenario, the current right of way procedure would require to stop all drone activities until the helicopter completes its mission. However, this procedure is highly inefficient and costly because quadrotors consume a lot of energy when hovering. In addition, the pilot error issue is not addressed. In a safety critical system, the proposed safety controller provide the last safety guarantee when everything in the current procedure failed.

II. LITERATURE REVIEW

A. *Formulation Assumptions*

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 [29], 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. A typical maximum speed of commercial helicopters is $160mph$, or $70m/s$. For a quadrotor, the maximum cruise speed is about $65km/h$, or $30m/s$. 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. In other words the quadrotor gives way to the helicopter. This is also Google's current proposal [4]. We capture this in two specific assumptions.

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

The first assumption preserves the "freedom of sky" for helicopter pilots. Horizontal maneuvers are preferred for two reasons. First, the helicopter produces unpredictable strong vertical vortex rings [30], seriously affecting the controllability of the quadrotor in vertical maneuvers. To account for the vortex uncertainty, a much larger safety region is necessary. Second, losing altitude in low-altitude flights increases the chance of hitting obstacles. Third, vertical maneuvers consume more power than horizontal ones. The second assumption enables us to take advantage of relative kinematics in modeling. It is conservative if we assume the helicopter pilot reacts rationally, meaning that he could only facilitate the avoidance maneuver when the quadrotor reacts.

B. *Collision Avoidance*

Aircraft collision avoidance algorithms has a rich history in the literature, and is generally referred to as the free-flight problem. There are geometry and probability based approaches [31], [32], as well as optimal control based methods [33]–[36]. Our goal is to reduce the side effect of avoidance as much as possible, which implies optimality.

The first attempt on addressing the collision avoidance problem in an optimal manner is presented in [35], [36]. 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 [34], 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 due to delays from rotational inertia and aerodynamic drag. Although the model difference is almost negligible in the low-acceleration ($\sim 1.5m/s^2$) low-speed ($\sim 2m/s$) drone-drone avoidance scenario [34], it is no longer valid in the high-acceleration ($\sim 10m/s^2$) high-speed ($\sim 100m/s$) drone-helicopter scenario.

In this paper, we modified the safety controller in [34] to fit our drone-helicopter scenario, and demonstrated that the avoidance could be handled much more efficiently. First, we only use the safety set from [34] as an initial guess, and enlarge the safety set via an iterative algorithm to account for uncertainties mentioned in Section I. Second, the optimal control is only applied to the quadrotor, and the helicopter flies freely as if the quadrotor does not exist.

III. Dynamic Model

First, we need a quadrotor dynamic model, which is used later in Section IV-C and IV-B. We adopt the same model introduced in [37], 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. The following is a list of variables used to describe the dynamics of a quadrotor.

- $\boldsymbol{X} = [X\ Y\ Z]^T$: quadrotor position in $\mathcal{W}$;
- $\boldsymbol{V} = [V_X\ V_Y\ V_Z]^T$: 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;
- $\boldsymbol{I} = diag(I_x, I_y, I_z)$: 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.
- $\boldsymbol{g}$: 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 $\boldsymbol{U}$ satisfy (1) with input constraints. Physically, the control inputs $U_1, U_2 l, U_3 l, U_4$ 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 mg, translation drag $c_t \boldsymbol{V}$, rotational drag $c_r \boldsymbol{\omega}^2$, and Coriolis forces from quadrotor body rotation and motor rotations. Define $R_{\mathcal{B}\to\mathcal{W}}$ 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 [37].

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

$$\begin{aligned} \dot{\boldsymbol{X}} &= \boldsymbol{V} \\ \dot{\boldsymbol{\Theta}} &= R_v^{-1}\boldsymbol{\omega} \\ \dot{\boldsymbol{V}} &= \boldsymbol{g} + \frac{1}{m}\left(-c_t\boldsymbol{V} + R_{\mathcal{B}\to\mathcal{W}}\begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}\right) \\ \dot{\boldsymbol{\omega}} &= I^{-1}\left(\boldsymbol{\omega}\times(I\boldsymbol{\omega}) - \boldsymbol{\omega}\times\left(I_r\begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix}\right) - c_r\boldsymbol{\omega}^2 + \begin{bmatrix} U_2 l \\ U_3 l \\ U_4 \end{bmatrix}\right) \end{aligned} \tag{2}$$

IV. SAFETY CONTROLLER

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 [34]. The safety controller includes two components. The first component is an optimal horizontal avoidance maneuver, $\boldsymbol{u_h^*} = [a_h^*\ \theta_h^*]^T$, which represents the

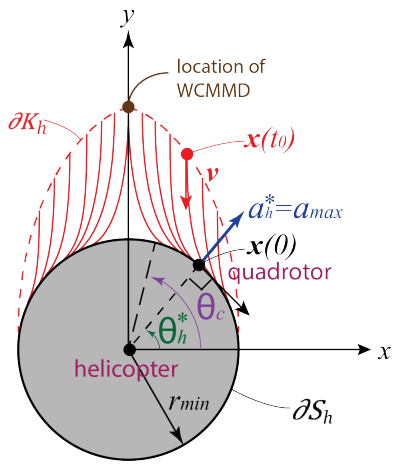


Fig. 1: The safety set S_h is in gray and the avoid set K_h is in red.

quadrotor's acceleration $\boldsymbol{u_h^*}$ with magnitude $a_h^* \in [0, a_{max}]$ and direction $\theta_h^* \in [0, 2\pi]$, where θ_h^* is measured from the positive x-direction in the relative frame centered at the helicopter (Figure 1). The second component is a switching surface, or the avoid set boundary ∂K_h, in which the optimal avoidance maneuver $\boldsymbol{u_h^*}$ must be initiated to avoid intrusion into a safety set S_h. We will elaborate the two components below in details.

Figure 1 highlights some position trajectories in solid red lines under optimal control $\boldsymbol{u_h^*}$. The helicopter is at the origin, and the quadrotor has a relative position $\boldsymbol{x}(t) = (x(t), y(t))$ and is heading toward the helicopter with relative velocity $\dot{\boldsymbol{x}}(t)$. The quadrotor should not enter the safety set S_h in the present time $t = 0$, defined by the solid gray circle with radius r_{min} around the helicopter. The set of all $\boldsymbol{x}(0)$ lies on the boundary of ∂S_h.

To avoid entering S_h, the quadrotor starts the avoidance maneuver at some initial time $t = t_0$ in the past when touching the avoid set K_h. The set of all possible $\boldsymbol{x}(t_0)$ 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 [29], in which there is no restrictions on control. If optimal control $\boldsymbol{u_h^*}$ is applied during $[t_0, 0]$, then the quadrotor would fly tangent to the boundary of the safety set ∂S_h at the present time $t = 0$, and does not enter S_h.

For convenience, the relative frame is rotated such that $\dot{x}(t_0) = 0$ and $\dot{y}(t_0) < 0$ on ∂K_h. In other words, points on ∂K_h only have velocity components in the -y axis. For legibility, we abuse the notation here by hiding this rotation. A backward rotation should be made after the optimal control angle θ_h^* is computed [34].

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

$$\begin{aligned} \dot{\boldsymbol{x}}_{\boldsymbol{h}}(t) &= f(\boldsymbol{x_h}(t), \boldsymbol{u_h}(t)) \\ \frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} &= \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ a_h(t)cos\theta_h(t) \\ a_h(t)sin\theta_h(t) \end{bmatrix} \end{aligned} \tag{3}$$

Consider the dynamical system (3) over the time interval $[t_0, 0]$ and define a continuously differentiable function $l : \mathbb{R}^4 \to \mathbb{R}$ in (4). If $l(\boldsymbol{x_h}(0)) \geq 0$, then we are outside of S_h [29].

$$l(\boldsymbol{x_h}(0)) = x^2(0) + y^2(0) - r_{min}^2 \tag{4}$$

The objective is to maximize the value function $J(\boldsymbol{x_h}(t), \boldsymbol{u_{2h}}(t), t) = l(\boldsymbol{x_h}(0))$ subjected to dynamical and input constraints, defined in (5). Note that the value function only has a terminal value at time $t = 0$.

$$\begin{aligned} \max_{\boldsymbol{u_{2h}}} \quad & J(\boldsymbol{x_h}(t), \boldsymbol{u_{2h}}(t), t) = l(\boldsymbol{x_h}(0)) \\ \text{subject to} \quad & f(\boldsymbol{x_h}, \boldsymbol{u_{2h}}) - \dot{\boldsymbol{x}}_{\boldsymbol{h}} = 0 \\ & 0 \leq a_{2h} \leq a_{max} \\ & 0 \leq \theta_{2h} \leq 2\pi \end{aligned} \tag{5}$$

Use Pontryagin's Maximum Principle [34], we obtain the optimal acceleration for $t \in [t_0, 0]$.

$$a_h^*(t) = a_{max}; \quad \theta_h^*(t) = atan2\left(\frac{y(0)}{x(0)}\right) \tag{6}$$

Secondly, to answer the question on when to apply the optimal control, we need to compute the avoid set $\partial K_h = \{(x(t_0), y(t_0))\}$ by kinematics [38]. Define the relative velocity $v = |\dot{y}(t_0)|$. Given a particular θ_h^* and r_{min} on ∂S_h, we have

$$\begin{aligned} x(t_0) &= \left(r_{min} - \frac{vsin^2\theta_h^*}{2a_{max}}\right) cos\theta_h^* \\ y(t_0) &= \left(r_{min} + \frac{v}{2a_{max}}\left(1 + cos^2\theta_h^*\right)\right) sin\theta_h^* \end{aligned} \tag{7}$$

From Figure 1, 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 (8). At $\theta_h^* = \theta_c$ (or $x(t_0) = 0$) and speed v, we have the worst-case scenario, giving us the worst-case minimum maneuver distance (WCMMD) to be $y(t_0)\Big|_{\theta_h^*=\theta_c}$ in (7).

$$\theta_c = sin^{-1}\frac{\sqrt{2a_{max}r_{min}}}{v}, \quad 2a_{max}r_{min} < v^2 \tag{8}$$

Each point on ∂K_h maps uniquely to a θ_h^* on ∂S_h. Therefore, we first generate ∂K_h using (7) for a dense set of $\theta_h^* \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$, then compare the quadrotor's relative position $\boldsymbol{x}(t)$ to points on ∂K_h. If $\boldsymbol{x}(t)$ is very close to a particular point on ∂K_h, then we can find the corresponding θ_h^* and apply the optimal acceleration $\boldsymbol{u_h^*} = [a_h^* \ \theta_h^*]^T$.

B. Horizontal Control Conversion

To convert the horizontal kinematic control $\boldsymbol{u_h^*}$ to horizontal control inputs 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, convert the optimal acceleration $[a^*, \theta^*]$ from the relative frame to the quadrotor body frame $\mathcal{B}$, and then convert it to the desired pitch (θ_d) and roll (ϕ_d). Equation (9) summarizes the process.

$$\begin{aligned} a_x^* = a_h^* cos(\theta^* - \psi) \quad a_y^* = a_h^* sin(\theta^* - \psi) \\ \theta_d = -atan2(a_x^*, g) \quad \phi_d = atan2(a_y^*, g) \end{aligned} \tag{9}$$

Lastly, a position controller with DSC filters [39] is implemented to track θ_d and ϕ_d defined in (9) using the control inputs U_2 and U_3 defined in (1). The detail implementation is not presented for simplicity.

C. Iterative Algorithm

With control conversion (9), we can apply kinematic control law (6) at ∂K_h, defined by (7) and (8), to the dynamical system (2) with control inputs (1). However, the quadrotor will intrude the safety set S_h due to control delay introduced from rotational inertia and aerodynamic drag in the full dynamic model. To resolve the intrusion, we need to compute the worst-case earlier

Data: dt, $wcert$

Result: $wcert$

dt; % time increment

$wcert = 0$; % current WCERT

$\boldsymbol{x}(t) = worst\_case(wcert)$;

while $min_t|\boldsymbol{x}(\boldsymbol{t})| < r_{min}$ **do**

  $wcert = wcert + dt$;

  $\boldsymbol{x}(t) = worst\_case(wcert)$;

end

Algorithm 1: Iterative algorithm to compute WCERT.

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 $\boldsymbol{x}(t) = worst\_case(wcert)$ simulates the worst-case avoidance scenario when the kinematic controller reacts $wcert$ earlier, and return the relative horizontal position $\boldsymbol{x}(t)$ for the whole simulation period.

We start the simulation from the safety sets (7). If the minimum relative distance $min_t|\boldsymbol{x}(\boldsymbol{t})|$ is less than r_{min}, then we increment $wcert$ and simulate again. In this case, we set dt to be $0.02s$ in the case 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 WCERT(a_{max},v). With this look-up table, WCMMD can be re-computed. 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. To obtain usable results, system identification should be performed to establish conservative quadrotor parameters, which is out of the scope of this paper and not discussed. Interested readers can refer to [37].

D. The Hybrid Controller

The position controller with DSC filters mentioned in Section IV-B and the safety controller together constitute a hybrid controller. The safety controller decides when the safety maneuver

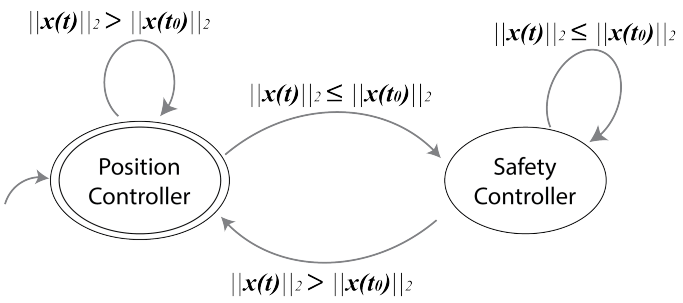


Fig. 2: The hybrid automaton indicating the controller switching mechanism.

starts and terminates. Figure 2 shows the hybrid automaton of the controller switching mechanism. At the beginning, the quadrotor receives 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 (7). The switching occurs when the quadrotor intrudes ∂K_h, or$||\boldsymbol{x}(t)||_2 \leq ||\boldsymbol{x}(t_0)||_2$. Once the safety controller is activated, the quadrotor avoids the helicopter by applying a_{max} in the optimal direction θ_h^*. After $||\boldsymbol{x}(t)||_2 > ||\boldsymbol{x}(t_0)||_2$, 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

The simulations are divided into two parts. The first part examines the behavior of the safety controller, and the second part presents how the hybrid automaton behaves in the scenario defined at the end of Section I.

Before presenting the results, we should first justify the parameters chosen for the simulations. The parameter values, listed in (10), are chosen carefully and conservatively to model a realistic quadrotor model (3DR Solo [40]). However, these parameters vary a lot from different quadrotors, communication systems, and wind conditions. Therefore, the results are not intended to cover all possible cases. What we can guarantee is that our safety control algorithm could give conservative results given conservative parameters for a given quadrotor with a known communication delay and bounded wind disturbance.

$$
\begin{array}{lll}
m = 1.5\ kg & r_{min} = 20m & I = diag([5\ 5\ 8]e\text{-}3)\ kg \cdot m^2 \\
k_m = 10^{-6}\ Nm \cdot s^2 & l = 0.2m & k_f = 5.0 \times 10^{-6}\ N \cdot s^2 \\
c_t = 0.50\ Ns/m & c_r = 0.10\ Nm \cdot s^2 & \boldsymbol{g} = [0\ 0\ 9.81]^T\ m/s^2
\end{array}
\tag{10}
$$

There are four key parameters which dominates the simulation results, namely the rotational inertia (RI) I, the translation drag coefficient c_t, the communication delay Δt_{delay}, and the minimum separation distance r_{min}.

- $\boldsymbol{I} = diag(I_x, I_y, I_z)$: We approximate the rotational inertia (RI) by approximating the 3DR Solo as a solid cylinder of radius $0.1m$ and height $0.1m$. The calculation follows from [38], and the system identification results in [37] validates our calculation.
- c_t: Instead of adopting results from [37], which yields an unrealistically large terminal speed, we picked a more conservative value of $0.5Ns/m$, which yields a terminal horizontal speed of $v_{max} = 50m/s$, by assuming a maximum horizontal thrust of $\sqrt{3}mg$ and translation drag of $c_t v_{max}$ [41].
- Δt_{delay}: In all the feasible collaborative SAA technologies discussed in Section I, ADS-B gives the largest bounded communication delay, $2.0s$. We use this conservative value throughout the simulations, to approximate the worst-case behavior [25], [42].
- r_{min}: The minimum separation distance r_{min} is tunable to satisfy future minimum separation standards. It is set $20m$, or twice the size of a typical helicopter, throughout the simulations.

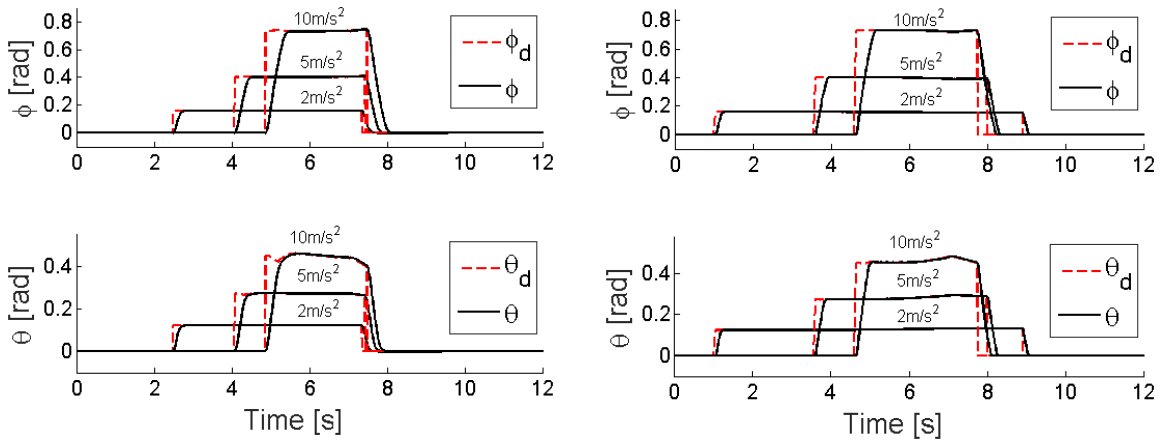
The simulation is expected to be realistic, although a hardware implementation is necessary to confirm the effectiveness of the hybrid controller. It will be left as future work.

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 a full 3D quadrotor model. Then, we present the WCMMD and WCERT results after running Algorithm 1 defined in Section IV-C, covering all practical accelerations and relative velocities for our specific setup. All safety sets are produced by griding 120 points evenly for $\theta_h^* \in [0, \theta_c] \cup [\pi - \theta_c, \pi]$.

1) safety controller behavior: Figure 3 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 [37]. The quadrotor is originally at rest. The parameters are

$$a_{max} = [2\ 5\ 10]\, m/s^2 \qquad v = 70m/s$$



(a) without drag force

(b) with drag force

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

In Figure 3a, 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 an 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 increases almost propotional 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 3b), 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.

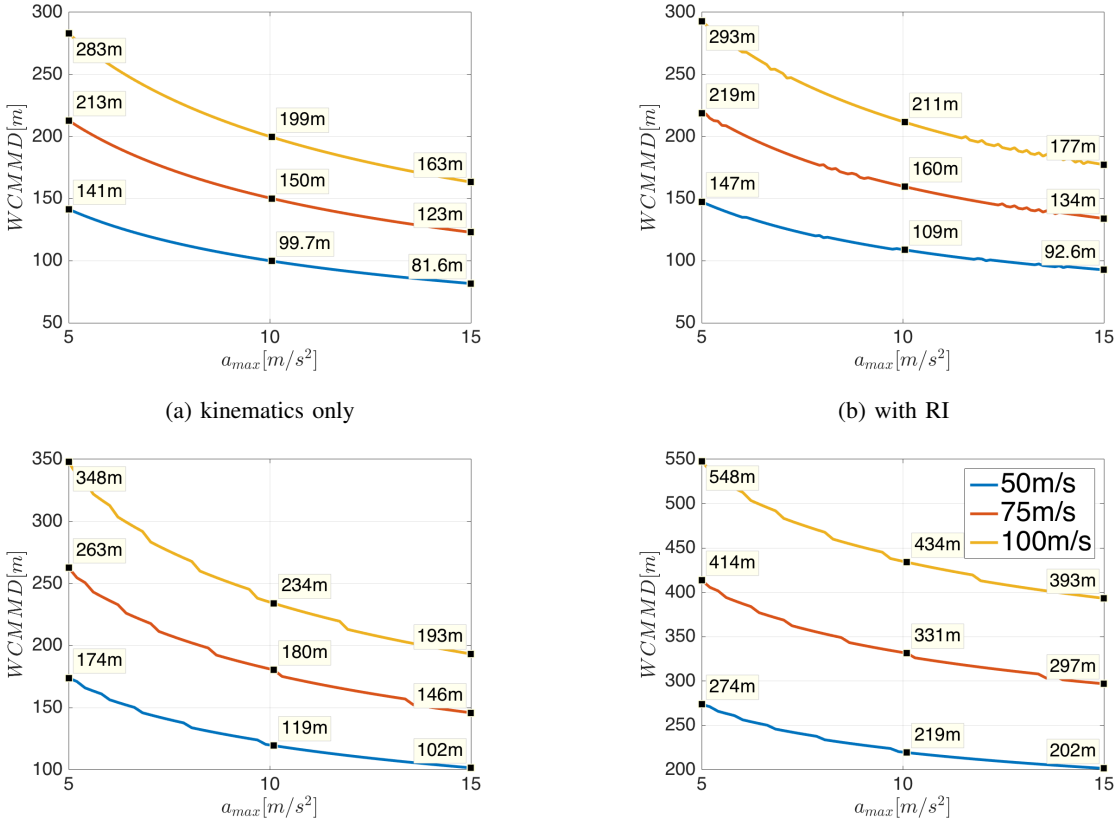
2) worst-case scenarios: In this section, we discuss the simulation results of the safety controller considered in Section IV. The WCMMD and WCERT are observed for a range of maximum horizontal relative velocity v and maximum horizontal accelerations a_{max}. In these worst-case scenarios, the quadrotor and helicopter are heading directly to each other. The effect of rotational inertia (RI), drag forces, and communication delay described in Section V are evaluated.

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 $2mg$, which can generate a maximum horizontal acceleration of $17m/s^2$ when tilting at 60° 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 $70m/s$, while a quadrotor has a maximum cruise speed of $30m/s$. Therefore, we set the maximum horizontal relative speed v to be between 50 and $100m/s$.

For clarity, we only present WCMMD results for three relative speeds (Figure 4). Figure 4a shows the WCMMD result by using the horizontal safety controller on a kinematic quadrotor. It serves as the base of comparison. 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 $202m$ (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 $548m$. 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 $100m$ (30%) to $200m$ (60%).

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 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_{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 5b), the WCERT decreases as a_{max} increases. It says that as a_{max} gets larger, it becomes easier to



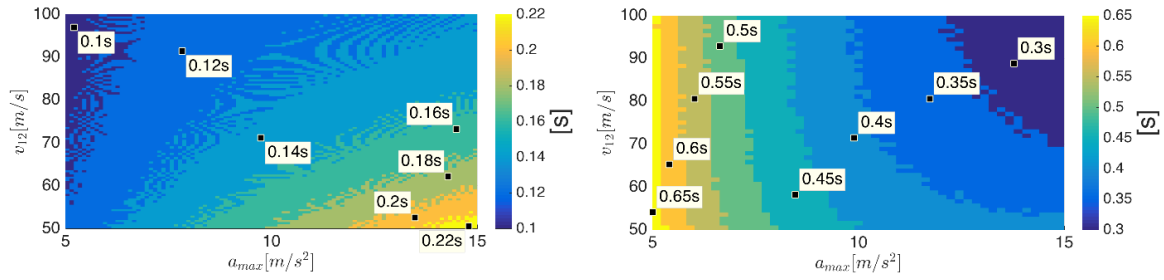
(a) kinematics only

(b) with RI

(c) with RI and drag

(d) with RI, drag, and comm. delay

Fig. 4: Summary of WCMMD for the horizontal safety controller, as defined in Section IV-A ($r_{min} = 20m$).



(a) with RI

(b) with RI and drag

Fig. 5: Summary of WCERT for the horizontal safety controller, as defined in Section IV-C ($r_{min} = 20m$).

overcome the drag, and the WCERT becomes smaller. To take into account the communication delay, one can simply add the delay ($2.0s$ for ADS-B) on top of Figure 5.

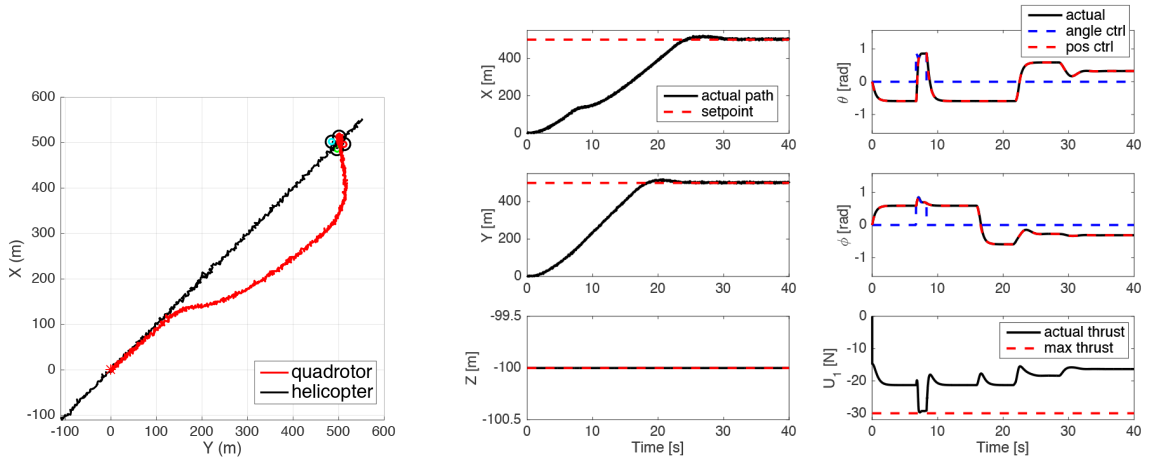
B. Hybrid Controller

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 $2mg$, which can give a maximum horizontal acceleration of $14m/s^2$ when hovering. Figure 6 shows a high-speed high-acceleration scenario. In Figure 6a, the quadrotor takes off from the origin and is trying to get to the point $(X,\ Y,\ Z) = (500m,\ 500m,\ \text{-}100m)$ 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 $\boldsymbol{V_{wind}}$ similar to [43]. In this scenario, the parameters are

$$a_{max} = 14m/s^2 \quad \boldsymbol{V_{wind}} = [\text{-}10,\ \text{-}10,\ 0]m/s$$
$$v_{heli} = 70m/s \qquad v_{quad} = 28.3m/s \qquad v = 98.3m/s$$

By employing the safety controller, the quadrotor successfully avoids the helicopter and gets to the final destination (Figure 6a). Assume that there is no communication delay, when the WCERT is set to $0.1s$, the minimum relative distance between the two vehicles is $21.8m$, 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 $2.0s$ communication delay, giving a WCERT value of $2.1s$.

Figure 6b shows the time history of the horizontal command and control signals. In the X, Y, and Z time history plots, the red dashed lines indicate desired destination, and the black solid lines give the actual flight trajectory. 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



(a) horizontal position trajectory

(b) time history of reference and control signals.

Fig. 6: The quadrotor avoids a helicopter on the way to a destination.

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 were 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 safety controller is presented in the context of high-speed manned-unmanned aircraft collision avoidance. A computationally feasible 2D optimal safety controller minimizing the avoidance duration is modified to be suitable for a 3D quadrotor model with rotational inertia and aerodynamic drag. The combined hybrid controller is capable of performing optimal avoidance when flying to a destination autonomously.

Compared to existing techniques, the proposed safety controller has two main advantages. First, it has a sense of optimality compared to [31], [32], [44], [45]. The controller is only enabled when necessary. Second, it is computationally feasible, which is hard to achieve with existing optimization techniques such as mixed-integer programming [36] and other numerical methods on solving the Hamilton-Jacobi equations [46].

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 [47]. Lastly, a hardware implementation is necessary to confirm its effectiveness in real life.

REFERENCES

[1] "Operation and Certification of Small Unmanned Aircraft Systems." Federal Aviation Administration, 2015, 14 CFR Parts 21, 43, 45, 47, 61, 91, 101, 107, and 183.

[2] P. Kopardekar, "Safely Enabling Low-Altitude Airspace Operations: Unmanned Aerial System Traffic Management (UTM)," 2015.

[3] Determining Safe Access with a Best Equipped, Best-Served Model for Small Unmanned Aircraft Systems. (Date last accessed 01-April-2016). [Online]. Available: http://utm.arc.nasa.gov/docs/Amazon_Determining%20Safe%20Access%20with%20a%20Best-Equipped,%20Best-Served%20Model%20for%20sUAS[2].pdf

[4] Google UAS Airspace System Overview. (Date last accessed 01-April-2016). [Online]. Available: http://utm.arc.nasa.gov/docs/GoogleUASAirspaceSystemOverview5pager[1].pdf

[5] Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems. (Date last accessed 01-April-2016). [Online]. Available: http://utm.arc.nasa.gov/docs/Amazon_Revising%20the%20Airspace%20Model%20for%20the%20Safe%20Integration%20of%20sUAS[6].pdf

[6] G. Li, S. P. Baker, J. G. Grabowski, and G. W. Rebok, "Factors associated with pilot error in aviation crashes," *Aviation, space, and environmental medicine*, vol. 72, no. 1, pp. 52–58, 2001.

[7] ART SYS 360. (Date last accessed 01-April-2016). [Online]. Available: http://www.artsys360.com/

[8] Hokuyo UTM-30LX. (Date last accessed 01-April-2016). [Online]. Available: https://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html

[9] PointGrey Bumblebee XB3. (Date last accessed 01-April-2016). [Online]. Available: https://www.ptgrey.com/support/downloads/10132

[10] B. Korn and C. Edinger, "Uas in civil airspace: Demonstrating sense and avoid capabilities in flight trials," in *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*. IEEE, 2008, pp. 4–D.

[11] M. Russ, M. Vohla, P. Stütz, and S. OYoung, "Lidar-based object detection on small uav: Integration, experimentation and results," in *AIAA Infotech Conference*, 2012.

[12] S. Hrabar, "3d path planning and stereo-based obstacle avoidance for rotorcraft uavs," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 807–814.

[13] T. Tomić, K. Schmid, P. Lutz, A. Dömel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 3, pp. 46–56, 2012.

[14] J. Byrne, M. Cosgrove, and R. Mehra, "Stereo based obstacle detection for an unmanned air vehicle," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2830–2835.

[15] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[16] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[17] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicles using dense stereo and optical flow," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2004, pp. 3718–3725.

[18] G. R. Rodríguez-Canosa, S. Thomas, J. del Cerro, A. Barrientos, and B. MacDonald, "A real-time method to detect and track moving objects (datmo) from unmanned aerial vehicles (uavs) using a single camera," *Remote Sensing*, vol. 4, no. 4, pp. 1090–1111, 2012.

[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.

[20] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a uav," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3309–3316.

[21] P. Cornic, P. Garrec, S. Kemkemian, and L. Ratton, "Sense and avoid radar using data fusion with other sensors," in *Aerospace Conference, 2011 IEEE*. IEEE, 2011, pp. 1–14.

[22] L. R. Salazar, R. Sabatini, S. Ramasamy, and A. Gardi, "A novel system for non-cooperative uav sense-and-avoid," in *proceedings of European Navigation Conference*, 2013.

[23] 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.

[24] Sagetech. [Online]. Available: http://www.sagetechcorp.com/

[25] "Airworthiness Approval of Automatic Dependent Surveillance - Broadcst(ADS-B) Out Systems." Federal Aviation Administration, 2012, AC 20-165A.

[26] K. Dar, M. Bakhouya, J. Gaber, M. Wack, and P. Lorenz, "Wireless communication technologies for its applications [topics in automotive networking]," *Communications Magazine, IEEE*, vol. 48, no. 5, pp. 156–162, 2010.

[27] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (gps) accuracy and reliability," *Journal of forestry*, vol. 103, no. 4, pp. 169–173, 2005.

[28] PING2020 uAvioni. [Online]. Available: http://www.uavionix.com/products/ping2020/

[29] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.

[30] J. G. Leishman and A. Bagai, "Challenges in understanding the vortex dynamics of helicopter rotor wakes," *AIAA journal*, vol. 36, no. 7, pp. 1130–1140, 1998.

[31] J.-W. Park, H.-D. Oh, and M.-J. Tahk, "Uav collision avoidance based on geometric approach," in *SICE Annual Conference, 2008*. IEEE, 2008, pp. 2122–2126.

[32] J. Krozel, T. Mueller, and G. Hunter, "Free flight conflict detection and resolution analysis," in *AIAA Guidance, Navigation, and Control Conf., Paper*, 1996, pp. 96–3763.

[33] P. Menon, G. Sweriduk, and B. Sridhar, "Optimal strategies for free-flight air traffic conflict resolution," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 202–211, 1999.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] F. P. Beer and E. Johnston, "Statics and dynamics," 1972.

[39] B. Song and J. K. Hedrick, *Dynamic surface control of uncertain nonlinear systems: an LMI approach*. Springer Science & Business Media, 2011.

[40] 3DR Solo Smart Drone. (Date last accessed 01-April-2016). [Online]. Available: http://3drobotics.com/solo-drone/

[41] B. R. Munson, D. F. Young, and T. H. Okiishi, *Fundamentals of fluid mechanics*. New York, 1990.

[42] A. Helfrick, *Principles of avionics*. Avionics Communications, 2010.

[43] 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.

[44] R. Y. Gazit and J. D. Powell, "Aircraft collision avoidance based on gps position broadcasts," in *Digital Avionics Systems Conference, 1996., 15th AIAA/IEEE*. IEEE, 1996, pp. 393–399.

[45] K.-Y. Kim, J.-W. Park, and M.-J. Tahk, "Uav collision avoidance using probabilistic method in 3-d," in *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*. IEEE, 2007, pp. 826–829.

[46] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *Automatic Control, IEEE Transactions on*, vol. 50, no. 7, pp. 947–957, 2005.

[47] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," *arXiv preprint arXiv:1503.07253*, 2015.