# An Assume-Guarantee Framework for Multiple-Obstacle Collision Avoidance

**Kaushik Nallan**
PhD student
Rensselaer Polytechnic Institute
Troy, NY, USA

**Sandipan Mishra**
Associate Professor
Rensselaer Polytechnic Institute
Troy, NY, USA

**A. Agung Julius**
Associate Professor
Rensselaer Polytechnic Institute
Troy, NY, USA

## ABSTRACT

In this paper, an assume-guarantee reasoning approach is developed for obstacle avoidance in unmanned aerial vehicles (UAVs) in the presence of multiple obstacles in an obstacle field. This construct assumes certain properties of the environment and the vehicle to guarantee the safety and performance of the UAV (in this case, executing safe collision-avoidance trajectories). In the presence of a single obstacle, the assumptions on the environment and the vehicle parameters are constructed such that the UAV can plan a safe trajectory once the obstacle is detected. The approach to guaranteeing safety in the presence of multiple obstacles requires enforcing additional assumptions which is done by constructing a region of influence (RoI) around each obstacle, whose size depends on the environment and the vehicle parameters. The safe combinations of these parameters (*codified as contracts*) are developed such that the RoIs in the obstacle field do not intersect. The aforementioned approach is then used to decompose the general multiple-obstacle avoidance problem into a sequential single-obstacle avoidance problem by constructing an induction-based algorithm framework. The proposed methodology is validated by an illustrative example with minimum obstacle detection range, maximum allowed cruise velocity, maximum allowable agility as vehicle properties; and maximum obstacle size and minimum obstacle separation as environmental properties. Contract generation for specific scenarios and implementation of sequential avoidance on a 6-DoF quadcopter simulation are demonstrated. Finally, the effect of tracking error on the contract-based framework is discussed, along with a mechanism to incorporate this source of uncertainty into the contract.

## NOTATION

$D_{od}$ - obstacle detection range
$D_{sep}$ - minimum obstacle separation in an obstacle field
$e_{tr}$ - tracking error between the actual position and commanded position
$E_{tr}$ - maximum tracking error in position
$\mathbb{E}$ - feasible set of maximum position tracking error
$\mathbf{F}$ - continuous chains of integrators for the differentially flat dynamics
$f_s$ - 6-DoF dynamics
$\mathbf{G}$ - continuous input matrices for the differentially flat dynamics
$G$ - safe region infront of the obstacle
$\bar{G}$ - unsafe region infront of the obstacle
$g$ - gravity
$\hat{g}$ - 'collision set boundary' is defined as $\hat{g}(x,y) = 0$
$J$ - cost function
$K$ - rotational inertia in $\psi$
$m$ - mass of the UAV
$\bar{\mathscr{O}}$ - set of the states of obstacles
$\mathscr{O}_i$ - state of the obstacle of interest '$i$'
$\mathbf{p}$ - 6-DoF states of the UAV (position, attitude and their derivatives)
$\mathbf{p}_0$ - initial state of the UAV
$\mathbb{P}_D$ - set of allowable final states of the UAV
$path$ - $x,y$ values of a trajectory
$\mathbf{q}$ - states of UAV in the differentially flat dynamics
$\mathbb{S}$ - set representing the 'region of influence'
$T$ - total available thrust
$t_0/t_f$ - initial/final times
$\mathbb{U}$ - feasible input set of the 6-DoF model
$\bar{\mathbb{U}}$ - feasible input set of the differentially flat model
$\mathbf{u}$ - inputs to the 6-DoF model
$\bar{\mathbf{u}}$ - inputs to the differentially flat model
$u_\psi$ - yaw moment
$\mathbb{V}$ - feasible set of velocity tracking error
$\hat{V}$ - actual velocity of the UAV while tracking
$V_{cruise}$ - maximum cruise velocity
$vel$ - velocity vectors along a trajectory
$\mathbb{W}$ - feasible input set of simplified differentially flat model
$\mathbf{w}$ - continuous simplified synthetic input
$\mathbf{X}$ - state of the simplified 2-D dynamics
$\mathbb{X}$ - state space of the simplified 2-D dynamics
$x_{pob}$ - $x$-coordinate of the post-obstacle border
$y_{l1}/y_{l2}$ - lane parameters
$x,y,z$ - planned position of the UAV in the inertial frame
$\hat{x},\hat{y},\hat{z}$ - actual UAV position while tracking
$\phi/\theta/\psi$ - roll/pitch/yaw angles
$\phi_{max}/\theta_{max}$ - maximum roll/pitch angles
$\delta V_x$ - control tracking error of velocity in $x$-direction
$\delta V_y$ - control tracking error of velocity in $y$-direction

Presented at the Vertical Flight Society's 77th Annual Forum & Technology Display, Palm Beach, Florida, USA, May 11–13, 2021.

## INTRODUCTION

With increasing use of unmanned aerial vehicles (UAVs) in military and civilian applications such as search and rescue, mapping, and exploration; it is important that UAV navigation is fully autonomous to reduce human errors. Since many applications involve traversing through environments cluttered with obstacles, developing a general framework for path-planning and control strategies (of an autonomous UAV) is challenging. Typically, the approaches for path-planning assume that the information regarding the environment (map) is known. For example, in (Ref. 1) the authors constructed an artificial potential field generated by the target (attracts) and the obstacles (repels) to achieve a feasible trajectory. Similarly, in approach (Ref. 2) the authors used Voronoi diagrams and Djikstra's algorithm to produce continuous curvature trajectories. Since information of the map might not be completely known a priori, the authors in (Ref. 3) assumed a probability map of obstacles and used the Bellman Ford algorithm to plan trajectories. When no information of the environment is provided, purely data driven approaches are typically used as shown in (Ref. 4), where the UAV is trained by an expert before hand. A major drawback of the above approaches is that they do not incorporate the dynamics of the UAV while planning, which might lead to collision while tracking the planned trajectory. The papers (Ref. 5) and (Ref. 6) plan the trajectory by incorporating vehicle dynamics but such approaches are generally limited to scenarios where the environment is known.

Even when obstacle information is explicitly given, it is crucial to be able to *guarantee the existence of a safe trajectory* before planning for time-efficient trajectories. Some approaches for guaranteeing safety involve computing the *Backward Reachable Set* (BRS) for a single obstacle, as discussed in (Ref. 7). This BRS is defined such that when the UAV enters this set, it will inevitably collide with the obstacle. Thus, as long as the UAV is outside the BRS, collision avoidance is possible. Although this framework can be applied independently on multiple obstacles, safety can not be guaranteed since avoiding one obstacle might lead to collision with another. Other approaches have focused on computing the BRS for multiple obstacles by solving the Hamilton-Jacobi-Isaac's equation (HJI) as shown in (Ref. 8). This approach can be computationally expensive and unusable especially when the exact obstacle sizes and locations are unknown.

In order to guarantee the existence of safe trajectories for path planning without explicit knowledge of the environment, this paper exploits *assume-guarantee reasoning*, wherein the overall performance of the system is guaranteed if individual subsystems guarantee certain properties of their own while assuming certain properties of other subsystems. Assume-guarantee reasoning has been used to obtain optimal control policy for traffic signals in (Ref. 9) and also for self-driving cars in (Ref. 10), where each car guarantees to follow an action structure while assuming similar guarantees from other vehicles. In this paper, we leverage this assume-guarantee reasoning for multiple-obstacle collision avoidance for UAVs. As an illustrative example for this paper, guarantees for safe obstacle avoidance are constructed by determining relationships between the maximum cruise velocity, agility of the UAV (available thrust, maximum allowed roll and pitch), controller performance (tracking error), navigation subsystem performance (obstacle detection range), and the nature of environment (e.g.,, maximum obstacle size, minimum separation distance between obstacles in the obstacle field).

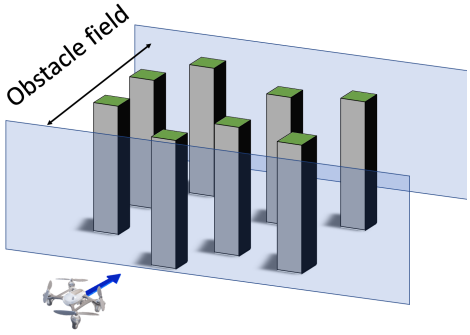



**Figure 1. General scenario for multiple obstacle avoidance**

## PROBLEM STATEMENT

Consider the scenario shown in Fig. 1, where a cruising UAV must fly through an obstacle field (with unknown obstacle locations *a priori*) to reach a target. The guarantee that UAV can safely fly through this obstacle field is contingent on the feasibility of the optimization problem in (1) as shown below

$$\begin{aligned} \min_{\mathbf{p}(\cdot),\mathbf{u}(\cdot)} \quad & J(\mathbf{p}(\cdot),\mathbf{u}(\cdot)), \\ s.t. \quad & \dot{\mathbf{p}}(t) = f_s(\mathbf{p}(t),\mathbf{u}(t)) \\ & \mathbf{p}(t_0) = \mathbf{p}_0, \quad \mathbf{p}(t_f) \in \mathbb{P}_D \\ & \mathbf{p}(\cdot) \notin \bar{\mathscr{O}}, \quad \mathbf{u} \in \mathbb{U}, \end{aligned} \tag{1}$$

where $\mathbf{p}$ denotes the state, $\mathbf{u}$ denotes the control input, $\mathbb{U}$ is the admissible set of inputs and $\bar{\mathscr{O}}$ is the set of obstacles. Note that the target ($\mathbb{P}_D$) is a set of permissible states or way-points beyond the obstacle field. However, confirming the existence of a feasible trajectory is not trivial due to two primary reasons. First, the non-linearity of the dynamics of the UAV (even with a simplified model) significantly increases the complexity of checking feasibility for all possible combinations of obstacle locations. Second, the feasible set for obstacle avoidance problem is in general non-convex. If we assume the obstacles $\mathscr{O}_i$ themselves to be convex, the safe region is the complement of the set $\bar{\mathscr{O}} = \mathscr{O}_1 \cup \mathscr{O}_2 \cup \ldots \mathscr{O}_n$, which is typically non-convex, resulting in weak or non-existent theoretical guarantees on the feasibility. Thus, there is no mechanism to obtain the necessary and sufficient conditions which determine the existence of a feasible trajectory, *a priori*.

**Guaranteeing Feasibility:** In this paper, we first develop a general framework (sufficient conditions) for guaranteeing

the feasibility of the multiple obstacle path planning problem (optimization) described in (1). The key idea explored here is to guarantee this feasibility by enforcing assumptions on the obstacles $\mathcal{O}_i$, the initial and terminal states ($\mathbf{p}_0$ and $\mathbb{P}_D$), the admissible control input set $\mathbb{U}$ as well as the dynamics $f_s$. For the scenario considered in this paper as shown in Fig. 2, these assumptions translate to relationships between the maximum cruise velocity $V_{cruise}$, obstacle detection range $D_{od}$, maximum obstacle size $S_{max}$ , minimum obstacle separation $D_{sep}$, vehicle agility (maximum thrust, maximum allowable roll and pitch) and maximum tracking error $E_{tr}$. The relationships between the above parameters such that (1) is feasible are 'contracts', which are developed in the later sections.

**Sequential Planning for Multiple Obstacle Avoidance:** If the feasibility of the optimization problem (1) is guaranteed, we propose a path-planning scheme for multiple obstacles, wherein the UAV determines the nearest 'obstacle of interest' (at any given time) and avoids obstacles sequentially. Naturally, this requires that the obstacles do not interfere with each other. That is, during the course of avoiding one obstacle, the UAV stays sufficiently far away from other obstacles, a condition guaranteed by the construction of the contract.

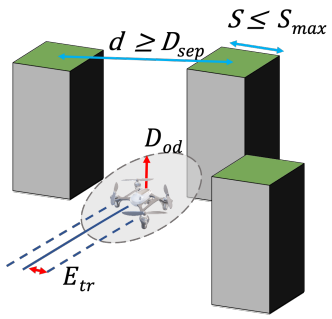


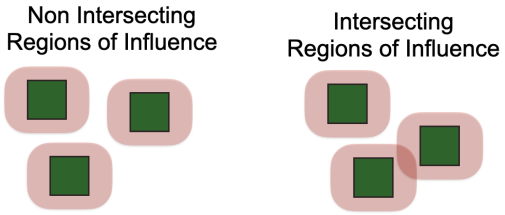

**Figure 2. Specific scenario for multiple obstacle avoidance**

## SOLUTION APPROACH: REGIONS OF INFLUENCE FOR OBSTACLES

In this section, we first discuss the idea behind the guaranteeing safety for a UAV cruising in the presence of a single obstacle and describe the limitations when the idea is directly extended to the multiple obstacle scenario. We then explore the key idea behind guaranteeing feasibility of the optimization problem (1) and the algorithmic approach to sequentially avoid multiple obstacles.

We first present the scenario where a UAV is cruising towards its target in an unknown environment in the presence of a single obstacle with a known maximum size. As discussed in (Ref. 11), we enforce assumptions on the UAV parameters and environmental parameters such that the contract is satisfied. This contract is constructed numerically by determining the unsafe combinations of these parameters that lead to the infeasibility of optimization problem (1). Instead of computing the infeasible combinations for all possible scenarios, the approach in (Ref. 11) determines the 'worst case' scenario such that the feasible combinations corresponding to the worst case scenario remain feasible for all other scenarios. The worst case scenario in the presence of a single obstacle assumes that the UAV is cruising head-on towards the obstacle and that the obstacle is right outside the UAV's obstacle detection range. This approach is usually computationally expensive since set of feasible positions (region outside the obstacle) is non-convex, thereby, causing the conventional gradient-based algorithms to be unreliable while solving the optimization problem (1).

**Key Idea for Multiple Obstacle Avoidance**: In the multiple obstacle scenario, the set of feasible positions is also non-convex since the obstacle set (the union of convex obstacles) is non-convex, which renders the numerical approach ineffective. Further, determining the 'worst' case scenario (for the construction of contracts) is not straightforward since the effects of obstacles on the constraints in (1) are coupled. To illustrate this, consider the case where the initial state of the UAV is such that we can guarantee a safe trajectory in the presence of a single obstacle $\mathcal{O}_1$. In a second scenario, let the UAV have the same initial state as before and is such that we can guarantee a safe trajectory in the presence of an obstacle $\mathcal{O}_2$ (without $\mathcal{O}_1$). However, in the presence of both obstacles $\mathcal{O}_1$ and $\mathcal{O}_2$, we cannot guarantee the existence of a safe trajectory because trying to avoid one of the obstacle might lead to collision with the other. *Therefore, we now investigate the conditions on the location/size of obstacles such that 'nearby' obstacles do not interfere with the safety guarantees on the UAV while avoiding one obstacle.*




(a) Safe (b) Unsafe

**Figure 3. Schematic of the obstacles and their respective Regions of Influence (top view)**

We introduce the notion of a **Region of Influence** (RoI) around the obstacles, as shown in Fig. 3, such that for any state of the UAV outside or on the boundary of this region, there exists a safe trajectory (that avoids collision with the obstacle) that stays inside the region until the obstacle is completely avoided. If the RoIs of individual obstacles are non-intersecting, we can isolate the effect of obstacles from each other. This way, we can address the multiple obstacle avoidance problem by breaking it down into a sequence of sub-problems, each of which deals with avoiding a single obstacle. If the RoI can be constructed such that the above statements hold true, the feasibility of the optimization problem

is guaranteed as long as the RoIs of obstacles do not intersect with each other. In order to reduce conservativeness, the RoI should be as tight (small) as possible, while retaining its guarantee.

## PRELIMINARIES

In this section, we briefly describe a rigid body dynamics UAV model and derive the simplified dynamics used for trajectory planning and contract generation. For a general 6-DoF quadcopter model as described in (Ref. 12), the system states are inertial position (North-East-Down frame), Euler angles, and their respective derivatives. However, since the pitch ($\phi$) and roll ($\theta$) are regulated $5-10$ times faster than the other states by the inner loop as discussed in (Ref. 12) and (Ref. 13), this paper treats them as control inputs for trajectory planning in the outer loop. The states for this model are position, velocity, yaw angle and yaw rate ($\mathbf{p} = [x\ \dot{x}\ y\ \dot{y}\ z\ \dot{z}\ \psi\ \dot{\psi}]^T$), while the control inputs are thrust, roll and pitch angles and yawing moment ($\mathbf{u} = [T\ \phi\ \theta\ u_\psi]^T$). The non-linear dynamics (as derived in (Ref. 12)) can be expressed as $\dot{\mathbf{p}} = f_s(\mathbf{p}, \mathbf{u})$ and $f_s$ is such that the relations in Eq. (2) are satisfied.

$$\begin{aligned}
\ddot{x} &= -\tfrac{T}{m}(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi);\\
\ddot{y} &= -\tfrac{T}{m}(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi);\\
\ddot{z} &= g - \tfrac{T}{m}\cos\phi\cos\theta;\\
\ddot{\psi} &= k u_\psi.
\end{aligned} \tag{2}$$

Converting this into its differentially flat form (Ref. 11), we obtain an equivalent linear dynamics and the original inputs as an explicit function of synthetic inputs as shown in Eq. (3) and Eq. (4), where $\mathbf{q}$, $\bar{\mathbf{u}}$ is the new state and synthetic control input, respectively.

$$\begin{aligned}
&\dot{\mathbf{q}} = \mathbf{F}\mathbf{q} + \mathbf{G}\bar{\mathbf{u}};\\
&\mathbf{q} = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \psi, \dot{\psi}]^T;\\
&\bar{\mathbf{u}} = [\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\psi}]^T,
\end{aligned} \tag{3}$$

where $\mathbf{F}$ is a continuous chain of integrators and the endogenous relations are,

$$\begin{aligned}
\phi &= \tan^{-1}\left(\tfrac{-\ddot{x}\sin\psi + \ddot{y}\cos\psi}{\sqrt{(g-\ddot{z})^2 + (\ddot{x}\cos\psi + \ddot{y}\sin\psi)^2}}\right);\\
\theta &= -\tan^{-1}\left(\tfrac{\ddot{x}\cos\psi + \ddot{y}\sin\psi}{g-\ddot{z}}\right);\\
T &= m\sqrt{\ddot{x}^2 + \ddot{y}^2 + (g-\ddot{z})^2};\\
u_\psi &= \tfrac{1}{k}\ddot{\psi}.
\end{aligned} \tag{4}$$

The synthetic input $\bar{\mathbf{u}}$ must be chosen such that the constraints on the original inputs are satisfied. To this end, the relationships in Eq. (2) can be used to map the admissible set $\mathbb{U}$ for the original input to the admissible set $\bar{\mathbb{U}}$ for the synthetic input, as shown in Fig. 4. Note that the equivalent linear dynamics along with the admissible synthetic input set is an *exact* transformation of the original system.

**Simplified Model for Contract Generation**: Although the planned trajectories are typically for a 6-DoF rigid body model of the UAV, developing guarantees on feasibility of (1) can be done by simplifying the dynamics conservatively. This has been done such that for any given initial state, trajectories generated using the simplified dynamics are feasible for the 6-DoF dynamics. To simplify the dynamics, we first eliminate the dependence on $z$ by constraining $\bar{U}$ such that $\ddot{z} = 0$, which is justified since obstacles in this paper are assumed to be pillar-like. Further simplifying by assuming $\psi = 0$ eliminates the dependence on the heading angle ($\psi$) leads to the dynamics in Eq. (5) (double integrator chains) as shown

$$\dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 \end{bmatrix}\mathbf{X} + \begin{bmatrix} 0 & 0\\ 1 & 0\\ 0 & 0\\ 0 & 1 \end{bmatrix}\mathbf{w}, \tag{5}$$

where $\mathbf{X} = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T \in \mathbb{X}$ is the simplified state and $\mathbf{W} = \begin{bmatrix} \ddot{x} & \ddot{y} \end{bmatrix}^T \in \mathbb{W}$ is the simplified synthetic control input. Here, $\mathbb{X}$ and $W$ are the state space and simplified synthetic input set respectively. Substituting $\ddot{z} = 0$ and $\psi = 0$ in Eq. (2), we obtain the endogenous relationships between simplified synthetic input and the original input as shown in Eq. (6) below.

$$\begin{aligned}
\ddot{x} &= -g\tan\theta,\\
\ddot{y} &= g\tan\phi\sec\theta.
\end{aligned} \tag{6}$$

For box-constraints on roll and pitch angles (i.e. $|\phi| \le \phi_{max}$ and $|\theta| \le \theta_{max}$), the simplified synthetic input set is approximated to be an axis-aligned rectangle as shown in Fig. 4. The feasible set of simplified synthetic input $\mathbb{W}$ can be defined as shown in Eq. (7) below.

$$\begin{aligned}
\mathbb{W} = \Big\{ (\ddot{x}, \ddot{y}) \;\Big|\; &\ddot{x} \in [-g\tan\theta_{max}, g\tan\theta_{max}],\\
&\ddot{y} \in [-g\tan\phi_{max}\sec\theta_{max}, g\tan\phi_{max}\sec\theta_{max}] \Big\}
\end{aligned} \tag{7}$$

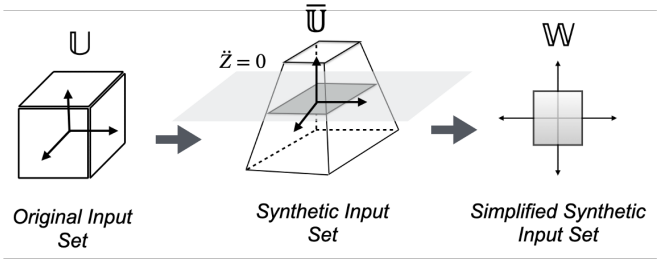



**Figure 4. Determining the simplified synthetic input set**

Although Eq. (7) shows that the bounds on the simplified synthetic input bounds are independent of $T_{max}$, the assumption of instantaneously achieving the desired roll and pitch angles is contingent on the value of $T_{max}$. Therefore, the synthetic input space $\mathbb{W}$ indirectly depends on $T_{max}$ (explicit relationship is not discussed in this paper).

## CONSTRUCTION OF REGION OF INFLUENCE

In this section, we use the simplified dynamics shown in Eq. (5) to construct the RoI for an obstacle such that: (**a**) Once the UAV (which is cruising towards the obstacle) enters this region, it must initiate a maneuver to avoid the obstacle, (**b**) while executing this maneuver to avoid the obstacle, it should stay inside the RoI, and (**c**) after the obstacle has been avoided, the UAV should be able to exit this region with its initial cruise velocity. Here, we elaborate on the necessary elements needed to construct the RoI as shown in Fig. 5. First, we have the *Collision Set Boundary* (CSB) determining the front boundary of the RoI. As discussed in the previous section, once the UAV crosses CSB with cruise velocity, it *must* execute an obstacle avoidance maneuver. Second, we construct a *lane* such the UAV stays inside the lane during the course of this maneuver. Finally, we construct a *Post Obstacle Border* (POB) such that if the UAV successfully executes the avoidance maneuver and stays in the lane, it should be able to exit the POB with its initial cruise velocity. Before we state the expressions for CSB, lanes and POB, we state the following theorem, which will then be used to obtain the analytical expressions of these elements.

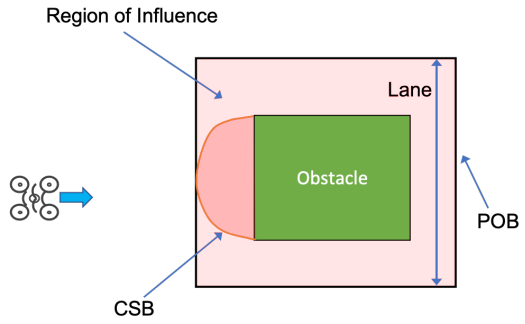



**Figure 5. Schematic of the region of influence (RoI)**

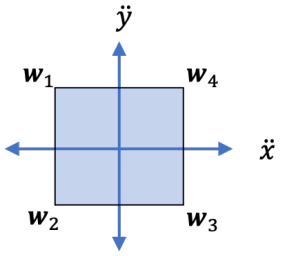



**Figure 6. Simplified synthetic input bounds**

**Theorem 1**. Consider an axis-aligned rectangular obstacle $\mathscr{O} = \{[x \;\; y]^T \mid x \in [x_{min}, x_{max}] \;,\; y \in [y_{min}, y_{max}]\}$ as shown in Fig. 5. For the dynamics shown in Eq. (5), initial state $\mathbf{X}_0 = [x \; \dot{x} \; y \; 0]^T$ and simplified synthetic input space $\mathbb{W}$ defined in Eq. (7), if applying the *constant* input $\mathbf{w}_i(\cdot)$ (input corresponding the $i^{th}$ vertex of $\mathbb{W}$ such that $\mathbf{w}(t) = \mathbf{w}_i, \forall t \geq 0$) as shown in Fig. 6 results in collision with $\mathscr{O}$ for every $i \in \{1,2,3,4\}$, then there *does not exist* any admissible input $\mathbf{w}(\cdot)$ (i.e., input signal where $\mathbf{w}(t) \in \mathbb{W}$, $\forall t \geq 0$) that avoids collision.

**Corollary**: As a consequence of Theorem 1, if there exists a feasible (safe) control input (that avoids collision), then the *constant input* $\mathbf{w}_i(t) = \mathbf{w}_i$ for some $i \in \{1,2,3,4\}$ (i.e., one of the vertices) also corresponds to a safe trajectory that avoids collision.

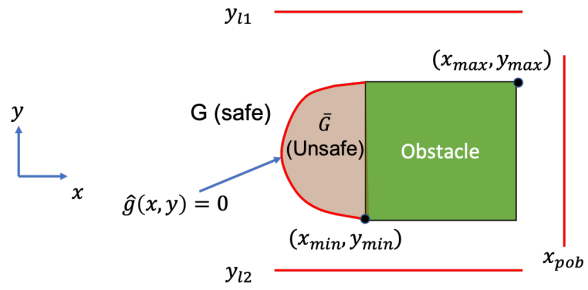



**Figure 7. Illustration of the parameters of CSB ($\hat{g}$), lane ($y_{l1}$, $y_{l2}$) and POB ($x_{pob}$)**

**Collision Set Boundary (CSB)** Here, we provide safety guarantees for a cruising UAV in the presence of a single obstacle by computing the initial states leading to inevitable collision with the obstacle (unsafe region). We then obtain the boundary (represented by an equation) separating the safe and unsafe regions (CSB), which is used to determine the RoI around the obstacle.

First, we define 'path' as a function $path : \mathbb{R}_{\geq 0} \times \mathbb{X} \times \mathbb{W}^{\infty} \to \mathbb{R}^2$, whose inputs are time, initial state and a control input signal and the output is the position. Similarly, we define $vel : \mathbb{R}_{\geq 0} \times \mathbb{X} \times \mathbb{W}^{\infty} \to \mathbb{R}^2$ whose output is the velocity vector. Note that $\mathbb{W}^{\infty}$ is the set of all admissible input signals.

Given the maximum cruise velocity, we define the safe region $G$ as the set of initial positions such that there exists a control input signal $\mathbf{w}(\cdot) \in \mathbb{W}^{\infty}$ that can avoid collision. Note that for analysis of safe and unsafe sets, we are only interested in the initial states that are 'before' the obstacle, i.e. $x \leq x_{min}$. $G$ can thus be described as below:

$$G = \Bigg\{ (x,y) \in \mathbb{R}^2 \;\Big|\; \exists\, \mathbf{w}(\cdot) \in W_{\infty} \; s.t \;\; \forall t \geq 0 \;\; \forall v \in [0, V_{cruise}], \; path\Big(t, [x \; v \; y \; 0]^T, \mathbf{w}(\cdot)\Big) \notin \mathscr{O}, \;\; x \leq x_{min} \Bigg\},$$

The unsafe set $\bar{G}$ is the complement of the safe set $G$,

$$\bar{G} = \Bigg\{ (x,y) \in \mathbb{R}^2 \;\Big|\; \exists t \geq 0, \; \exists v \in [0, V_{cruise}] \; s.t \;\; \forall \mathbf{w}(\cdot) \in W_{\infty}, \; path\Big(t, [x \; v \; y \; 0]^T, \mathbf{w}(\cdot)\Big) \in \mathscr{O}, \;\; x \leq x_{min} \Bigg\},$$

As shown in Fig. 7, there exists a boundary $\hat{g}(x,y)=0$ which separates the safe and unsafe regions, where the function $\hat{g}:\mathbb{R}^2\rightarrow\mathbb{R}$ is such that if $(x,y)\in G$, then $\hat{g}(x,y)>0$ and if $(x,y)\in\bar{G}$, then $\hat{g}(x,y)\leq 0$.

For the initial conditions of the UAV mentioned above (where $x\leq x_{min}$ and $\dot{x}\geq 0$), if applying constant input $\mathbf{w}_1(\cdot)$ and $\mathbf{w}_2(\cdot)$ leads to collision, applying $\mathbf{w}_3(\cdot)$ or $\mathbf{w}_4(\cdot)$ will result in collision with the obstacle (proof not discussed). Therefore, from **Theorem** 1 and the assumption on the initial conditions, we can conclude that the initial position lies in the unsafe region ($\bar{G}$) only if applying $\mathbf{w}_1(\cdot)$ or $\mathbf{w}_2(\cdot)$ throughout leads to collision with the obstacle. Therefore, we can express our safe region as $\bar{G}=\bar{G}_1\cap\bar{G}_2$. Here, $\bar{G}_i$ is the set of all initial positions such that applying constant input $\mathbf{w}_i(\cdot)$ leads to collision with the obstacle.

$$\bar{G}_1=\Bigg\{(x,y)\in\mathbb{R}^2\;\Big|\;\exists t\geq 0,\ \exists v\in[0,V_{cruise}]\ s.t.\ path\Big(t,[x\ v\ y\ 0]^T,\mathbf{w}_1(\cdot)\Big)\in\mathscr{O},\quad x\leq x_{min}\Bigg\},$$

$$\bar{G}_2=\Bigg\{(x,y)\in\mathbb{R}^2\;\Big|\;\exists t\geq 0,\ \exists v\in[0,V_{cruise}]s.t.\ path\Big(t,[x\ v\ y\ 0]^T,\mathbf{w}_2(\cdot)\Big)\in\mathscr{O},\quad x\leq x_{min}\Bigg\}.$$

Each of these unsafe sets ($\bar{G}_1$ and $\bar{G}_2$) have their corresponding complementary safe sets ($G_1$ and $G_2$), which can be defined accordingly. Therefore, if the initial position is in $G_1$ (or $G_2$), there exists a trajectory that avoids the obstacle from above (or below), that is, at $x=x_{min}$, $y\geq y_{max}$ (or $y\leq y_{min}$).

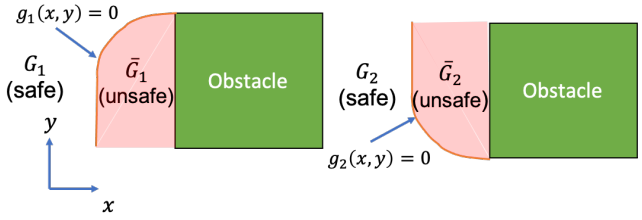
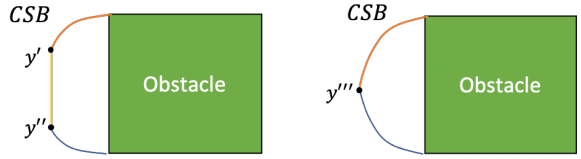



**Figure 8. Schematic of the functions which are used to construct the CSB**

As shown in Fig. 8, there exists a boundary $g_1(x,y)=0$ which separates the sets $G_1$ and $\bar{G}_1$. To obtain this, we first define the function $g_1:\mathbb{R}^2\rightarrow\mathbb{R}$ such that if $(x,y)\in G_1$, then $g_1(x,y)>0$ and if $(x,y)\in\bar{G}_1$, then $g_1(x,y)\leq 0$. Similarly, we have another boundary $g_2(x,y)=0$ which separates the sets $G_2$ and $\bar{G}_2$, where $g_2$ can be defined similar to $g_1$. The expressions for $g_1$ and $g_2$ are derived and discussed in the Appendix. Since the initial positions $(x,y)$ of the cruising UAV defining the unsafe region $\hat{g}(x,y)\leq 0$ is the intersection of the regions $g_1(x,y)\leq 0$ and $g_2(x,y)\leq 0$, the function $\hat{g}$ can be defined for $y\in(y_{min},y_{max})$ as

$$\hat{g}(x,y)=max\big(g_1(x,y),g_2(x,y)\big) \tag{8}$$

Depending on the maximum cruise velocity ($V_{cruise}$), maximum obstacle size ($S_{max}$) and simplified synthetic input space ($\mathbb{W}$), the above approach for computing the CSB ($\hat{g}(x,y)=0$) can result in two possible shapes of the boundary, as shown in Fig. 9.




(a) CSB corresponding lower cruise velocities

(b) CSB corresponding to higher cruise velocities

**Figure 9. Two possible shapes of the CSB**

The CSB takes the shape in Fig. 9(a) (as discussed in the Appendix) if the inequality in Eq. (9) is satisfied. In all other cases the CSB takes the shape shown in Fig. 9(b).

$$V_{cruise}\leq|\mathbf{w}_{1,x}|\sqrt{\frac{y_{max}-y_{min}}{|\mathbf{w}_{1,y}|+|\mathbf{w}_{2,y}|}}. \tag{9}$$

The analytical expressions for $y'$ and $y''$ corresponding to the CSB shown in Fig. 9(a) are shown below (derived in the Appendix).

$$y'=y_{max}-\frac{1}{2}|\mathbf{w}_{1,y}|\Big(\frac{V_{cruise}}{|\mathbf{w}_{1,x}|}\Big)^2,$$
$$y''=y_{min}+\frac{1}{2}|\mathbf{w}_{2,y}|\Big(\frac{V_{cruise}}{|\mathbf{w}_{2,x}|}\Big)^2. \tag{10}$$

**Lane**: The 'lane' around the obstacle is constructed as shown in Fig. 7, such that for any initial position of a UAV cruising towards the obstacle there exists a trajectory that avoids collision with the obstacle *while staying in the lane*. In this subsection, we first define the lane mathematically using its parameters $y_{l1}$ and $y_{l2}$ and then derive the expressions for these parameters.

In order to guarantee the existence of the safe trajectory, it is assumed that the initial position of the UAV $(\alpha,\beta)$ is inside the safe region i.e $(\alpha,\beta)\in G$. The goal is to obtain $y_{l1}$ and $y_{l2}$ such that for any $(\alpha,\beta)\in G$, there exists a control input signal $\mathbf{w}(\cdot)\in\mathbb{W}^\infty$ that satisfies the following:

$$y_{l2}\leq path_y\Big(t,[\alpha\ v\ \beta\ 0]^T,\mathbf{w}(\cdot)\Big)\leq y_{l1},\quad path\Big(t,[\alpha\ v\ \beta\ 0]^T,\mathbf{w}(\cdot)\Big)\notin\mathscr{O}, \tag{11}$$

for all time $t \geq 0$ and all $v \in [0, V_{cruise}]$. Here, the output of the function $path_y : \mathbb{R}_{\geq 0} \times \mathbb{X} \times \mathbb{W}_\infty \to \mathbb{R}$ is the y-component of the trajectory. The expressions for the parameters $y_{l1}$ and $y_{l2}$ (derived in Appendix) are shown below. Note that the expressions for lane parameters depend on the shape of the CSB.

$$y_{l1} = \begin{cases} y_{max} + \left|\frac{\mathbf{w}_{1,y}}{\mathbf{w}_{2,y}}\right|(y_{max} - y') & If\ \ Eq.(9) \text{ holds} \\ y_{max} + \left|\frac{\mathbf{w}_{1,y}}{\mathbf{w}_{2,y}}\right|(y_{max} - y''') & \text{otherwise} \end{cases} \tag{12}$$

$$y_{l2} = \begin{cases} y_{min} - \left|\frac{\mathbf{w}_{2,y}}{\mathbf{w}_{1,y}}\right|(y'' - y_{min}) & If\ \ Eq.(9) \text{ holds} \\ y_{min} - \left|\frac{\mathbf{w}_{2,y}}{\mathbf{w}_{1,y}}\right|(y''' - y_{min}) & \text{otherwise} \end{cases} \tag{13}$$

**Post Obstacle Border (POB)**: In the previous subsections, we have defined CSB and lane for a given obstacle such that when the UAV is outside the CSB, there exists a trajectory such that the UAV avoids the obstacle while staying in the lane. Here, we define the post obstacle border by its parameter $x_{pob}$ such that when the UAV follows the trajectory mentioned above, it should cross the obstacle with its initial cruise velocity.

For similar initial conditions as in the previous subsection $(\alpha, \beta) \in G$, we would like to obtain $x_{pob} \in \mathbb{R}$ such that for any $(\alpha, \beta) \in G$, there always exists a control input signal $\mathbf{w}(\cdot) \in \mathbb{W}^\infty$ and a time $t' \geq 0$ that satisfies all of the equations in Eq. (14).

$$\begin{aligned} path_x\Big(t', [\alpha\ v\ \beta\ 0]^T, \mathbf{w}(\cdot)\Big) &= x_{pob} \\ vel_x\Big(t', [\alpha\ v\ \beta\ 0]^T, \mathbf{w}(\cdot)\Big) &= v \\ y_{l2} \leq\ path_y\Big(t, [\alpha\ v\ \beta\ 0]^T, \mathbf{w}(\cdot)\Big) &\leq y_{l1} \\ path\Big(t, [\alpha\ v\ \beta\ 0]^T, \mathbf{w}(\cdot)\Big) &\notin \mathscr{O}, \end{aligned} \tag{14}$$

for all $t \in [0, t']$ and any $v \in [0, V_{cruise}]$. Here, the output of the function $path_x : \mathbb{R}_{\geq 0} \times \mathbb{X} \times \mathbb{W}_\infty \to \mathbb{R}$ is the $x$-component of the trajectory. Also, the output of the function $vel_x : \mathbb{R}_{\geq 0} \times \mathbb{X} \times \mathbb{W}^\infty \to \mathbb{R}$ is the x-component of the velocity.
The parameter corresponding to POB is $x_{pob}$ and can be expressed (derivation in Appendix) as:

$$x_{pob} = max\left\{x_{min} + \frac{V_{cruise}^2}{2|\mathbf{w}_{1,x}|}\ ,\ x_{max}\right\}. \tag{15}$$

**Region of Influence**: We denote the RoI as $\mathbb{S} \subset \mathbb{R}^2$. For a given axis aligned rectangular obstacle, we can express $\mathbb{S}$ as shown in Eq. (16). Note that $conv(\cdots)$ represents the convex hull and $x_c = arg\ min_x\ g(y)$ .

$$\mathbb{S} = conv\big\{(x_c, y_{l2})\ ,\ (x_c, y_{l1})\ ,\ (x_{pob}, y_{l2})\ ,\ (x_{pob}, y_{l1})\big\} \tag{16}$$

To summarize, given the maximum cruise velocity $V_{cruise}$, maximum obstacle size $S_{max}$ and vehicle agility ($T_{max}$, $\phi_{max}$ and $\theta_{max}$), we can construct the RoI using the expressions derived above.

## CONSTRUCTION OF CONTRACTS

As mentioned earlier, our objective is to enforce assumptions (i.e., generate contracts) on UAV parameters ($V_{cruise}$, $\phi_{max}$, $\theta_{max}$, $D_{od}$) and the environmental parameters ($S_{max}$, $D_{sep}$) such that the optimization problem (1) is feasible, and thus a safe flight is assured. To illustrate the construction of such contracts, we consider single obstacle as well as multiple obstacle scenarios. Instead of solving (1) for each combination of parameters, we exploit the analytical constructions of RoI discussed in the previous section for obtaining the contracts. To determine the contract in the single obstacle scenario, we construct the CSB and ensure that the UAV does not cross the CSB before detecting the obstacle. In the presence of multiple obstacles, in addition to satisfying the single obstacle contract, we construct the RoI for each obstacle and ensure that the minimum inter-obstacle separation is such that no two RoIs in the obstacle field intersect.

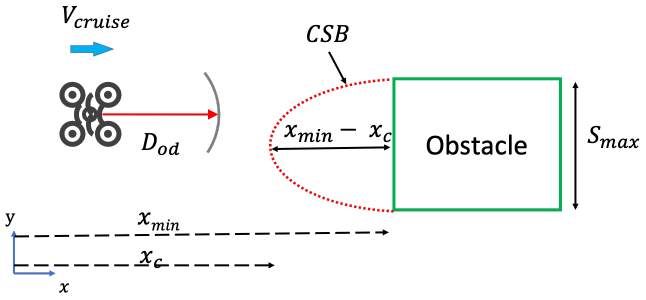



**Figure 10. Schematic of a UAV cruising in the presence of a single obstacle, whose maximum expected size is $S_{max}$**

### Single Obstacle Contracts

Consider the scenario as shown in Fig. 10, where the UAV is cruising in an environment in the presence of a single obstacle (unkown location) whose maximum expected size ($S_{max}$) is known. We wish to obtain a relationship between the obstacle detection range, agility and the maximum allowable cruise velocity such that the once the UAV detects the obstacle, it should be able to avoid collision. From the previous section, we can construct a CSB (which depends on $S_{max}$, $V_{cruise}$ and $\phi_{max}$) such that crossing the CSB will lead to inevitable collision with the obstacle. Therefore, for each combination of $V_{cruise}$ and $\phi_{max}$ (and $\theta_{max}$), we compute the CSB from Eq. (8) and force the minimum required $D_{od}$ to be equal to $x_{min} - x_c$. For a maximum expected obstacle size $S_{max} = 4m$, we obtain

the hypersurface that separates the safe and unsafe combinations as shown in Fig. 11. If the agility of the UAV is fixed, it can be observed that an increase in the obstacle detection range allows for greater maximum velocity. Similarly, for a fixed cruise velocity, reduction in the obstacle detection range must be compensated by increasing the agility of the UAV in order to remain safe.

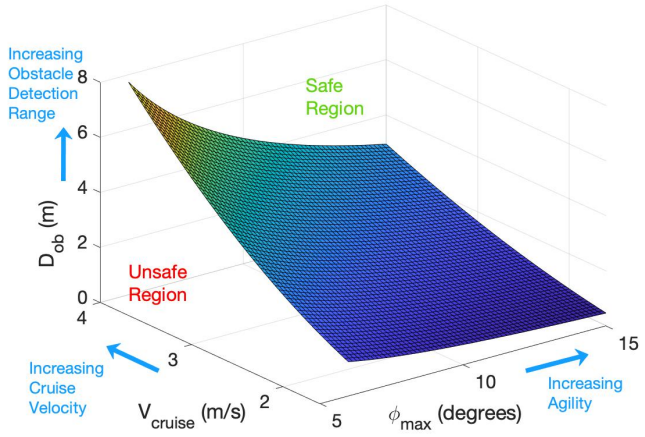



**Figure 11. Contract plot used for calculating the maximum cruise velocity $V_{cruise}$ to avoid a single obstacle in the presence of limited obstacle detection range and agility. Here, the maximum expected obstacle size $S_{max} = 4m$**

### Multiple Obstacle Contracts

When the UAV is cruising in an obstacle field in the presence of multiple obstacles, the UAV parameters need to satisfy additional contracts along with the single obstacle contract to guarantee its safety. An example of a multiple obstacle contract is the relationship between the maximum obstacle size $S_{max}$, maximum cruise velocity $V_{cruise}$ and minimum inter-obstacle separation $D_{sep}$ (center-center distance as shown in Fig. 2) such that the cruising UAV not only avoids the detected obstacle but also avoids collision with all neighboring obstacles during the avoidance maneuver. To obtain this relationship, we construct RoIs around an obstacle for different combinations of $V_{cruise}$ and $S_{max}$ and for each combination, we calculate the minimum separation required between two such obstacles such that their ROIs do not intersect. For a given maximum agility $\phi_{max} = 15°$, we obtain the hypersurface separating the safe and unsafe regions as shown in Fig. 12. For a fixed maximum obstacle size, increase in the minimum separation between obstacles will permit a greater maximum cruise velocity. Similarly, when the minimum separation distance is fixed, increase in the expected obstacle size will lower the maximum cruise velocity in order to guarantee safety.

Assuming that the single obstacle contract is satisfied and the maximum expected obstacle size is fixed, we can obtain the relation between minimum obstacle separation, maximum

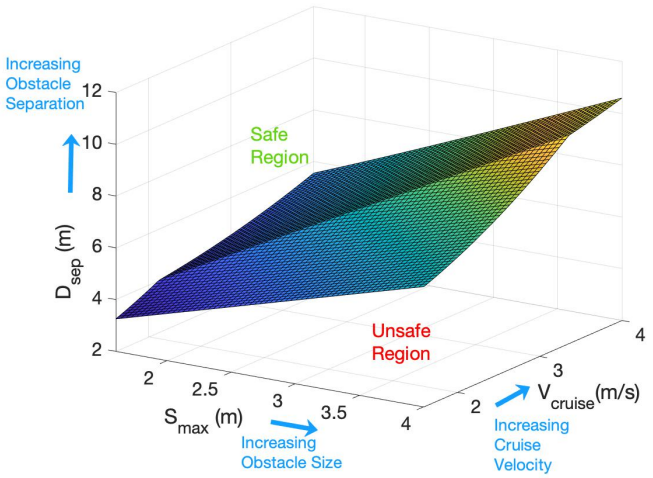



**Figure 12. Contract between minimum obstacle separation $D_{sep}$, obstacle size $S_{max}$ and vehicle cruise velocity $V_{cruise}$. Here $\phi_{max} = 15°$**

cruise velocity and agility of the UAV. We can see from Fig. 13 that when the maximum cruise velocity is fixed, decrease in the agility of the UAV will require higher minimum obstacle separation in order to remain safe.

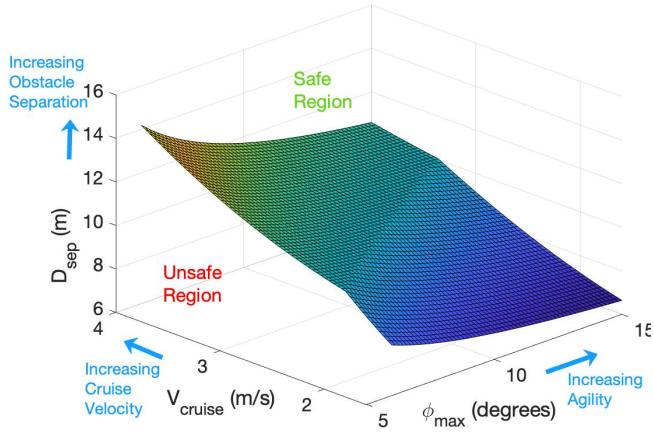



**Figure 13. Contract between minimum obstacle separation $D_{sep}$, vehicle cruise velocity $V_{cruise}$ and vehicle agility $\phi$. Here, the maximum expected obstacle sie($S_{max} = 4m$**

.

## SEQUENTIAL PLANNING FOR MULTIPLE OBSTACLES

Assuming that the parameters are such that the contracts (in the previous section) are satisfied i.e. RoIs do not intersect and $D_{od} \geq x_{min} - x_c$ , we present an approach to sequentially solve the multiple obstacle avoidance path planning problem. First, we define **Obstacle of Interest** $\mathcal{O}_i$ , the immediate obstacle in the vicinity of the UAV such that the UAV will collide with $\mathcal{O}_i$, if no control action is taken. Then, we define

the **pre-condition** on the UAV with respect to $\mathscr{O}_i$ such that the UAV is cruising towards the obstacle and is exactly on the left boundary of the RoI. Finally, we define the **post-condition** on the UAV with respect to $\mathscr{O}_i$ such that the UAV is cruising away from the obstacle and is on the POB.

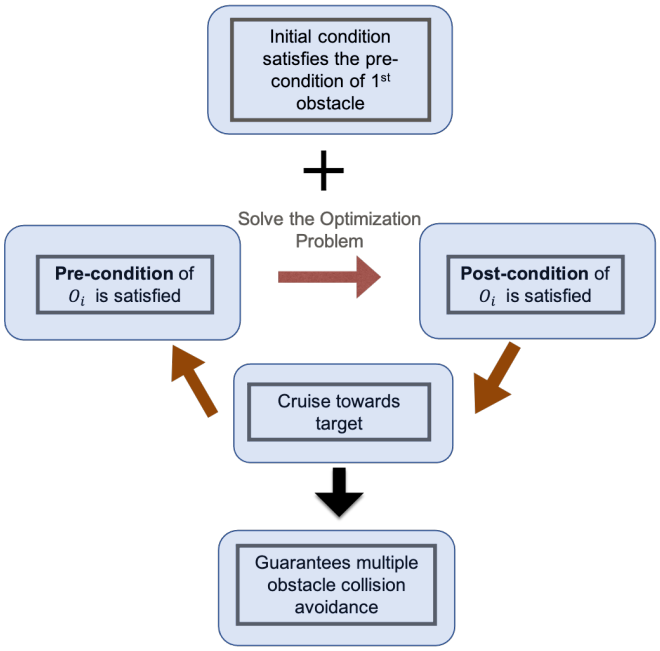



**Figure 14. Induction-based Algorithm for sequential-avoidance in an obstacle field**

$$
\begin{aligned}
\min_{\mathbf{q}(\cdot),w(\cdot)} \quad & J_i(\mathbf{q},\bar{\mathbf{u}}), \qquad (17)\\
s.t. \quad & \dot{\mathbf{q}} = \mathbf{F}\mathbf{q} + \mathbf{G}\bar{\mathbf{u}};,\\
& x(t_0) = (x_c)_i, \quad \dot{x}(t_0) \leq V_{cruise}\\
& \dot{y}(t_0) = 0, \quad y(t_0) \in [(y_{l2})_i, (y_{l1})_i]\\
& x(t_f) = (x_{pob})_i, \quad \dot{x}(t_f) = \dot{x}(t_0), \quad \dot{y}(t_f) = 0\\
& [x(t)\ y(t)]^T \in \mathbb{S}_i, \quad [x(t)\ y(t)]^T \notin \mathscr{O}_i, \quad \forall t \in [t_0, t_f]\\
& \bar{\mathbf{u}} \in \bar{\mathbb{U}}
\end{aligned}
$$

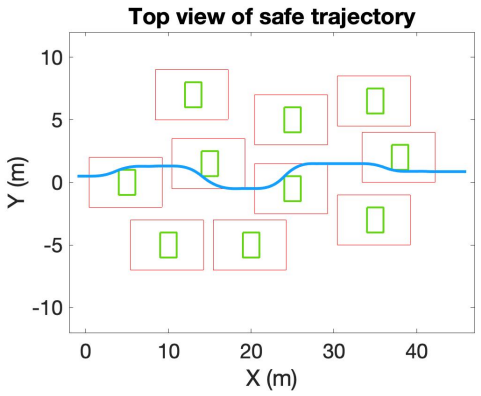
Given the maximum allowable velocity, if the initial state of the UAV satisfies the pre-condition of the first obstacle of interest $\mathscr{O}_1$, we can guarantee the existence of a safe trajectory such that the UAV avoids the obstacle $\mathscr{O}_1$, stays in the RoI of $\mathscr{O}_1$ and reaches the state which satisfies the post-condition of $\mathscr{O}_1$. Since we assume that there is no intersection between the RoIs, this trajectory would not intersect any other obstacle. Therefore, the UAV can follow this trajectory, reach the post-condition of $\mathscr{O}_1$ and continue to cruise until it satisfies the pre-condition of its next obstacle of interest. Using this induction argument (as illustrated in Fig. 14), the UAV can avoid multiple obstacles sequentially with guaranteed safety. To obtain the trajectory from the states satisfying pre-condition of $\mathscr{O}_i$ to the states satisfying post-condition of $\mathscr{O}_i$, we solve the optimization problem in (17), which constraints the trajectory to lie inside the RoI. Thus the optimization problem in (1) can be broken down into a sequence of single obstacle avoidance problems (17). Here, we represent the RoI of $\mathscr{O}_i$ as $\mathbb{S}_i$.

**Algorithm Implementation**: We illustrate the application of the sequential multiple obstacle avoidance algorithm developed in the previous section for *6-DoF path planning* for a UAV. We consider a scenario where the UAV (quadcopter) is cruising in the positive $x$-direction towards an obstacle field (shown as green boxes in Fig. 15(a)) and the obstacles are such that (1) the initial position is outside the CSB of every obstacle (2) RoI of any two obstacles (shown as red boxes) do not overlap. Therefore, we guarantee the existence of a safe trajectory and implement the induction scheme shown in Fig. 14 to obtain the trajectory. The initial position of the UAV is $[x\ y\ z] = [0m\ 0m\ -4m]$, with initial x-velocity being $2.8m/s$. The UAV plans a trajectory (shown in blue in Fig. 15(a)) to avoid the left-most obstacle by ignoring all other obstacles, then cruises in the x-direction with the same velocity. With the induction scheme implemented, it then encounters the center obstacle and the same process is repeated. Finally, after all three obstacles have been avoided, the final position of the UAV is $[x\ \ y\ \ z] = [46m\ \ 0.3m\ \ -4m]$. Figure 15(b) shows that the UAV gains altitude while trying to avoid the obstacle since increasing acceleration along $z$ increases maneuverability. Note that $z$ axis is pointed downwards in the *NED* frame.

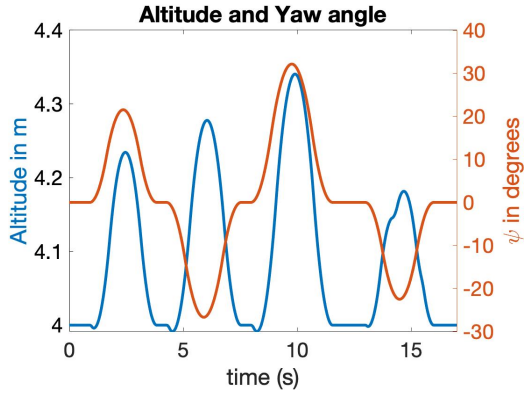
## INCORPORATING TRACKING ERROR INTO CONTRACTS

In the previous section, we guaranteed that the UAV can avoid multiple obstacles sequentially and planned safe trajectories by solving the optimization problem Eq. (17) for each obstacle of interest $\mathscr{O}_i$. However, these planned trajectories are generated assuming that UAV dynamics differentially flat as in Eq. (3), which is equivalent to the 6-DoF dynamics in Eq. (2). The dynamics in Eq. (3) assumes that the roll $\phi$ and pitch $\theta$ angles are instantly achieved. However, since the inner loop regulation/tracking is not perfect, there will always be some finite tracking error, depending on the quality of the trajectory tracking controller. Here, we first evaluate the performance of a given controller used to track the trajectory shown in Fig. 15 and later discuss the general idea behind incorporating error in position and error in velocity into the contract-based framework such that the implementation of the algorithm shown in Fig. 14 can still guarantee safety of the UAV.

The controller used to validate the planned trajectories in Fig. 15 uses dynamic inversion for a non-linear state space model of the quadcopter as discussed in (Ref. 14). The states are position, Euler angles and their respective derivatives. The control input is the rotational speed of the motor (RPM) and it is assumed that the thrust produced by the motor is proportional to the square of RPM. The controller was implemented on a simulation platform (MATLAB) and the actual trajectories were obtained as shown in Fig. 16, where we can see

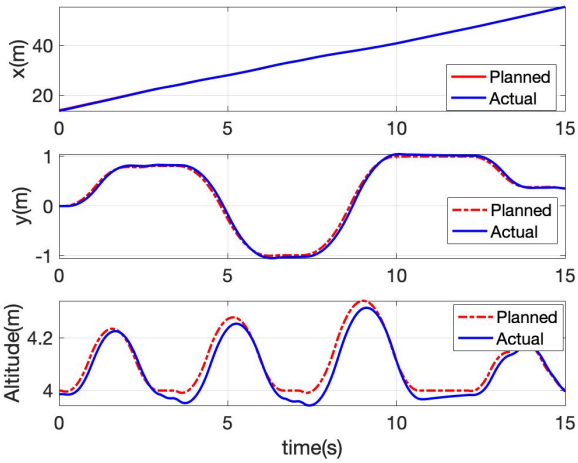



(a) Path of the planned trajectory

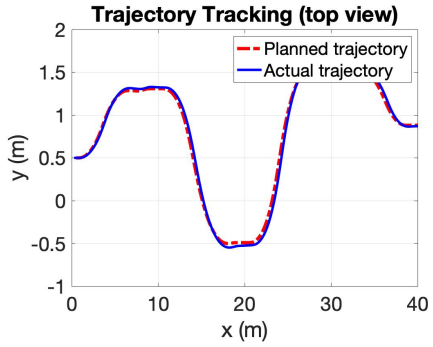



(b) Altitude and heading angle of the planned trajectory

**Figure 15. Implementation of the induction based algorithm in the 6-DoF planning where the maximum agility is $\frac{T_{max}}{m} = 2g$ and $\phi_{max} = 5°$. An additional constraint on $\ddot{\psi}$ is imposed for increasing smoothness in the resulting heading angle**




(a) Comparing the planned and actual trajectories




(b) The top view comparison between the planned and the actual path of the UAV (x-y plane)

**Figure 16. Results for trajectory tracking using non-linear Dynamic Inversion-based controller**

that the planned trajectories are tracked with reasonable precision. Since the simplified dynamics in Eq. (3) which has been used to plan trajectories assumes that the roll and pitch angles can be instantaneously achieved, we can conclude that this assumption is acceptable for planning purposes.

**Position Tracking Error**: For a given scenario, assuming that the tracking error for a particular controller is known, we define the position tracking error as a function of time $e_{tr}(t) = \sqrt{(x(t)-\hat{x}(t))^2 + ((y(t)-\hat{y}(t))^2}$, where $\hat{x}(t), \hat{y}(t)$ are the actual trajectory paths as shown in Fig. 17(a). Note that we assume $e_{tr}$ is computed for the reference trajectory corresponding to the poorest tracking. Now we can define the maximum tracking error for a planned trajectory by $E_{tr}$, where $E_{tr} = \max_t \ e_{tr}(t)$. Even if the planned trajectories do not intersect with the obstacle, existence of a non-zero $E_{tr}$ might result in collision of the UAV with the obstacle. Therefore, we would like to take consider the tracking error beforehand while planning trajectories. In this case, we plan the trajectories for an inflated virtual obstacle as shown in Fig. 17(b) such that even though the actual trajectory might intersect with the virtual obstacle, the UAV will remain safe since the actual trajectory will not be intersecting with the real obstacle. If the obstacle of interest in $\mathscr{O}_i$, the virtual obstacle ($\hat{\mathscr{O}}_i$) can be expressed as

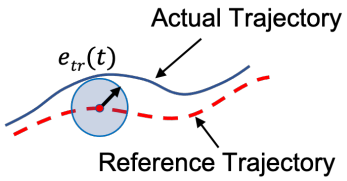
$$\hat{\mathscr{O}}_i = \mathscr{O}_i \bigoplus \mathbb{E}, \tag{18}$$

where $\mathbb{E} = \left\{ (x,y) \;\middle|\; x \in [-E_{tr}, E_{tr}],\ y \in [-E_{tr}, E_{tr}] \right\}$ and $\bigoplus$ denotes the Minkowski sum.

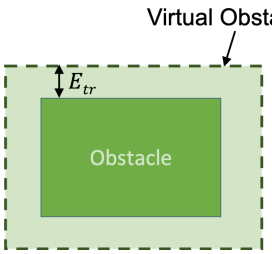
**Velocity Tracking Error**: For a given scenario and particular controller, let the maximum tracking error in velocity in the $x$-direction be $\delta V_x$ and $y$-direction be $\delta V_y$. Therefore, when the UAV is outside the RoIs, the actual velocity of the UAV ($\hat{\mathbf{V}}$) will satisfy $\hat{\mathbf{V}} \in [0, V_{cruise}] \bigoplus \mathbb{V}$, where $\mathbb{V} = \left\{ (v_x, v_y) \;\middle|\; v_x \in [-\delta V_x, \delta V_x],\ v_y \in [-\delta V_y, \delta V_y] \right\}$. If $\delta V_y$ is small, the velocity tracking error can be incorporated into the contract by constructing the RoI assuming the worst case scenario, where the actual cruise velocity is $\hat{\mathbf{V}} = (V_{cruise} + \delta V_x, 0)$. If the RoIs are intersecting, $V_{cruise}$ needs to be decreased such that the contract for the multiple obstacle scenario is satisfied.

## CONCLUSIONS

To guarantee the safety of a UAV passing through an obstacle field with unknown obstacles, the framework proposed




(a) Schematic of the position tracking error in the x-y plane where the actual position of the UAV at time $t$ can be anywhere in the blue region




(b) Schematic of the virtual obstacle (inflated) to compensate for position tracking error

**Figure 17. Incorporating position tracking error into the contract.**

in this paper enforces assumptions on the vehicle and environmental parameters, which are contracts. These contracts were obtained analytically by constructing a Region of Influences (RoI) around each obstacle such that obstacle field with non-intersecting RoIs guarantee safety of the UAV. Assuming that the contracts are satisfied, an induction-based algorithm is proposed such that the UAV can plan trajectories to avoided obstacles sequentially before reaching its target. The algorithm also guarantees that the each trajectory that is planned to avoid a particular obstacle completely lies inside its RoI. After showing that the planned trajectories can be tracked with reasonable accuracy in the presence of a controller, this paper proposes a methodology to incorporate the tracking error (both position and velocity) into the assume-guarantee framework.

Author contact: Kaushik Nallan nallak@rpi.edu, Sandipan Mishra mishrs2@rpi.edu and A. Agung Julius agung@ecse.rpi.edu

## APPENDIX

### Detailed derivation of CSB, lane and POB parameters

**Collision Set Boundary (CSB)**: Given an axis-aligned rectangular obstacle, we define $f_1 : \mathbb{R} \to \mathbb{R}$ such that if the UAV is initially on the curve $x = f_1(y)$ and is cruising with $V_{cruise}$, applying the constant control input $\mathbf{w}_1$ will result in the UAV reaching the point $(x_{min}, y_{max})$, which is the top corner of the obstacle. Using the double integrator dynamics of the system, we obtain the expression for $f_1$, which is

$$f_1(y) = x_{min} - V_{cruise}\sqrt{\frac{2(y_{max}-y)}{|\mathbf{w}_{1,y}|}} + \frac{|\mathbf{w}_{1,x}|}{|\mathbf{w}_{1,y}|}(y_{max}-y). \quad (19)$$

Next, we define $f_c : \mathbb{R} \to \mathbb{R}$ such that if the UAV is initially on the curve $x = f_c(y)$ and is cruising with $V_{cruise}$, applying the constant control input $\mathbf{w}_1$ (or $\mathbf{w}_2$) will result in the UAV reaching $x_{min}$ with $\dot{x} = 0$. Since the dynamics are decoupled in $x$ and $y$, $f_c$ is a constant function expressed as

$$f_c(y) = \frac{V_{cruise}^2}{2|\mathbf{w}_{1,x}|}. \quad (20)$$

Lastly, we define $y'$ such that the curves $f_1$ and $f_c$ intersect at $(\frac{V_{cruise}^2}{2|\mathbf{w}_{1,x}|}, y')$. If the UAV cruising from this point, applying $\mathbf{w}_1$ will not only result in reaching the point $(x_{min}, y_{max})$, but also reaches the point with $\dot{x} = 0$.

$$y' = y_{max} - \frac{1}{2}|w_{1,y}|\Big(\frac{V_{cruise}}{|w_{1,x}|}\Big)^2. \quad (21)$$

Now, $g_1$ can be expressed as a piecewise function of $f_1$ and $f_c$ as shown below.

$$g_1(x,y) = \begin{cases} x - f_1(y) & y' \le y \le y_{max} \\ x - f_c & y \le y' \end{cases} \quad (22)$$

Similarly, we define $f_2 : \mathbb{R} \to \mathbb{R}$ such that if the UAV is initially on the curve $x = f_2(y)$ and is cruising with $V_{cruise}$, applying the constant control input $\mathbf{w}_2$ will result in the UAV reaching the point $(x_{min}, y_{min})$, which is the bottom corner of the obstacle. The expression for $f_2$ is

$$f_2(y) = x_{min} - V_{cruise}\sqrt{\frac{2(y-y_{min})}{|\mathbf{w}_{2,y}|}} + \frac{|\mathbf{w}_{2,x}|}{|\mathbf{w}_{2,y}|}(y-y_{min}). \quad (23)$$

We define $y''$ such that the curves $f_2$ and $f_c$ intersect at $(\frac{V_{cruise}^2}{2|w_{1,x}|}, y'')$. If the UAV cruising from this point, applying $w_2$ will not only result in reaching the $(x_{min}, y_{min})$, but also reaches the point with $\dot{x} = 0$.

$$y'' = y_{min} + \frac{1}{2}|\mathbf{w}_{2,y}|\Big(\frac{V_{cruise}}{|\mathbf{w}_{2,x}|}\Big)^2. \quad (24)$$

Now, $g_2$ can be expressed as a piecewise function of $f_2$ and $f_c$ as shown below.

$$g_2(x,y) = \begin{cases} x - f_2(y) & y_{min} \le y \le y'' \\ x - f_c & y \ge y'' \end{cases} \quad (25)$$

The Collision Set Boundary (CSB) is then defined by $\hat{g}(x,y) = 0$, where for $y \in (y_{min}, y_{max})$, we have

$$\hat{g}(x,y) = max\big(g_1(x,y), g_2(x,y)\big). \quad (26)$$

From Eq. 22, Eq. 25 and Eq. 26, we can see that the shape CSB will depend on the relationship between $y'$ and $y''$. If $y' > y''$, the corresponding condition on $V_{cruise}$ is obtained using Eq. (21) and Eq. (24) which is

$$V_{cruise} \le |\mathbf{w}_{1,x}|\sqrt{\frac{y_{max}-y_{min}}{|\mathbf{w}_{1,y}|+|\mathbf{w}_{2,y}|}}. \quad (27)$$

If $y' < y''$, there will be $y'''$ which is obtained from the intersection of $f_1$ and $f_2$ by solving the equation below.

$$f_1(y''') = f_2(y'''). \tag{28}$$

**Lane**: Since applying no simplified synthetic control input will result in crossing the CSB before colliding with the obstacle, we obtain the expressions for the lane parameters by analysing trajectories starting from the CSB.

For any initial position $(\alpha, \beta)$ of the UAV lying on $x = f_1(y)$, applying $\mathbf{w}_1(\cdot)$ would lead to reaching the point $(x_{min}, y_{max})$ with $\dot{x} \geq 0$ and $\dot{y} \geq 0$. The upper bound of the lane $\bar{y}$ depends on the minimum y-value the UAV gains (after avoiding the obstacle) before $\dot{y}$ becomes 0. From the double integrator dynamics and the synthetic input bounds, we have

$$\bar{y}(\beta) = y_{max} + \left|\frac{\mathbf{w}_{1,y}}{\mathbf{w}_{2,y}}\right|(y_{max} - \beta). \tag{29}$$

Similarly, for any initial position $(\alpha, \beta)$ of the UAV lying on $x = f_2(y)$, applying $\mathbf{w}_2(\cdot)$ would lead to reaching the point $(x_{min}, y_{min})$ with $\dot{x} \geq 0$ and $\dot{y} \leq 0$. The lower bound of the lane depends on the minimum y-value the UAV gains in $-y$ direction before $\dot{Y}$ becomes 0 ($\underline{y}(\beta)$). The expression for $\underline{y}$ is

$$\underline{y}(\beta) = y_{min} - \left|\frac{\mathbf{w}_{2,y}}{\mathbf{w}_{1,y}}\right|(\beta - y_{min}). \tag{30}$$

If the initial position $(\alpha, \beta)$ of the UAV lying on $f_c$, applying a constant input of only $\mathbf{w}_{1,x}(\cdot)$ would lead to reaching the outer edge of the obstacle with $\dot{x} = 0$ and $\dot{y} = 0$. In this case, the UAV can avoid the obstacle with the upper bound of lane as $y_{max}$ and lower bound as $y_{min}$.

For all possible initial conditions $(\alpha, \beta)$, we can see, from (29) and (30), that the upper boundary of the lane depends on the smallest value of $\beta$ and the lower boundary depends on the largest value of $\beta$. Therefore, the lane parameters $y_{l1}$ and $y_{l2}$ can be expressed as

$$y_{l1} = \begin{cases} y_{max} + \left|\frac{w_{1,y}}{w_{2,y}}\right|(y_{max} - y') & If\ \ Eq.(9) \text{ holds} \\ y_{max} + \left|\frac{w_{1,y}}{w_{2,y}}\right|(y_{max} - y''') & \text{otherwise} \end{cases} \tag{31}$$

$$y_{l2} = \begin{cases} y_{min} - \left|\frac{w_{2,y}}{w_{1,y}}\right|(y'' - y_{min}) & If\ \ Eq.(9) \text{ holds} \\ y_{min} - \left|\frac{w_{2,y}}{w_{1,y}}\right|(y''' - y_{min}) & \text{otherwise} \end{cases} \tag{32}$$

**Post Obstacle Border:** It has been established that for any initial position of the UAV lying on $x = f_1(y)$ or $x = f_2(y)$ or $x = f_c$ would result in reaching $x = x_{min}$ with $\dot{x} \geq 0$. The parameter for POB is obtained such that there exists a safe trajectory starting from any $(\alpha, \beta)$ whose $\dot{x} = \dot{x}(t = 0) \leq V_{cruise}$ at $x_{pob}$. Therefore, we consider the worst case scenario where $\dot{x} = 0$ at $x = x_{min}$ and $\dot{x} = V_{cruise}$ at $x = x_{pob}$ to obtain the expression for $x_{pob}$ as shown

$$x_{pob} = x_{min} + \frac{V_{cruise}^2}{2|\mathbf{w}_{1,x}|}. \tag{33}$$

Since crossing the POB implies that the UAV leaves the RoI the obstacle (successful avoidance), we enforce an additional constraint, which is $x_{pob} \geq x_{max}$.

## ACKNOWLEDGMENTS

This work is carried out at the Rensselaer Polytechnic Institute under the Army/Navy/NASA Vertical Lift Research Center of Excellence (VLRCOE) program, grant number W911W61120012, with Dr. Mahendra Bhagawat and Dr. William Lewis as Technical Monitors.

## REFERENCES

1. Budiyanto, A., Cahyadi, A., Adji, T. B., and Wahyunggoro, O., "UAV obstacle avoidance using potential field under dynamic environment," 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 2015.

2. Candeloro, M., Lekkas, A., Sørensen, A., and Fossen, T., "Continuous Curvature Path Planning Using Voronoi Diagrams and Fermat's Spirals," IFAC Proceedings Volumes (IFAC-PapersOnline), Osaka, Japan, Vol. 9, 2013.

3. Jun, M., and D'Andrea, R., "Path Planning For Unmanned Aerial Vehicles In Uncertain and adversarial environments," *Cooperative Control: Models, Applications and Algorithms*, Springer, Boston, MA, 2003.

4. Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., and Hebert, M., "Learning monocular reactive UAV control in cluttered natural environments," *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013.

5. Spedicato, S., and Notarstefano, G., "Minimum-Time Trajectory Generation for Quadrotors in Constrained Environments," *IEEE Transactions on Control Systems Technology*, Vol. 26, (4), 2018, pp. 1335–1344.

6. Mellinger, D., and Kumar, V., "Minimum snap trajectory generation and control for quadrotors," *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13, 2011.

7. Frew, E., and Sengupta, R., "Obstacle avoidance with sensor uncertainty for small unmanned aircraft," *IEEE Conference on Decision and Control (CDC)*, Nassau, Bahamas, 2004.

8. Tomlin, C. J., Mitchell, I., Bayen, A. M., and Oishi, M., "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, Vol. 91, (7), 2003, pp. 986–1001.

9. Sadraddini, S., Rudan, J., and Belta, C., "Formal Synthesis of Distributed Optimal Traffic Control Policies," *IEEE International Conference on Cyber-Physical Systems (ICCPS)*, Pittsburgh, PA, USA, 2017.

10. Phan-Minh, T., Cai, K. X., and Murray, R. M., "Towards Assume-Guarantee Profiles for Autonomous Vehicles," *IEEE Conference on Decision and Control (CDC)*, Nice, France, 2019.

11. Alimbayev, T., Moy, N., Nallan, K., Mishra, S., and Julius, A., "A contract based approach to colllision avoidance for UAVs," *Vertical Flight Society 76th Annual Forum*, held online, 2020.

12. Idres, M., Mustapha, O., and Okasha, M., "Quadrotor trajectory tracking using PID cascade control," *IOP Conference Series: Materials Science and Engineering, Putrajaya, Malaysia*, Vol. 270, 12 2017, pp. 012010.

13. Lee, S.-h., Kang, S. H., and Kim, Y., "Trajectory tracking control of quadrotor UAV," *11th International Conference on Control, Automation and Systems*, Gyeonggi-do, South Korea, October 26-29, 2011.

14. Das, A., Subbarao, K., and Lewis, F., "Dynamic inversion with zero-dynamics stabilisation for quadrotor control," *Control Theory  Applications, IET*, Vol. 3, 04 2009, pp. 303 – 314.