A Contract Based Approach to Collision Avoidance for UAVs

Talgat Alimbayev
MS student
RPI
Troy, NY, USA

Nicholas Moy
MS Student
RPI
Troy, NY, USA

Kaushik Nallan
PhD Student
RPI
Troy, NY, USA

Sandipan Mishra
Associate Professor
RPI
Troy, NY, USA

A. Agung Julius
Associate Professor
RPI
Troy, NY, USA

ABSTRACT

In this work, a contract-based reasoning approach is developed for obstacle avoidance in unmanned aerial vehicles (UAV's) under evolving subsystem performance. This approach is built on an assume-guarantee framework, where each subsystem (guidance, navigation, control and the environment) assumes a certain level of performance from other subsystems and in turn provides a guarantee of its own performance. The assume-guarantee construct then assures the performance of the overall system (in this case, safe obstacle avoidance). The implementation of the assume-guarantee framework is done through a set of *contracts* that are encoded into the guidance subsystem, in the form of a set of inequality constraints in the trajectory planner. The inequalities encode the relationships between subsystem performance and operational limits that ensure safe and robust operation as the performance of the control and navigation subsystems and environment evolve over time. The contract inequalities can be obtained analytically or numerically using an optimization based path planner and UAV simulation. The methodology is evaluated in the context of head-on obstacle avoidance, where the contracts are constructed in terms of (1) minimum obstacle detection range, (2) expected obstacle size, (3) maximum allowed cruise velocity, (4) maximum allowable thrust, roll and pitch angles, and (5) inner-loop tracking performance. Numerical and analytical generation of these contracts for this scenario is demonstrated. Finally, in-flight contract enforcement is illustrated for typical scenarios.

NOTATION

C_P - controller subsystem performance metadata [m]
C_N - sensor performance metadata [m]
$\mathscr{C}_0$ - polytopic obstacle set
$\mathscr{C}$ - collision set
C and D - polytope equation parameters
D_{od} - obstacle detection range
Env - information about the operating environment
E_{tr} - trajectory tracking error or deviation from the commanded trajectory
$\mathbf{F}/\hat{\mathbf{F}}$ - continuous and discrete chains of integrators
f_s - simplified 6DOF dynamics
$\mathbf{G}/\hat{\mathbf{G}}$ - continuous and discrete input matrices
g - gravity
J - cost function
K - rotational inertia in ψ
k - discrete time index
m - mass of the UAV
N - number of discrete time intervals
$\mathscr{N}_p$ and $\mathscr{N}_u$ - mappings for the the flat output y and its derivatives to the state and input, respectively
$\mathscr{O}$ - state of the obstacle
p - set of states of the UAV (position, attitude and their derivatives)
p_0/p_D - initial and desired final states of the UAV
$\mathbf{q} = [X, \dot{X}, Y, \dot{Y}, Z, \dot{Z}, \psi, \dot{\psi}]^T$
$\hat{\mathbf{q}}(k) = [X(kt_s), \dot{X}(kt_s), Y(kt_s), \dot{Y}(kt_s), Z(kt_s), \dot{Z}(kt_s), \psi(kt_s), \dot{\psi}(kt_s)]^T$,
R_{obst} - obstacle size
T - total thrust
T_{max} - maximum thrust available to the controller subsystem
t_0/t_f - initial/final times
u - set of synthetic inputs
u_ψ - yaw moment
V_{max} - maximum cruise velocity
$\mathbf{w}/\hat{\mathbf{w}}$ - continuous and discrete sets of $[\ddot{X}, \ddot{Y}, \ddot{Z}, u_\psi]^T$
X, Y, Z - position of the UAV
X_B, Y_B, Z_B - body reference frame
x_r, y_r, z_r - reference signals sent to the controller
$\mathbf{y}$ - differentially flat output $[X, Y, Z, \psi]^T$
Γ - the contract in the form of inequality
$\phi/\theta/\psi$ - roll/pitch/yaw angles
ϕ_{max}/θ_{max} - maximum roll/pitch angles
$\phi_r/\theta_r/\psi_r$ - reference attitude sent to the controller
Ξ_1 - set of allowed UAV states
Ω - set of V_{max}, ϕ_{max}, θ_{max} and T_{max}

INTRODUCTION

With the ever-growing use of unmanned aerial vehicles (UAVs) in civilian and military applications, ensuring robust performance under evolving (and degrading) subsystem performance is particularly critical for safe operation. While the standard guidance, navigation, and control (GNC) architecture typically guarantees safe operation under nominal system performance, it is typically not able to provide similar guarantees when subsystems are performing under off-nominal

Presented at the VFS International 76th Annual Forum & Technology Display, Virginia Beach, Virginia, October 6–8, 2020.

conditions as it assumes that the performance of each subsystem (or component) does not change with time. However, in many operational scenarios, the performance of subsystems of the UAV may change. For example, the performance of the control subsystem may degrade as a result of decreased maximum thrust available due to the damage to rotors or motors (Refs. 1, 2). Similarly, the performance of the navigation subsystem can vary due to changes in the environment, say, degradation of visual environments due to brown out or fog (Ref. 3). There are a handful of well-explored avenues of dealing with the response of the overall control architecture to such system degradation. Some approaches focus on sensing algorithms and alternative sensing hardware to mitigate a loss of sensor performance (Ref. 4). In (Ref. 5), the sensor degradation is quantified and used to schedule gains, enabling an increase in performance over a more conservative design, while still ensuring satisfactory performance. Model reference adaptive control is another mechanism to respond to UAV performance degradation (Ref. 6).

One of the major issues with such methodologies is the lack of a unifying underlying framework. To address this issue, there is now a growing body of research on *contract based system design* approaches, which offer a more standardized and generalizable methodology. In this approach, a system is decomposed into smaller, easier to analyze subsystems using the assume-guarantee reasoning paradigm (Ref. 7). Under the assume-guarantee reasoning, each subsystem guarantees a certain level of performance while assuming that other subsystems perform at a certain level of performance as well. The overall system is constructed in a such a way that if all subsystems hold true to their guarantee, the performance of the system is guaranteed to be satisfactory.

Assume-guarantee reasoning methods have been used in a variety of applications. For example, in (Ref. 8), such a strategy was used for designing an optimal control policy of traffic light signals. The authors partitioned a traffic network into smaller sub-networks. Each sub-network assumes that other sub-networks have an access to the global clock and the volume of vehicles leaving an adjacent sub-network does not exceed a certain level. Each sub-network guarantees the same requirements as stated above. An assume-guarantee reasoning framework was also proposed for autonomous vehicles in (Ref. 9). There, the vehicles guarantee that their actions follow a certain action structure and the assumption the vehicles make is that other cars follow a certain structure as well and have a predictor to assess if their actions follow the structure. Such an approach was also demonstrated for a safety critical aircraft power system in (Ref. 10).

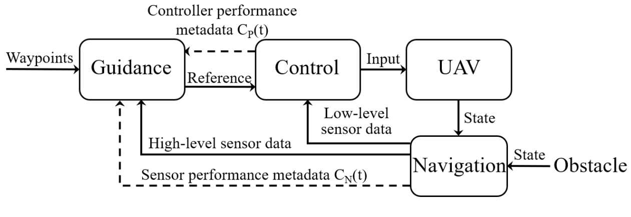

Figure 1: Proposed GNC architecture.

In this paper, we propose a contract based reasoning approach for UAV control. The proposed approach uses a modified GNC architecture, as shown in Figure 1. The overall UAV GNC system is decomposed into subsystems and each subsystem assumes a certain level of performance from other subsystems and guarantees a certain level of performance itself. Essentially, we use a contract that resides in the guidance subsystem and encodes the relationship between subsystem performance and operational limits to ensure safe and robust operation as the performance of the control and navigation subsystems and environment evolve over time. This contract can encode this relationship for a variety of specific scenarios, and ensure safety so long as all subsystems satisfy the contract. This paper presents general analytical and numerical approaches to obtaining this contract, and illustrates it specifically for the scenario of obstacle avoidance.

PROBLEM STATEMENT

Consider the typical guidance-navigation-control system where some metrics of real-time controller performance ($C_P(t)$) and real-time sensor performance ($C_N(t)$) are available on-the-fly, as shown in Fig. 1. Naturally, it is desirable to utilize this information to dynamically adapt to the changes in the performance of the subsystems. In this paper, we develop a framework to use $C_P(\cdot)$ and $C_N(\cdot)$ in the guidance subsystem as shown in Figure 1 to adapt to the subsystem performance changes.

Optimization-based guidance: The guidance subsystem is typically formulated as an optimization problem, (1), where $p(\cdot)$ and $u(\cdot)$ are the system state and (feedforward) input trajectories to be determined.

$$\min_{p(\cdot),u(\cdot)} \quad J(p,u), \tag{1}$$

$$s.t. \quad \dot{p}(t) = f_s(p(t),u(t)), \tag{2}$$

$$p(t_0) = p_0, \quad p(t_f) = p_D, \tag{3}$$

$$p(\cdot) \notin \mathscr{O}, p(\cdot) \in \Xi_1(\Omega), u(\cdot) \in \Xi_2(\Omega) \tag{4}$$

where (1) is the cost function (flight time, fuel, flight distance etc.), (2) is simplified (nonlinear) dynamics of the UAV, (3) is initial and terminal state constraints and (4) is obstacle avoidance, path and input constraints.

One mechanism to adapt to changes in navigation and control subsystem performance, $C_N(\cdot)$ and $C_P(\cdot)$ is by embedding a set of constraints, termed a '*contract*', into the (guidance) optimization problem in real-time. The contract is a set of inequalities, $\Gamma(\cdot) \leq 0$, that relate changing values of $C_N(\cdot)$ and $C_P(\cdot)$ to the limits on operational parameters Ω such that to ensure safe and robust performance, where Ω is defined as a set of operational parameters that define path and input constraints. Ω can be, for example, maximum allowed velocity, maximum trajectory roll or pitch angles etc. These constraints are enforced by appending $\Gamma(\cdot) \leq 0$ to the original optimization problem in (1). In this paper, we design a methodology to determine the relationships that make up $\Gamma(\cdot) \leq 0$.

$$\Gamma(C_P, C_N, \Omega, Env) \leq 0 \tag{5}$$

where (5) is the contract between controller, navigation and guidance subsystems. Note that the contract is also dependent on the state of the environment (*Env*).

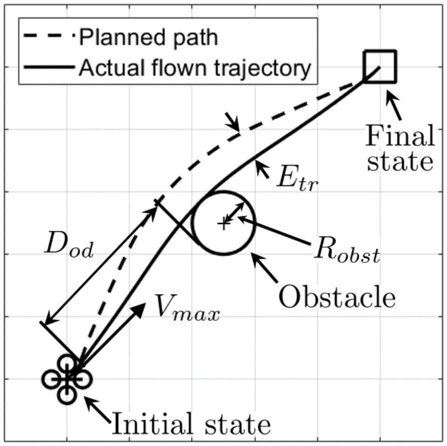

Figure 2: Obstacle avoidance scenario. The UAV is flying from the initial state to the final state with a static obstacle in between. The obstacle is detected at some distance, D_{od}, away from the UAV. The planned obstacle avoidance trajectory is tracked with some deviation from it, E_{tr}.

Problem Scenario: Obstacle Avoidance. Naturally, the choice (and construction) of the contract $\Gamma(\cdot) \leq 0$ in (5) depends on the specific operational scenario for the UAV. In this paper, as an exemplar, we consider the obstacle-avoidance scenario shown in Figure 2, where the UAV is flying from one waypoint to another waypoint with a static obstacle in its way. This is the case of the head-on collision avoidance. The expected obstacle size, R_{obst}, is what characterizes the operational environment (*Env*). The performance $C_N(\cdot)$ of the navigation subsystem is quantified by the guaranteed obstacle detection range, D_{od}, i.e. the distance within which *every obstacle is guaranteed to be detected.* The controller subsystem performance, $C_P(\cdot)$, is quantified by the trajectory tracking accuracy, E_{tr}, which may be assessed on-the-fly by looking at the current in-flight tracking performance. For this scenario, $\Gamma(\cdot) \leq 0$ is the relationship between D_{od}, E_{tr}, operational parameters and environmental information such that the collision free flight is guaranteed. In particular, the operational parameters considered are the maximum cruise velocity, V_{max}, maximum roll and pitch angles, ϕ_{max} and θ_{max}, and maximum available thrust, T_{max}, respectively. If (6) is satisfied, then trajectories planned by the guidance subsystem with the parameters Ω are guaranteed to be safe and robust to the changes in the performance of the subsystems.

$$\Gamma(\underbrace{E_{tr}}_{C_P}, \underbrace{D_{od}}_{C_N}, \underbrace{V_{max}, \phi_{max}, \theta_{max}, T_{max}}_{\Omega}, \underbrace{R_{obst}}_{Env}) \leq 0 \tag{6}$$

Remark: Though we consider obstacle avoidance here, the methodologies proposed can be extended to general flight operation.

OPTIMIZATION-BASED PATH PLANNING WITH COLLISION AVOIDANCE

Nonlinear dynamics model. As noted in (1) , the guidance subsystem requires a simplified dynamical model of the UAV to plan trajectories (in this paper, we limit our exposition to a quadcopter). The coordinate frame convention is North-East-Down, the major forces acting on the quadcopter are gravity, mg, and rotor thrust, T, and the major moment is the moment u_ψ around the yaw axis. For a quadcopter, the inner (attitude) loop controls the roll and pitch dynamics 5-10 times faster than the rest of the states (Refs. 11, 12) and thus they are assumed to be instantaneously achievable and treated as inputs to the system. Therefore, the input to the system is $u = [T, \phi, \theta, u_\psi]$, and the dynamics equations of the form $\dot{p} = f_s(p, u)$ can be rewritten as in (7), where $\ddot{X}$, $\ddot{Y}$ and $\ddot{Z}$ are accelerations in a NED ground frame and ϕ, θ and ψ are roll, pitch and yaw angles respectively.

$$\begin{aligned}
&\ddot{X} = -\tfrac{T}{m}(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi);\\
&\ddot{Y} = -\tfrac{T}{m}(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi);\\
&\ddot{Z} = g - \tfrac{T}{m}\cos\phi \cos\theta;\\
&\ddot{\psi} = K u_\psi
\end{aligned} \tag{7}$$

Differential Flatness. As demonstrated in (Refs. 13–15), differentially flat form of the system facilitates path planning. Paths may be planned entirely in the differentially flat states and input-space, and then mapped back to feasible system state and input trajectories. As (8) shows, all the inputs can be expressed through the differentially flat output, $\mathbf{y} = [X, Y, Z, \psi]^T$, and its time derivatives.

$$\begin{aligned}
&\phi = \tan^{-1}\left(\tfrac{-\ddot{X}\sin\psi + \ddot{Y}\cos\psi}{\sqrt{(g-\ddot{Z})^2 + (\ddot{X}\cos\psi + \ddot{Y}\sin\psi)^2}}\right);\\
&\theta = -\tan^{-1}\left(\tfrac{\ddot{X}\cos\psi + \ddot{Y}\sin\psi}{g - \ddot{Z}}\right);\\
&T = m\sqrt{\ddot{X}^2 + \ddot{Y}^2 + (g - \ddot{Z})^2};\\
&u_\psi = \tfrac{1}{K}\ddot{\psi}
\end{aligned} \tag{8}$$

Following the work in (Ref. 13) and using (8), it is possible to represent the system in a differentially flat form because there exist functions $\mathscr{N}_p$ and $\mathscr{N}_u$ that map the flat output $\mathbf{y}$ and its derivatives to the state and input, respectively. Then, the nonlinear dynamics of the system $\dot{p} = f_s(p, u)$ can be expressed in the equivalent form shown below in (9), with nonlinearity transferred to bounds on the synthetic input $\mathbf{w}$, so that $\mathbf{F}$ contains only a chain of integrators for each of the flat outputs.

$$\begin{aligned}
&\dot{\mathbf{q}}(t) = \mathbf{F}\mathbf{q}(t) + \mathbf{G}\mathbf{w}(t);\\
&\mathbf{q}(t) = [X(t), \dot{X}(t), Y(t), \dot{Y}(t), Z(t), \dot{Z}(t), \psi(t), \dot{\psi}(t)]^T;\\
&\mathbf{w}(t) = [\ddot{X}(t), \ddot{Y}(t), \ddot{Z}(t), \ddot{\psi}(t)]^T\\
&p(\cdot) = \mathscr{N}_p(\mathbf{y}(\cdot), \dot{\mathbf{y}}(\cdot), \ddot{\mathbf{y}}(\cdot));\;\; u(\cdot) = \mathscr{N}_u(\mathbf{y}(\cdot), \dot{\mathbf{y}}(\cdot), \ddot{\mathbf{y}}(\cdot))
\end{aligned} \tag{9}$$

$\mathbf{F}$ and $\mathbf{G}$ are defined as given below in (10):

$$\dot{\mathbf{q}}(t) = \underbrace{\begin{bmatrix} J_2 & 0 & 0 & 0 \\ 0 & J_2 & 0 & 0 \\ 0 & 0 & J_2 & 0 \\ 0 & 0 & 0 & J_2 \end{bmatrix}}_{\mathbf{F}} \mathbf{q}(t) + \underbrace{\begin{bmatrix} L_2 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ 0 & 0 & L_2 & 0 \\ 0 & 0 & 0 & L_2 \end{bmatrix}}_{\mathbf{G}} \mathbf{w}(t) \tag{10}$$

which is a bank of pure integrator-chains, with:

$$J_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{11}$$

Discretization. In order to solve the original trajectory planning problem in (1), it is first necessary to temporally discretize the system. The system is discretized using the zero-order hold method with the sample time, t_s, as shown in (12), where N is a pre-selected number.

$$\hat{\mathbf{q}}(k+1) = \hat{\mathbf{F}}\hat{\mathbf{q}}(k) + \hat{\mathbf{G}}\hat{\mathbf{w}}(k); \quad t_s = \frac{t_f - t_0}{N-1} \tag{12}$$

where $\hat{\mathbf{q}}(k) = [X(kt_s), \dot{X}(kt_s), Y(kt_s), \dot{Y}(kt_s), Z(kt_s), \dot{Z}(kt_s), \psi(kt_s),\ \dot{\psi}(kt_s)]^T$, $w(k) = [\ddot{X}(kt_s), \ddot{Y}(kt_s), \ddot{Z}(kt_s), \ddot{\psi}(kt_s)]^T$, $k \in [0, N]$ is the state and input at k^{th} time instant. $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ are defined as given below in 13.

$$\hat{\mathbf{F}} = e^{\mathbf{F}t_s}, \quad \hat{\mathbf{G}} = \int_0^{t_s} e^{\mathbf{F}\tau}\mathbf{G}\, d\tau \tag{13}$$

assuming zero-order hold on the input.

Optimization problem restatement. If the cost function is chosen to be the total flight time, $t_f - t_0$, the general optimization (1) becomes a time optimal problem with dynamics given by (12). The boundary conditions are determined by the state of the UAV at t_0 and t_f. The map from the synthetic input to the system input given in (8) along with the limits imposed on $|\phi|$, $|\theta|$, $|T|$ and $|u_\psi|$ are used to generate the path constraints for the optimization problem. The UAV must avoid the (convex) obstacle. The construction of the contract requires a bound on the maximum horizontal speed, V_{max}. Thus, the boundary conditions and path and obstacle constraints are as follows in (14)

Boundary conditions:
$\hat{\mathbf{q}}(1) = \hat{\mathbf{q}}_{\mathbf{0}}; \quad \hat{\mathbf{w}}(1) = \hat{\mathbf{w}}_0; \quad \hat{\mathbf{q}}(N) = \hat{\mathbf{q}}_{\mathbf{f}}; \quad \hat{\mathbf{w}}(N) = \hat{\mathbf{w}}_f;$

Path constraints:

$$\begin{aligned}
&[\hat{\mathbf{q}}, \hat{\mathbf{w}}] \in \Xi_1 : \\
&0 \leq \sqrt{\dot{X}^2(kt_s) + \dot{Y}^2(kt_s)} \leq V_{max}; \\
&z_{min} \leq z(kt_s) \leq z_{max}; \\
&\left| \tan^{-1}\left(\frac{-\ddot{X}(kt_s)\sin\psi(kt_s) + \ddot{Y}(kt_s)\cos\psi(kt_s)}{\sqrt{(g-\ddot{Z}(kt_s))^2 + (\ddot{X}(kt_s)\cos\psi(kt_s) + \ddot{Y}(kt_s)\sin\psi(kt_s))^2}}\right) \right| \leq \phi_{max}; \\
&\left| -\tan^{-1}\left(\frac{\ddot{X}(kt_s)\cos\psi(kt_s) + \ddot{Y}(kt_s)\sin\psi(kt_s)}{g - \ddot{Z}(kt_s)}\right) \right| \leq \theta_{max}; \\
&T_{min} \leq m\sqrt{\ddot{X}(kt_s)^2 + \ddot{Y}(kt_s)^2 + (g - \ddot{Z}(kt_s))^2} \leq T_{max}; \\
&\left|\dot{\phi}(kt_s)\right| \leq \dot{\phi}_{max}; \quad \left|\dot{\theta}(kt_s)\right| \leq \dot{\theta}_{max}; \\
&u_{\psi min} \leq \tfrac{1}{K}\ddot{\psi}(kt_s) \leq u_{\psi max};
\end{aligned} \tag{14}$$

Obstacle avoidance constraints:

$$\begin{aligned}
&\mathbf{q} \notin \mathscr{O}, \text{ for a circular column obstacle:} \\
&(X(kt_s) - X_{obst})^2 + (Y(kt_s) - Y_{obst})^2 \geq R_{obst}^2; \text{ and} \\
&\hat{\mathbf{q}}(k) \notin \mathscr{C}_0 \text{ for a polytopic obstacle, where } \mathscr{C}_0 \text{ is the set} \\
&\text{that defines the polytopic obstacle;}
\end{aligned} \tag{15}$$

The discretized optimization problem can be stated as follows

$$\{t_f*, \hat{\mathbf{w}}^*(\cdot), \hat{\mathbf{q}}^*(\cdot)\} = \underset{t_f, \hat{\mathbf{q}}(\cdot), \hat{\mathbf{w}}(\cdot)}{\arg\min} \quad J = t_f - t_0, \tag{16}$$

$$s.t \quad \hat{\mathbf{q}}(k+1) = \hat{\mathbf{F}}\hat{\mathbf{q}}(k) + \hat{\mathbf{G}}\hat{\mathbf{w}}(k), \tag{17}$$

$$\hat{\mathbf{q}}(1) = \hat{\mathbf{q}}_0, \hat{\mathbf{q}}(N) = \hat{\mathbf{q}}_f, \hat{\mathbf{w}}(1) = \hat{\mathbf{w}}_0, \hat{\mathbf{w}}(N) = \hat{\mathbf{w}}_f \tag{18}$$

$$\hat{\mathbf{q}}, \hat{\mathbf{w}} \in \Xi_1, \hat{\mathbf{q}} \notin \mathscr{O}, \tag{19}$$

where (16) is the cost function in the form of the total flight time, (17) is the differentially flat discrete-time dynamics, (18) are the initial and terminal boundary constraints and (19) are the path and obstacle obstacle constraints.

Solving the minimization problem in (16) gives the optimal input $\hat{\mathbf{w}}^*(\cdot)$ and corresponding trajectory $\hat{\mathbf{q}}^*(\cdot)$. Using the equations in (7) and (8), $\hat{\mathbf{w}}^*(\cdot)$ and $\hat{\mathbf{q}}^*(\cdot)$ can be converted to the reference signals for the UAV controller as shown in Figure 1. The reference signals are $x_r, y_r, z_r, \phi_r, \theta_r, \psi_r$.

Construction of the contract term: The remainder of the paper will suggest an additional set of inequality constraints $\Gamma(E_{tr}, D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, Env) \leq 0$ that can be appended to (16) to guarantee safe and robust performance of the system as the performance of its components evolves with time. An interpretation of $\Gamma(\cdot) \leq 0$ is that it encodes the safe operational conditions, i.e. where the statements $\mathbf{q} \in \Xi_1, \mathbf{q} \notin \mathscr{O}$ are satisfied. Then, when $\Gamma(\cdot) \leq 0$ cannot be satisfied, safe operation is not guaranteed. The infeasibility of (16) can be used to identify unsafe or impossible operating conditions, which can be encoded in $\Gamma(\cdot) \leq 0$. To implement this, an analytical approach can be used to identify states that lead to collision with the obstacle. Alternatively, the guidance path planning algorithm (16) can be used to numerically find infeasible combinations of V_{max}, D_{od}, ϕ_{max}, θ_{max} and R_{obst}.

ANALYTICAL CONSTRUCTION OF CONTRACT

In this section, we obtain the contract, $\Gamma(\cdot) \leq 0$, by identifying the states that lead to an inevitable collision with the obstacle, given path and input constraints. We do so by introducing the notion of collision set which is the set of all UAV states that lead to a collision irrespective of the future inputs. We also discuss an efficient way to compute the collision set and how the collision set along with the path and input constraints is used to generate the inequalities that make up the contract $\Gamma(\cdot) \leq 0$ to guarantee safe operation of the UAV.

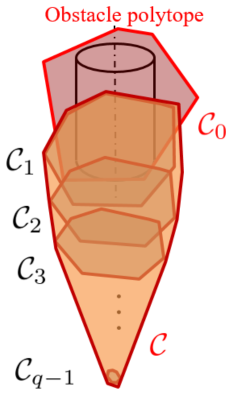

Figure 3: Illustration of the collision set. If the system state enters the collision set, $\mathscr{C} = \cup_{i=1}^{q-1}\mathscr{C}_i$, a crash is inevitable.

The Collision Set

To start, we define the *unsafe* or *collision set*, i.e., the set of states for which there does not exist an input sequence that makes the state avoid the obstacle in (20).

$$\begin{aligned}\mathscr{C} = \{\hat{\mathbf{q}}(k)|\forall(\hat{\mathbf{w}}(k)\in\mathbf{W}, \hat{\mathbf{w}}(k+1)\in\mathbf{W},\dots,\\ \hat{\mathbf{w}}(k+\hat{k})\in\mathbf{W}),\ \exists\hat{k}\in\mathbb{Z}^+|\hat{\mathbf{q}}(k+\hat{k})\in\mathscr{O}\}\end{aligned} \tag{20}$$

Conversely, the state of the UAV being outside $\mathscr{C}$ guarantees the existence of a feasible path that avoids $\mathscr{O}$. Note that it is not necessary to prescribe any initial conditions for this approach, nor are any avoidance paths constructed.

$\hat{k}$-Step Collision Set To compute (20), we form recursively defined sets $\mathscr{C}_{\hat{k}}$ for each subsequent backwards time step from the time step when a collision occurs. Then, we can take the union of these sets to obtain $\mathscr{C}$, as shown in Figure 3. We define the set of states $\mathscr{C}_{\hat{k}}$ in (21) such that for any allowable input, the state will evolve into $\mathscr{C}_{\hat{k}-1}$ in the next time step, and consequently into $\mathscr{O}$ in $\hat{k}$ time steps.

The initialization of the recursion is formed by taking $\hat{k}=1$ and $\mathscr{C}_{\hat{k}-1}=\mathscr{O}$. By construction, we can now express $\mathscr{C}$ as in (22), where $\mathscr{C}_q=\emptyset$.

$$\mathscr{C}_{\hat{k}} = \{\hat{\mathbf{q}}(k)|\forall\hat{\mathbf{w}}(k)\in\mathbf{W}, \hat{\mathbf{q}}(k+1)\in\mathscr{C}_{\hat{k}-1}\} \tag{21}$$

$$\mathscr{C} = \mathscr{C}_0\cup\mathscr{C}_1\cup\ldots\cup\mathscr{C}_{q-1} \tag{22}$$

Efficient Computation of the Collision Set

In order to compute the propagation of these sets, we take advantage of the linear propagation of the differentially flat states and use convex approximations for the input bounds and obstacle so that computations only need to be done for the vertices of the resulting polytopes. While this problem will grow exponentially, a method to prevent the growth will be presented. Let the obstacle $\mathscr{O}$ be approximated by a p-faceted polytope $\mathscr{C}_0=\{\hat{\mathbf{q}}|C_0\hat{\mathbf{q}}\le D_0\}$. Let the allowable input set be approximated by $\mathbf{W}^C = conv(\mathbf{w}_1,\mathbf{w}_2,\dots,\mathbf{w}_m)$, where $conv(\cdot)$ means the convex hull of the argument. For this approach, we then require $\hat{\mathbf{w}}(k)\in\mathbf{W}^C$, $\forall k$.

Letting $\hat{\mathbf{q}}$ evolve as in (12) and replacing $\mathbf{W}$ with $\mathbf{W}^C$, we can specialize (21) to a system with linear state propagation, as shown in (23). Then, it can be shown that the *conjunction* of the conditions in (24) is necessary and sufficient for elements $\hat{\mathbf{q}}$ to be in $\mathscr{C}_{\hat{k}}$.

$$\mathscr{C}_{\hat{k}} = \{\hat{\mathbf{q}}|\forall\hat{\mathbf{w}}\in\mathbf{W}^C, \hat{\mathbf{F}}\hat{\mathbf{q}}+\hat{\mathbf{G}}\hat{\mathbf{w}}\in\mathscr{C}_{\hat{k}-1}\} \tag{23}$$

$$\hat{\mathbf{F}}\hat{\mathbf{q}}+\hat{\mathbf{G}}\mathbf{w}_1\in\mathscr{C}_{\hat{k}-1},\quad \dots,\quad \hat{\mathbf{F}}\hat{\mathbf{q}}+\hat{\mathbf{G}}\mathbf{w}_m\in\mathscr{C}_{\hat{k}-1} \tag{24}$$

Then, we obtain a closed propagation from $\mathscr{C}_{\hat{k}-1}$ to $\mathscr{C}_{\hat{k}}$ by concatenating those conditions for each vertex of the allowable input set $\mathbf{w}_j$ as in (25). To prevent the exponential growth in number of rows as we continue this process to define subsequent sets $\mathscr{C}_{\hat{k}+1},\mathscr{C}_{\hat{k}+2},\dots,\mathscr{C}_{q-1}$. Note that the ith row of the inequality in (25) represents a hyperplane with the normal vector given by the ith row of $C_{\hat{k}}$ and offset given by the ith element of $D_{\hat{k}j}$. Further, $C_{\hat{k}}$ is independent of w_j.

The full set of constraints is then p sets of m parallel hyperplanes, so we can condense the inequalities into p rows at each step as shown in (26), where $D_{\hat{k}ji}$ is the ith element of $D_{\hat{k}j}$. Propagation continues until it is found that $C_{\hat{k}}\hat{\mathbf{q}}\le D_{\hat{k},min}$ is infeasible. Now that each $\mathscr{C}_{\hat{k}}$ has been computed, we can find the union $\mathscr{C}$. Although each set $\mathscr{C}_{\hat{k}}$ is convex, $\mathscr{C}$ need not be convex, in general. To obtain the closed propagation, we start with the inequalities that describe the polytope $\mathscr{C}_{k-1}=\{\hat{\mathbf{q}}|C_{k-1}\hat{\mathbf{q}}\le D_{k-1}\}$ and find C_k and D_k (matrices describing the set $\mathscr{C}_k$) as shown below.

$$\begin{aligned} C_{\hat{k}-1}\hat{\mathbf{q}} &\le D_{\hat{k}-1},\\ C_{\hat{k}-1}(\hat{\mathbf{F}}\hat{\mathbf{q}}+\hat{\mathbf{G}}\mathbf{w}_j) &\le D_{\hat{k}-1},\\ C_{\hat{k}-1}\hat{\mathbf{F}}\hat{\mathbf{q}} &\le D_{\hat{k}-1}-C_{\hat{k}-1}\hat{\mathbf{G}}\mathbf{w}_j,\\ C_{\hat{k}}\hat{\mathbf{q}} &\le D_{\hat{k}j}.\end{aligned} \tag{25}$$

$$C_{\hat{k}}\hat{\mathbf{q}} \le \begin{bmatrix}\min\{D_{\hat{k}11},D_{\hat{k}21},\dots,D_{\hat{k}m1}\}\\ \min\{D_{\hat{k}12},D_{\hat{k}22},\dots,D_{\hat{k}m2}\}\\ \vdots\\ \min\{D_{\hat{k}1p},D_{\hat{k}2p},\dots,D_{\hat{k}mp}\}\end{bmatrix} = D_{\hat{k},min}. \tag{26}$$

Extraction of Contract from the Collision Set

For the above analytical approach, we have imposed constraints on the synthetic input such that the corresponding system inputs are always within the system input bounds. Hence, the constructed collision set not only depends obstacle parameters R_{obst} but also on the system input bounds i.e. $\phi_{max},\theta_{max},T_{max}$.

In this section, we are interested to generate a contract (Γ) from the collision set ($\mathscr{C}$). In other words, given system input bounds ($\phi_{max},\theta_{max},T_{max}$), obstacle parameters ($R_{obst}$) and maximum allowable velocity V_{max}, we are interested to calculate the minimum distance from the obstacle such that the collision can be avoided. Technically, this minimum distance also depends on the position and velocity direction of the UAV. For the safety of the vehicle, we would like our contract to be coordinate-free, that is, the contract should be valid for any position or velocity of the UAV, given the distance to the obstacle and maximum speed. Therefore, it is sufficient to generate a contract that captures the worst case scenario, which is trying to avoid head-on collision with the obstacle. Though any coordinate frame can be used to compute the collision set, the problem can be simplified if the coordinate frame is fixed on the face of obstacle and is oriented such that it easily captures the head-on collision scenario. The obtained collision set is a union of (finite) convex polytopes as shown below.

$$\mathscr{C} = \mathscr{C}_0 \cup \mathscr{C}_1 \cup \mathscr{C}_2 \cup \ldots \cup \mathscr{C}_r, \tag{27}$$

where each of the $\mathscr{C}_k$ is set of inequalities in states of the UAV. In our case of head-on collision and simplified coordinate frame, we obtain the velocity (magnitude) and distance inequalities. For every $\mathscr{C}_k$, only one inequality corresponds to the crash conditions with our particular face (of obstacle) of interest. Given the maximum velocity of UAV (V_{max}), we can calculate the minimum required distance to the obstacle such that inevitable collision can be avoided in k time steps. Therefore, the inequality for each $\mathscr{C}_k$ can be interpreted as shown below.

$$D_{ob}\, c_{k,11} + V_{max}\, c_{k,12} \leq d_{k1} \tag{28}$$

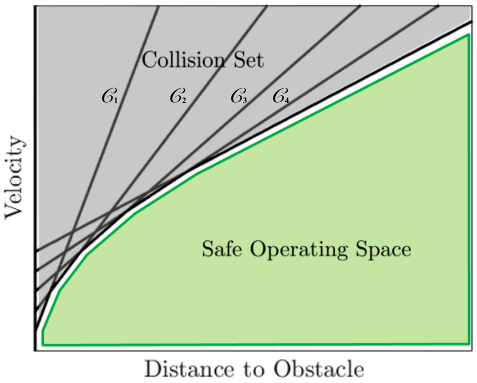

Figure 4: Obtaining Velocity-Distance Curve using analytical approach

Each inequality, which has a corresponding collision set, can be plotted together as shown in Figure 4. Also, each inequality corresponds to a straight line (linear inequalities) and separates the safe and unsafe regions. We can obtain the overall unsafe region from the plot which is the union of individual unsafe regions. The remaining region in the plot is of our interest since it is safe. This way, we obtain our contract $\Gamma(\cdot) \leq 0$ which is the safe region in the plot.

Since the computed collision set depends on the bounds on inputs and size of the obstacle, our inequalities that make up the contract can be simplified and expressed as $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) \leq 0$. It should be noted that the collision sets provide us an analytical framework to construct the contract but not necessarily an analytical expresssion for $\Gamma(.)$. There is another way to obtain the inequalities that make up the contract as discussed in the next section.

NUMERICAL CONSTRUCTION OF CONTRACT

As was stated before, the infeasibility of (16) can be interpreted as an unsafe or impossible operating condition. The guidance path planning algorithm (16) can be run iteratively to find these conditions, i.e. the infeasible combinations of V_{max}, D_{od}, ϕ_{max}, θ_{max} and R_{obst}. The feasible and infeasible combinations of the aforementioned parameters are separated by a hyper-surface $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) = 0$ into two regions. The region encoded as $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) \leq 0$ is feasible and safe. The region encoded as $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) > 0$ is infeasible and as such unsafe. The next section will compare both analytical and numerical approaches of obtaining the contract.

Comparison of Analytical and Numerical Construction of contract

Before proceeding with comparing the analytical and numerical approaches, it is necessary to describe the nature of the obstacles used in both approaches. The numerical solution uses a circular column obstacle of radius $1m$. The analytical approach approximates the circular column with two square columns: one inscribed in the circle and the other square circumscribes the circle. This ensures that the result for the circular obstacle will be bounded by the results of the two square obstacles. The sides of the inscribed and circumscribed squares are $\sqrt{2}m$ and $2m$ respectively.

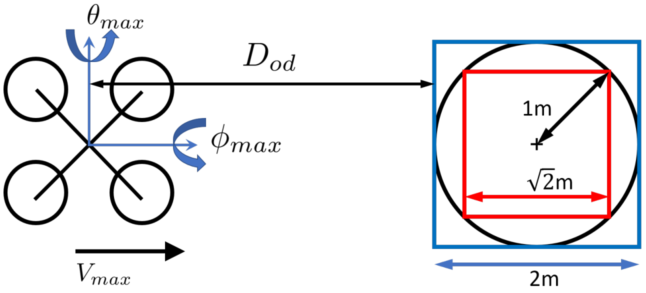

Figure 5: The plan view of the obstacles and UAV used in the construction of the contract. Square columns with the sides of $\sqrt{2}m$ and $2m$ are used in the analytical approach. The circular column with the radius of $1m$ is used in the Numerical approach.

The infeasible combinations of V_{max}, D_{od}, ϕ_{max}, θ_{max} and R_{obst} found iteratively can be cross-checked against the same combinations found by the analytical approach. The results of the cross-check are presented below in Figures 6 and 7. The infeasible and unsafe region is above each of the curves, i.e. flying with the operating condition from the region above the curve cannot guarantee safety. The results are presented for the obstacles defined above in Figure 5.

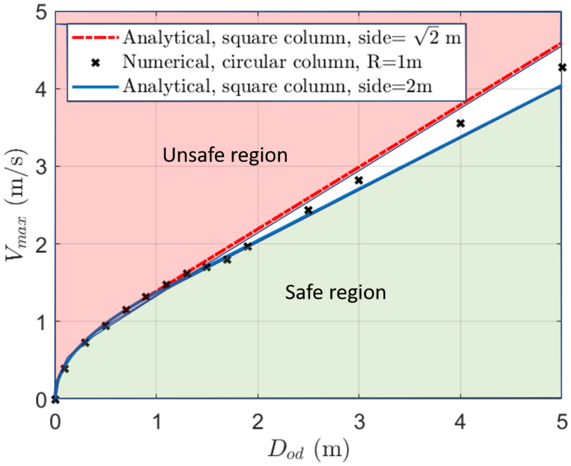

Figure 6: Comparison of analytical and numerical approaches. Plot of V_{max} vs D_{od}, $\phi_{max}= 5^o$, $\theta_{max} = 5^o$. Safe and feasible region is below each of the curves. The results correlate well. The numerical results for the circular column are underbound and overbound by the analytical results for two square columns as expected.

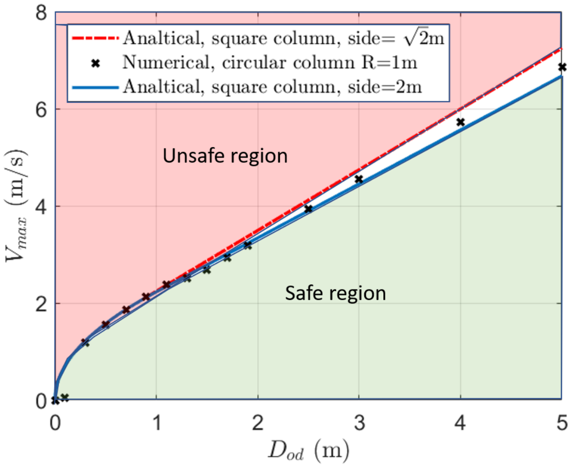

Figure 7: Comparison of analytical and numerical approaches. Plot of V_{max} vs D_{od}, ϕ_{max}=13^o, θ_{max}=13^o. Safe and feasible region is below each of the curves. The results correlate well. The numerical results for the circular column are underbound and overbound by the analytical results for two square columns as expected.

From Figures 6 and 7, we find that the analytical and numerical approaches *correlate well* for $D_{od} \leq 5m$. The discrepancy between analytical and numerical approaches can be attributed to the nature of the obstacle used in each of the approaches: a square column in the analytical case and circular column in the numerical case. Notice that as the maximum allowed roll and pitch angles are increased from 5^o to 13^o, the maximum allowed safe cruise velocity is increased as well. The greater roll and pitch angles allow for a more aggressive trajectory and thus it is possible to plan a trajectory for a higher cruise velocity with greater trajectory angles.

While contracts can be constructed using either of the approaches presented above, we proceed with the numerical approach as it facilitates implementation of the constraints in (14).

CONSTRUCTION OF GENERAL CONTRACTS

UAV simulation model

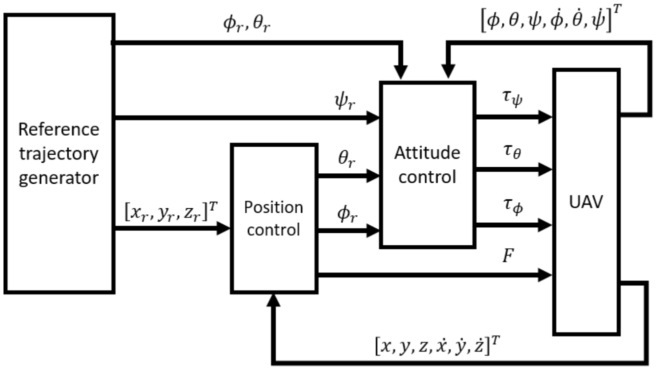

Figure 8: The general overview of the inner loop control algorithm. The controller subsystem is a series of cascaded PID controllers. The reference trajectory generator is the guidance subsystem.

Prior to describing the general contract that involves the performance of the Navigation and Controller subsystems as well as the Operational Environment, it is necessary to briefly describe the model used to assess the UAV simulation model that will be used to assess the Controller subsystem performance as well as to test the proposed methodology for obstacle avoidance. The model is a Simulink model of Parrot Mambo quadcopter which is a standard part of the Aerospace Blockset of MATLAB. It uses the empirical moment of inertia tensor, ω^2 model for thrust and torque, cascaded PID controller with x_r, y_r, z_r and ψ_r as reference signals and feedforward attitude (ϕ_r, θ_r) signal for improved reference tracking as shown in Figure 8. This model will be first used to assess the controller subsystem performance and later on to test if the proposed methodology can plan trajectories so as to safely avoid an obstacle. When assessing the controller subsystem performance it is also important to account for the possible controller performance degradation. The controller

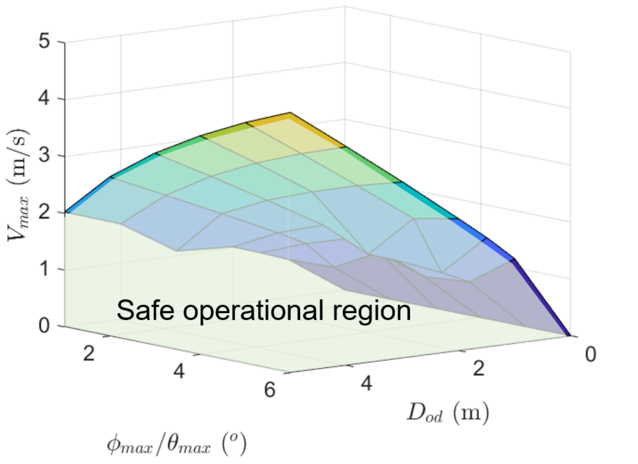

Figure 9: The result of the numerical sweep for various V_{max}, D_{od}, ϕ_{max} and θ_{max}. The region $\Gamma(\cdot) \leq 0$ becomes the contract and it encodes feasible combinations V_{max}, D_{od}, ϕ_{max} and θ_{max} for $R_{obst} = 1m$.

subsystem performance degradation is modelled as a decrease of thrust gain in the ω^2-model, which decreases the maximum thrust T_{max} available to the controller subsystem from the nominal value of thrust T_{nom}. The effect of the maximum thrust decrease is demonstrated in a later section along with the contour error which is the method used in this paper to quantify the controller subsystem performance.

Contract: Navigation subsystem

To construct the contract between the Navigation and Guidance subsystems, it is necessary to find the infeasible combinations of V_{max}, D_{od}, ϕ_{max}, θ_{max} and R_{obst} that the UAV is likely to encounter in its operation.

To do so, a numerical sweep through all of these combinations is performed. The infeasible combinations of V_{max}, D_{od}, ϕ_{max}, θ_{max} and R_{obst} can be encoded as a hyper-surface and region $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) \leq 0$ below the hyper-surface becomes the contract. To represent the region, we may fit a curve to the numerical sweep data points. For brevity, the equation of the fitted hyper-surface is omitted. The key idea is that when $\Gamma(\cdot) \leq 0$ is satisfied, the unsafe operating conditions are avoided. The result of the numerical sweep for $R_{obst} = 1m$ is shown below in Figure 9. As Figures 6, 7 and 9 show, greater obstacle detection range D_{od} and more trajectory agility (higher ϕ_{max} and θ_{max}) allow for higher cruise velocity V_{max}. Similar to Figures 6 and 7, the safe and feasible region is below and the unsafe and infeasible region is above the surface in Figure 9.

One interpretation of Figure 9 is that for some environment with a particular expected obstacle size (R_{obst}) changes in the obstacle detection range D_{od}, maximum allowed agility ϕ_{max} and θ_{max}, necessitate change in the cruise velocity V_{max} for any planned trajectory. The contract is thus:

"given an environment with a certain expected obstacle size (R_{obst}), if the navigation subsystem can guarantee detection of every obstacle within D_{od}, then the trajectories planned by the guidance subsystem subject to a maximum velocity V_{max} that corresponds to the selected ϕ_{max} and θ_{max} ensure that obstacle avoidance is possible."

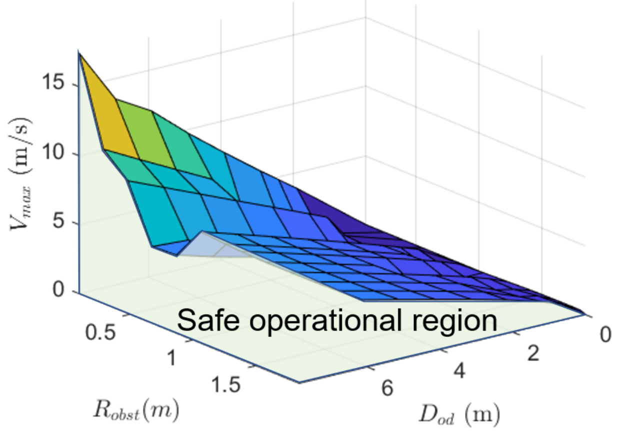

Figure 10: V_{max} as the function of obstacle size and obstacle detection range (R_{obst} and D_{od}) for $\phi_{max} = \theta_{max} = 7^o$. The safe and feasible region is below the surface.

This contract can be mathematically expressed as $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) \leq 0$ and is shared between the navigation and guidance subsystems. It is important to note that in Figure 9 the expected obstacle size, R_{obst}, was held constant at $1m$. Next, we show how the expected obstacle size affects the contract and how the controller subsystem performance can be included in the construction of the full contract.

Contract: Environment

Environmental variables can also we included in the contract. For example, the expected obstacle size (an environmental variable external to the UAV) also affects the maximum cruise velocity since an obstacle of a bigger size would require a lower cruise speed or more aggressive trajectory angles to avoid it. If the UAV is operating in a forest, then the most likely type of obstacle is a tree trunk or an occasional animal. When operating in the cities at the lower altitudes, then the most likely obstacle is a human, car or again a tree trunk. Having the knowledge about the environment (R_{obst}), it is possible to enforce the "speed limit" so as to guarantee safe operation. Figure 10 below shows how the obstacle size (R_{obst}) and obstacle detection range (D_{od}) influence the maximum cruise speed (V_{max}) for one set of maximum roll and pitch angles (ϕ_{max} and θ_{max}). As Figure 10 shows, a greater obstacle size would require the UAV to fly slower in order to safely avoid it. Similar to Figures 6, 7 and 9, the safe operating region is the region below the surface. Conducting the numerical sweep for various obstacle sizes, obstacle detection distances and maximum allowed pitch/roll angles

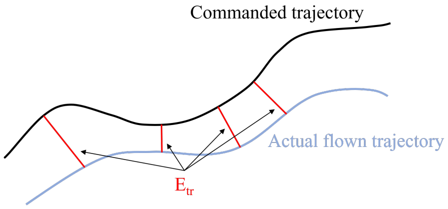

Figure 11: The contour error E_{tr} is the shortest distance to the commanded trajectory from the current state of UAV

would result in V_{max} as a function of the expected obstacle size, R_{obst} obstacle detection range, D_{od}, maximum trajectory angles, ϕ_{max} and θ_{max}. This function is encoded as a region $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}) \leq 0$ with R_{obst} replacing Env to incorporate the knowledge about the environment. The next and final step in defining the contract is to add the controller performance metrics to the contract.

Contract: Controller subsystem

When planning trajectories around obstacles, to safely avoid the obstacles, it is important to account for the deviation from the commanded trajectory because of the external disturbances, modeling uncertainty, and controller performance. The controller subsystem ultimately determines the accuracy of trajectory tracking. Thus, it is necessary to quantify the controller subsystem performance to be included in the contract. One way to do this is by using the contour error. This paper uses the well-known definition of contour error for CNC machining (Ref. 16): the shortest absolute distance from the commanded path to the current location of the UAV. Figure 11 illustrates the concept of the contour error.

Having defined the controller subsystem performance metrics, it is now possible to simulate trajectories with different V_{max}, ϕ_{max} and θ_{max} (see Figure 8). For illustrative purposes in this paper, the trajectories are simulated for different thrust limits ($0.6T_{max}$ to T_{max}) enforced in the controller subsystem, to mimic varying controller performance levels. The trajectory tracking contour error against maximum cruise velocity, V_{max}, and thrust available to the controller, T_{max}, for the fixed trajectory angles ϕ_{max} and θ_{max} is presented in Figure 12.

As can be seen from Figure 12, the contour error depends significantly on the controller performance (the maximum available thrust, T_{max}) and to a lesser extent on V_{max}. We can generate a hyper-surface for E_{tr} by sweeping through various values of V_{max}, ϕ_{max}, θ_{max} and T_{max}. Thus, the guarantee of the controller subsystem is that its performance will lie within the safe operational region shown above. The next subsection will show how the quality of the trajectory tracking can be incorporated in trajectory planning (and thus included the contract).

Virtual obstacle envelope: When planning safe trajectories around obstacles, it is important to take into account the contour error (i.e., the performance of the controller subsystem).

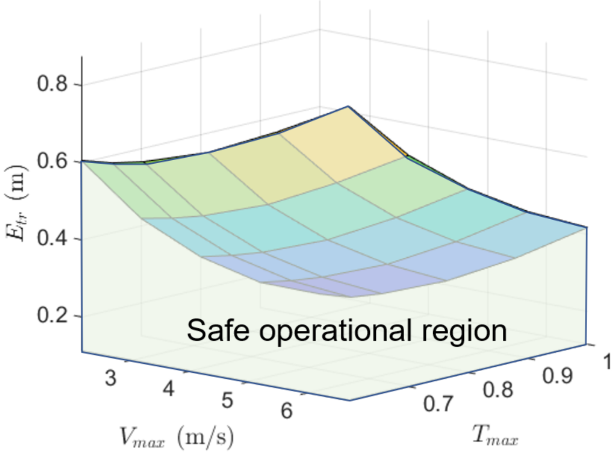

Figure 12: Contour error, E_{tr}, as a function of V_{max} and T_{max} for $\phi_{max} = \theta_{max} = 5^o$. If trajectory tracking accuracy has a value below the surface, then the UAV operates in the safe operational region.

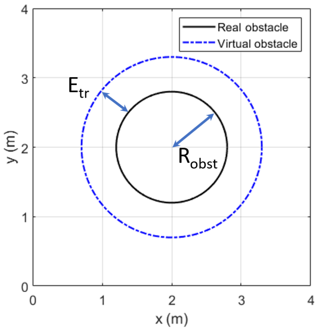

Figure 13: Real and inflated virtual obstacles. The obstacle is inflated to account for inaccurate trajectory tracking.

One way to do this is by increasing the obstacle size by an appropriate E_{tr} which is dependent on T_{max}, V_{max} and ϕ_{max}, θ_{max}. The increase in the obstacle size is shown below in the Figure 13.

As Figure 13 above shows, the size of the obstacle is increased in order to take into account the imprecise trajectory tracking. The size of the real obstacle (R_{obst}) is increased by E_{tr}. Thus, the trajectories are planned around a larger virtual obstacle. Increasing the virtual obstacle size decreases the effective distance to the obstacle and as the result the trajectory parameters V_{max}, ϕ_{max} and θ_{max} have to change to guarantee feasibility of (16) (and thus safe operation of the UAV). The size of the virtual obstacle is dependent on the controller performance, E_{tr}, and as such the region defined earlier needs to be updated to reflect that dependence. Thus, the region becomes $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}, E_{tr}) \leq 0$. The next section explains how data in Figures 9, 10 and 12 can be combined to construct the contract that can guarantee safe obstacle avoidance.

Analyzing the contract based reasoning mechanism

As was shown in Figures 6, 7 and 9, increasing allowed trajectory angles ϕ_{max} and θ_{max} enables a greater V_{max}, as this would permits planning more aggressive trajectories which in turn permits safe flights with higher cruise velocity V_{max}. Therefore, it might be tempting to plan trajectories for ϕ_{max} and θ_{max} as large as possible in order to fly at a greater V_{max}. However, as Figure 12 shows, increasing the maximum allowed cruise velocity V_{max} results in poorer tracking performance. This poor tracking performance then necessitates the change in the virtual obstacle size. Finally, the obstacle size itself might change as the UAV is flying through different environments. To take into account all of these parameters which are highly interlinked and interdependent the paper proposes to use the contract-based reasoning.

In summary, in the contract based paradigm, the system is decomposed into smaller subsystems and each subsystem assumes a certain level of performance of other subsystems and guarantees a certain level of performance itself. Thus, under this paradigm the navigation subsystem guarantees that every obstacle within a certain range (D_{od}) from the UAV is detected. The environment is assumed to have obstacle of a certain expected size, R_{obst}. The controller subsystem guarantees that given a certain trajectory parameters (V_{max}, ϕ_{max}/θ_{max}) the trajectory tracking accuracy will be no worse than a certain E_{tr}. Thus, the trajectories will be planned around a virtual obstacle of the size $R_{obst} + E_{tr}$. Finally, the guidance subsystem assumes that the navigation and controller subsystems have a certain levels of performance and it guarantees that the planned trajectories will satisfy the region $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}, E_{tr}) \leq 0$. Thus, if all the subsystems satisfy the contract then the planned trajectories would allow to safely avoid the obstacle. On the other hand if one of the subsystems does not satisfy the contract or in other words the inequality $\Gamma(\cdot) \leq 0$ is violated the safe obstacle avoidance cannot be guaranteed. To guarantee that the contract is satisfied the guidance subsystem is constantly running in the background and checks if the inequality $\Gamma(\cdot) \leq 0$ is satisfied or not. In case the inequality is not satisfied the guidance subsystem adjusts the trajectory such that the inequality is satisfied which is equivalent to the contract satisfaction. The next section presents the full contract implementation. The obstacle avoidance using the contract is demonstrated in Figures 14 and 16. The in-flight implementation of the contract is shown in Figures 17 and 18.

DEMONSTRATION OF CONTRACT IMPLEMENTATION

To demonstrate the contract implementation, the operational scenario in Figure 2 is used. The UAV is moving from some initial state to some final state with a static obstacle in-between the two. The obstacle is detected at the distance D_{od} away from the UAV. The controller subsystem performance can be off-nominal (i.e. less thrust available to the controller) and navigation subsystem performance can also change, i.e. the obstacle is detected closer to the UAV or father away from it. The guidance subsystem plans safe trajectories by taking into account all of these parameters.

Figures 14, 15 and 16 demonstrate obstacle avoidance with the full contract implemented. In order to do that, the guidance subsystem takes the following inputs to plan safe trajectories: the maximum available thrust, T_{max}, obstacle detection distance, D_{od}, and the expected obstacle size, R_{obst}. Given this information, the guidance subsystem adjusts operational limits to satisfy the contract $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}, E_{tr}) \leq 0$ and to guarantee safe obstacle avoidance as shown in the Figures 14, 15 and 16.

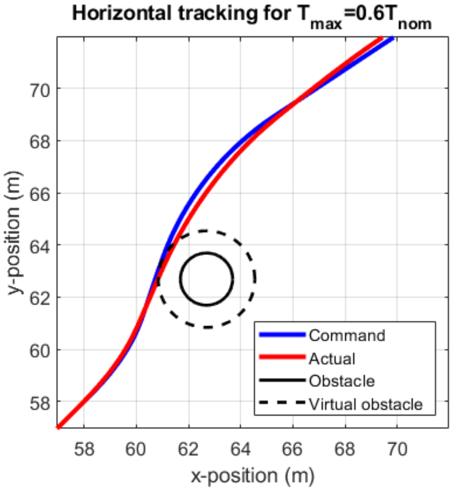

Figure 14: Testing the contract based reasoning for the obstacle avoidance scenario. The controller and navigation subsystems performance: $T_{max} = 0.6T_{nom}, D_{od} = 6m$. The trajectory parameters: $\phi_{max} = \theta_{max} = 7.03^o, V_{max} = 4.09m/s$. The expected obstacle size: $R_{obst} = 1m$. The trajectory tracking error: $E_{tr} = 0.85m$. The controller has only $0.6T_{nom}$ available to it and this necessitates a big virtual envelope around the obstacle. The UAV enters the virtual obstacle envelope but avoids the real obstacle safely. The guidance subsystem adjusts the cruise speeds accordingly.

The controller subsystem in Figure 14 has only $0.6T_{nom}$ available and as such it tracks trajectories poorly, i.e. E_{tr} is large. Poor trajectory tracking necessitates a large virtual obstacle envelope as is evident from Figure 14. As the controller subsystem performance is improved, i.e. more thrust is available to the controller ($0.8T_{nom}$ and T_{nom} in Figures 15 and 16 respectively), the size of the virtual obstacle envelope is decreasing. The UAV enters the virtual obstacle envelope for all cases (Figures 14-16) and safely avoids the real obstacle for all cases as well. Along with the controller subsystem performance, the navigation subsystem performance as well as the expected obstacle size are changing in Figures 14-16, the guidance subsystem adjusts the trajectory parameters (ϕ_{max}, θ_{max} and V_{max}) accordingly.

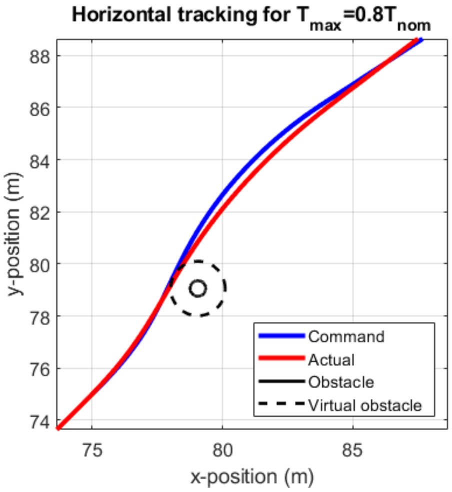

Figure 15: Testing the contract based reasoning for the obstacle avoidance scenario. The controller and navigation subsystems performance: $T_{max} = 0.8T_{nom}, D_{od} = 6m$. The trajectory parameters: $\phi_{max} = \theta_{max} = 9^o, V_{max} = 5.29m/s$. The expected obstacle size: $R_{obst} = 0.3m$. The trajectory tracking error: $E_{tr} = 0.75m$. The controller has $0.8T_{nom}$ available to it which is more than what it had in Figure 14 and as the result a smaller virtual obstacle envelope is required. The UAV enters the virtual obstacle envelope but avoids the real obstacle safely. The guidance subsystem adjusts the cruise speeds accordingly.

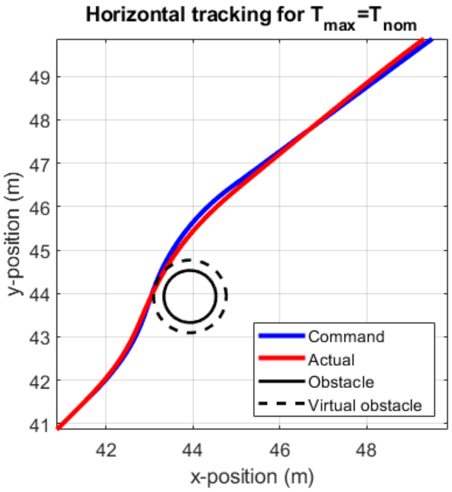

Figure 16: Testing the contract based reasoning for the obstacle avoidance scenario. The controller and navigation subsystems performance: $T_{max} = T_{nom}, D_{od} = 3m$. The trajectory parameters: $\phi_{max} = \theta_{max} = 9^o, V_{max} = 2.93m/s$. The expected obstacle size: $R_{obst} = 0.6m$. The trajectory tracking error: $E_{tr} = 0.24m$. This is the case of the nominal controller subsystem and as such the smallest virtual obstacle envelope is required. The guidance subsystem adjusts the cruise speeds accordingly.

In-flight contract implementation

To guarantee the safe UAV operation, it is desirable to satisfy the contract at all times. The guidance subsystem can be run in the background to constantly satisfy the contract $\Gamma(D_{od}, V_{max}, \phi_{max}, \theta_{max}, T_{max}, R_{obst}, E_{tr}) \leq 0$. The Figures 17 and 18 demonstrate how the contract can be constantly satisfied when the guidance subsystem is in real-time during the UAV operation.

As is evident from Figure 17, when the obstacle detection range is $3m$ and the performance of controller subsystem is nominal ($T_{max} = T_{nom}$), the guidance subsystem plans trajectories for a lower cruise velocity of around $3m/s$. Once the obstacle detection range is increased to $7m$ (the UAV has cleared a foggy area, for example), the guidance subsystem increases the cruise velocity to around $5.5m/s$. As the thrust available to the controller subsystem drops to $0.8T_{nom}$, E_{tr} is increased since the tracking performance deteriorated. Subsequently the guidance subsystem has to reduce the cruise velocity to around $5m/s$. The guidance subsystem continues to do the trajectory parameters adjustment to satisfy the contract as it is run in the background. The expected obstacle size is kept constant at $R_{obst} = 1m$ in Figure 17.

The guidance is also run constantly in the background in order to satisfy the contract in Figure 18. As the obstacle detection range is increased from $3m$ to $7m$, the maximum cruise velocity is increased to $5.3m/s$. As the obstacle size is decreased from $1m$ to $0.6m$, the maximum allowed cruise velocity is increased to $5.6m/s$. The thrust available to the controller subsystem is kept constant at $T_{max} = 0.8T_{nom}$ in Figure 18. The changes in the tracking accuracy, E_{tr}, are caused by the changes in the cruise velocity. E_{tr} is dependent on the trajectory parameters as was shown before in Figure 12.

The contract is continuously enforced in the background by the guidance subsystem assuring safe flight. As the obstacle detection range, D_{od}, changes due to changes in the environment or as the tracking performance, E_{tr}, changes, the maximum velocity, V_{max}, is adjusted accordingly to guarantee the safety.

CONCLUSIONS

A UAV may experience changes in the performance of its subsystems during the course of its operation. To ensure safe and robust performance of the UAV under these changes this paper proposed a contract based methodology. The paper also demonstrated analytical and numerical ways of obtaining the contract between guidance and navigation subsystems. It was shown that by using metadata about the performance of the subsystems such as obstacle detection range D_{od}, trajectory tracking accuracy E_{tr} and environment information, R_{obst}, the contract between the subsystems can be designed. The paper also demonstrated the safe flight performance with the contract enforced in the guidance subsystem.

Authors' contacts: Talgat Alimbayev alimbt@rpi.edu, Nicholas Moy moyn@rpi.edu, Kaushik Nallan nal-

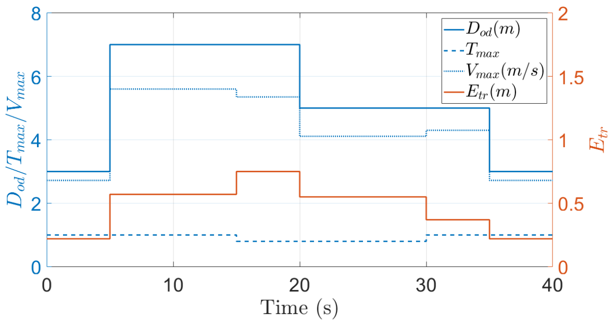

Figure 17: In-flight contract implementation. The performance of the navigation and controller subsystem are evolving over time. The environment stays the same with the expected obstacle size of $R_{obst} = 1m$. The guidance subsystem adjust the operational parameters so as to guarantee the safe operation. It does so by taking into account the performance of each of the subsystems and environment information.

lak@rpi.edu, Sandipan Mishra mishrs2@rpi.edu and A. Agung Julius agung@ecse.rpi.edu.

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 Bhagwat and Dr. William Lewis as Technical Monitors.

REFERENCES

1. Dydek, D., Annaswamy, A., and Lavretsky, E., “Adaptive Control of Quadrotor UAVs in the Presence of Actuator Uncertainties,” AIAA Infotech at Aerospace, Atlanta, GA, April 20-22, 2010.

2. Milhim, A. B., Zhang, Y., and Rabbath, C.-A., “Gain Scheduling Based PID Controller for Fault Tolerant Control of a Quad-Rotor UAV,” AIAA Infotech at Aerospace, Atlanta, GA, April 20-22, 2010.

3. Trickey, E., Church, P., and Cao, X., “Characterization of the OPAL obscurant penetrating LiDAR in various degraded visual environments,” Degraded Visual Environments: Enhanced, Synthetic, and External Vision Solutions 2013, Vol. 8737, 2013.

4. Khattak, S., Papachristos, C., and Alexis, K., “Visual-Thermal Landmarks and Inertial Fusion for Navigation in Degraded Visual Environments,” IEEE Aerospace Conference, Big Sky, MT, March 2-9, 2019.

5. Madhag, A., “Gain-Scheduled Control Based on Online Estimated Sensor Aging,” PhD Dissertation, Michigan State University, 2019.

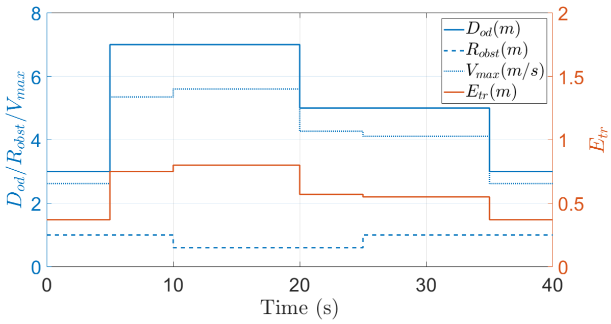

Figure 18: In-flight contract implementation. The performance of the navigation subsystem and operational environment are evolving over time. The performance of the controller subsystem stays the same with the maximum thrust available to the controller subsystem $T_{max} = 0.8T_{nom}$. The guidance subsystem adjusts the operational parameters so as to guarantee the safe operation. It does so by taking into account the performance of each of the subsystems and environment information.

6. Chamseddine, A., Zhang, Y., Rabbath, C.-A., Apkarian, J., and Fulford, C., “Model reference adaptive fault tolerant control of a quadrotor UAV,” Infotech at Aerospace, St. Louis, MO, March 29-31, 2011.

7. Sangiovanni-Vincentelli, A. L., Damm, W., and Passerone, R., “Taming Dr. Frankenstein: Contract-based design for cyber-physical systems,” *European journal of control*, Vol. 18, (3), 2012, pp. 217–238.

8. Sadraddini, S., Rudan, J., and Belta, C., “Formal Synthesis of Distributed Optimal Traffic Control Policies,” 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS), 2017.

9. Phan-Minh, T., Cai, K. X., and Murray, R. M., “Towards Assume-Guarantee Profiles for Autonomous Vehicles,” 2019 IEEE 58th Conference on Decision and Control (CDC), 2019.

10. Nuzzo, P., Xu, H., Ozay, N., Finn, J. B., Sangiovanni-Vincentelli, A. L., Murray, R. M., Donzé, A., and Seshia, S. A., “A contract-based methodology for aircraft electric power system design,” *IEEE Access*, Vol. 2, 2013, pp. 1–25.

11. 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.

12. Idres, M., Mustapha, O., and Okasha, M., “Quadrotor trajectory tracking using PID cascade control,” Materials Science and Engineering Conference Series, Vol. 270, 2017.

13. 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.

14. Murray, R. M., Rathinam, M., and Sluis, W., “Differential flatness of mechanical control systems: A catalog of prototype systems,” ASME international mechanical engineering congress and exposition, 1995.

15. Ferrin, J., Leishman, R., Beard, R., and McLain, T., “Differential flatness based control of a rotorcraft for aggressive maneuvers,” 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

16. Cheng, M., and Lee, C., “Motion Controller Design for Contour-Following Tasks Based on Real-Time Contour Error Estimation,” *IEEE Transactions on Industrial Electronics*, Vol. 54, (3), June 2007, pp. 1686–1695. DOI: 10.1109/TIE.2007.894691