Transition Trajectory Planning and Control for Quadrotor Biplanes in Obstacle Cluttered Environments

Kristoff McIntosh
Ph.D. Student
Rensselaer Polytechnic Institute
Troy, NY, USA

Jean-Paul Reddinger
Aerospace Engineer
DEVCOM Army Research Laboratory
Aberdeen Proving Ground, MD, USA

Jae Woo Kim
Ph.D. Student
Rensselaer Polytechnic Institute
Troy, NY, USA

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

ABSTRACT

This paper presents a trajectory planner and a control architecture capable of guiding a quadrotor biplane tailsitter (QRBP) through obstacle cluttered environments. The trajectory planner is formulated as an optimization problem that uses a differentially flat, point-mass model of a QRBP that considers wake effects on the aerodynamic forces generated during transition. Obstacle avoidance is realized as a state constraint in the optimization problem that defines 'no-flight zones' or regions where the QRBP cannot enter based on obstacle size and safety clearance requirements. The 6DOF control architecture is designed as a set of cascaded dynamic inversion controllers that use the aerodynamic feedforward signals produced by the trajectory planner to complete the inversion in the outer loop. To show the effectiveness of the obstacle avoidance path planning methodology, time-optimal trajectories are generated for two flight missions (the hover to forward flight and vice versa) through cluttered environments. The control architecture is validated on these two cases using a high fidelity flight dynamics simulation of a QRBP. The computational efficiency of the trajectory planner and the tracking performance of the control architecture are then empirically validated.

INTRODUCTION

Transitioning Unmanned Aerial Systems (UAS), capable of operating in and maneuvering between the vertical take-off/landing (VTOL) and fixed wing flight modes, have seen increasing interest of late (Ref. 1), (Ref. 2) due to the benefits and advantages inherent to their design, i.e. higher maneuverability, reduced take-off and landing footprint, increased endurance in hover, and larger capacity for payload delivery than their conventional counterparts.

From a control design standpoint, the development of effective and robust guidance-navigation-control (GNC) architectures for transitioning UAS for maneuvering between flight modes (hover to forward flight and vice versa) is a well-documented research problem. Prior literature on transitioning UAS (particularly for tailsitters) describe and validate elements of the GNC architecture in isolation (Ref. 3), (Ref. 4) (Ref. 5), (Ref. 6), as well as the design of the complete closed loop architecture (Ref. 7), (Ref. 8). While several advancements regarding autonomous control of transitioning UAS have been made over the last several years, they typically focus on flight mode transitions in obstacle-free scenarios. However, the types of missions a typical UAS must execute autonomously necessitate a mechanism to avoid obstacles and perform maneuvers within cluttered environments. Thus, incorporating obstacle avoidance into path planning methodologies and control architectures for transitioning UAS is crucial to their eventual real-world application.

Figure 1. CRC-20 Quad-Rotor Biplane Tailsitter (Ref. 9)

There are several well established approaches for trajectory planning and control for obstacle avoidance in autonomous UAS. For example, in (Ref. 10), Mechali et. al design a static obstacle avoidance and path planning method for a UAS based on a rectified rapidly exploring random tree (RRT*) algorithm. In (Ref. 11), Budiyanto et. al. apply a potential field

Presented at the Vertical Flight Society's 78th Annual Forum & Technology Display, Ft. Worth, Texas, USA, May 10–12, 2022.

approach to an optimal path planning algorithm of a group of UASs, which must generate feasible paths in a bounded region in space while considering static (stationary) obstacles, as well as the other UASs as dynamic(moving) obstacles. In (Ref. 12), Jayaweera et. al propose a 3D path planning technique for a multirotor UAV. The path planning architecture uses an enhanced dynamic artificial potential field (ED-APF) to track the position and velocity of a mobile ground vehicle while in an obstacle-populated environment. In (Ref. 13) and (Ref. 14), a reachibility-based control architecture is designed for a unmanned ground vehicle that computes the control law as a backwards-reachable set based on the known dynamics of the vehicle and the projected dynamics of a moving obstacle. This rechability-based controller is designed using Hamilton-Jacobi-Isaacs (HJI) reachability analysis, and is designed for onboard implementation.

The literature described above validates many suitable methods for the safe path planning and control of *VTOL* UAS flying through obstacle cluttered environments. However, many current path planning architectures for *transitioning* UAS generally do not consider obstacle avoidance in their methodologies. Thus, this paper seeks to explore the path planning problem for transitioning UAS to maneuver between the VTOL and fixed wing flight states (and vice versa) within obstacle cluttered environments. We expand on the optimal trajectory generation method for the quadrotor biplaine (QRBP) tailsitter described in (Ref. 15), (Ref. 16), which produces both an optimal flight path for a given flight mission and a set of projected feedforward state profiles useful for controller design. The scope of this paper is limited to path planning and flight control in an environment with static obstacles of known size and location. The performance of the proposed trajectory generation approach is evaluated in terms of computational performance (for real-time implementation and on-board planning), quality of trajectories obtained, and tracking performance in a high-fidelity simulation of the Common Research Configuration (CRC-20) QRBP (Fig. 1).

PROBLEM STATEMENT

Fig. 2 shows a visualization of the cluttered environment path planning scenario. Consider a QRBP with state $\vec{\mathbf{x}}$ and input $\vec{\mathbf{u}}$, whose motion is described by $\dot{\vec{\mathbf{x}}} = f(\vec{\mathbf{x}}, \vec{\mathbf{u}})$. Let χ be the feasible range of states within the state space. Further, let $p = p_1 \cup p_2 \cup \ldots p_n \in \chi$, prescribe the regions in the bounded state space that must be avoided. These regions, henceforth known as *no-fly zones*, represent regions of known size and location which the QRBP must maneuver around to perform a specific mission or a mission task element. The control problem is thus to guide the vehicle state $\vec{\mathbf{x}}$ from the specified initial flight condition to the terminal flight condition $\vec{\mathbf{x}}_0, \vec{\mathbf{x}}_f \in \chi$, respectively, such that $\vec{\mathbf{x}}(t) \notin p$. The GNC architecture for the QRBP consists of a *trajectory planner*, designed to generate the desired state trajectory $\vec{\mathbf{x}}^*$ and feedforward signals $\vec{\mathbf{u}}^*$ required to achieve the maneuver, and a *feedback controller* of the form $\vec{\mathbf{u}} = K(\vec{\mathbf{x}}, \vec{\mathbf{x}}^*)$, designed to drive $\vec{\mathbf{x}}$ along the desired trajectory. We assume all state feedback signals are readily available.

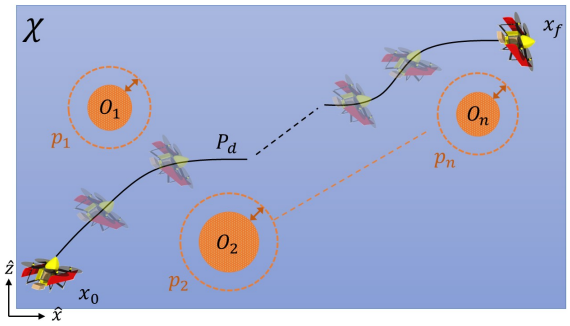

Figure 2. Visualization of Optimal Path Planning with Obstacle Avoidance.

6DOF QRBP Flight Dynamics Model for Simulation: The flight dynamics model cited from (Ref. 7) is used to represent the rigid body dynamics of a typical QRBP. This model considers gravitational, inertial, aerodynamic, and propulsive (thrust) forces. The standard 6DOF state representation of a rigid body is used for the vehicle, such that $\vec{\mathbf{x}} = [x\ y\ z\ \phi\ \theta\ \psi\ u\ v\ w\ p\ q\ r]^T$. The model also captures the effect of rotor wake over the biplane wings (and thus the aerodynamic forces and moments acting on the QRBP) using momentum theory. For a detailed description of the QRBP aerodynamic model, we refer the interested reader to (Ref. 7). A simulation model of the 20lb Common Reserach Configuration QRBP (Fig 1) is used to validate the performance of the control architecture.

The simulation model platform (Ref. 9) is based on the flight dynamics model described in (Ref. 7). The flight dynamics simulation of the CRC-20 is a full 6-DOF model consisting of 12 states and 4 control inputs. The state vector $\vec{\mathbf{x}}$, is defined using right handed inertial and body frames with a 321 (Roll-Pitch-Yaw) Euler rotation sequence. To avoid gimbal lock, the nose of the vehicle is oriented along the body y axis, changing the notions of vehicle pitch and roll to be ϕ and θ, respectively. The input vector $\vec{\mathbf{u}} = [\Omega_{COLL}\ \Omega_{LATR}\ \Omega_{LNGL}\ \Omega_{PEDL}]$ is a concatenated version of the physical control input $\bar{\mathbf{u}} = [\Omega_1\ \Omega_2\ \Omega_3\ \Omega_4]$ (the rotational speeds of each individual rotor), such that the elements of $\vec{\mathbf{u}}$ correspond to the noseward velocity v and the attitude angles θ, ϕ, and ψ, respectively.

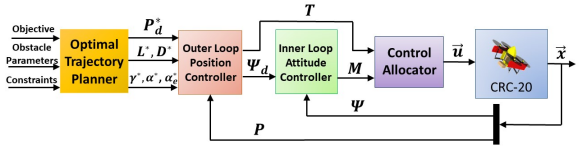

Figure 3. Control Architecture for CRC-20 QRBP.

Control Architecture Structure: Fig. 3 details the anatomy of the control architecture for the QRBP, which consists of a trajectory planner, an outer loop position controller, an inner loop attitude controller, and a control allocator. The trajectory planner generates the optimal flight path P_d^*, as well as the projected aerodynamic forces L^* (lift), D^* (drag), flight path angle γ^*, nominal angle of attack α^*, and effective angle of attack α_e^* needed to achieve desired flight path transition.

The trajectory planner considers the subset p representing the obstacles to be avoided in its formulation. Using P_d^* as a reference, the outer loop position controller uses the aerodynamic feedforward signals alongside inertial position error to calculate the vehicle's thrust command T_c and the desired attitude $\Psi_d = [\phi_d \text{ (pitch) } \theta_d \text{ (roll) } \psi_d \text{ (yaw)}]^T$ for position correction. The attitude controller corrects attitude error by adjusting the required control moments ($\mathrm{M} \triangleq [\mathscr{L} \ \mathscr{M} \ \mathscr{N}]^T$). The thrust command and control moments are then allocated to obtain the control input $\vec{\mathbf{u}} = [\Omega_1 \ \Omega_2 \ \Omega_3 \ \Omega_4]^T$.

TRAJECTORY PLANNER WITH OBSTACLE AVOIDANCE METHODOLOGY

The methodology for trajectory planning in obstacle cluttered environments is an expansion of the method described in (Ref. 16). Here, planning is cast as an optimization problem that generates the optimal target curve P_d^*, and the feedforward signals L^*, D^*, γ^*, α^*, α_e^* based on an optimization metric (cost function) specified by the vehicle user. A simplified model that accounts for the effects of rotor wake on the aerodynamic forces L and D is used in the optimization problem formulation. No-flight-zones of known size and location are defined as regional constraints in the bounded state space χ of the optimization problem for which the vehicle is not allowed to enter. The trajectory planner is described in detail in the following sections.

Simplified QRBP Model for Trajectory Planning: The simplified QRBP model considers only the longitudinal motion of the vehicle (i.e. translation in the vertical $x-z$ plane of the inertial frame and rotation (pitch) about the body frame y axis) for trajectory planning, while out of plane motion (i.e. translation along y axis of the inertial frame and roll/yaw rotations about the body frame x and z axes) is assumed to be stabilized through inner loop control. Similar to the 3DOF dynamic model of a conventional fixed-wing airplane cited in (Ref. 17), four state variables (the horizontal and vertical positions x and z, the inertial velocity V_i and the flight path angle γ) are required to fully describe the longitudinal dynamics of the QRBP in the inertial wind frame. The rotor thrust T and nominal angle of attack α are employed to steer the vehicle. Accordingly, the equations of motion are:

$$
\begin{aligned}
&\dot{x} = V_i \cos\gamma \\
&\dot{z} = V_i \sin\gamma \\
&\dot{V}_i = \frac{T\cos\alpha - L\sin(\alpha-\alpha_e) - D\cos(\alpha-\alpha_e)}{m} - g\sin\gamma \\
&\dot{\gamma} = \frac{T\sin\alpha + L\cos(\alpha-\alpha_e) - D\sin(\alpha-\alpha_e)}{mV_i} - \frac{g\cos\gamma}{V_i}
\end{aligned}
\tag{1}
$$

where:

$$
\begin{aligned}
&\alpha = \phi - \gamma \\
&V_w = 1.2\sqrt{\tfrac{T}{8\rho\pi R}} \\
&V_a = \sqrt{V_i^2 + V_w^2 + 2V_iV_w\cos\alpha} \\
&V_a\sin\alpha_e = V_i\sin\alpha \\
&L = 0.5\rho(C_{L_0} + C_{L_\alpha}\alpha_e)S_lV_a^2 \\
&D = 0.5\rho C_{D_0}S_dV_a^2
\end{aligned}
$$

Note V_w is the velocity of the wake generated by the rotors, V_a is the resultant airspeed as a result of rotor wake, and α_e is the effective angle of attack due to rotor wake (α_e represents the angle between the vehicle longitudinal axis and the airspeed vector V_a). Note that the longitudinal dynamics are represented such that the state $\boldsymbol{x} = [x, z, V_i, \gamma]^T$ and the input $\boldsymbol{u} = [T, \alpha]^T$.

To alleviate the potential computational cost for solving an optimal control problem with a nonlinear dynamic constraint, we invoke the concept of differential flatness to transform the nonlinear model ($\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u})$) into an equivalent linear form ($\dot{\boldsymbol{q}} = A\boldsymbol{q} + B\boldsymbol{v}$), where the virtual state $\boldsymbol{q}$ and the synthetic input $\boldsymbol{v}$ and are directly related to the state $\boldsymbol{x}$ and input $\boldsymbol{u}$ of the simplified nonlinear model through endogenous mappings defined by a set of flat outputs ($y = \{y_1, y_2, \ldots, y_n\}$) and a finite number of their derivatives. This process is discussed at length in previous work shown in (Ref. 16), and will be discussed with brevity here.

We choose the longitudinal and lateral inertial position to be the flat outputs of the system (i.e.$\{y_1 = x, y_2 = z\}$ and express the states of the simplified model in terms of the flat outputs as follows:

$$
\begin{aligned}
&x = y_1 && \dot{x} = \dot{y}_1 \\
&z = y_2 && \dot{z} = \dot{y}_2 \\
&V = \sqrt{\dot{y}_1^2 + \dot{y}_2^2} && \dot{V}_i = \tfrac{\dot{y}_1\ddot{y}_1 + \dot{y}_2\ddot{y}_2}{\sqrt{\dot{y}_1^2 + \dot{y}_2^2}} \\
&\gamma = \tan^-1(\tfrac{\dot{y}_2}{\dot{y}_1}) && \dot{\gamma} = \tfrac{\dot{y}_1\ddot{y}_2 - \dot{y}_2\ddot{y}_1}{\dot{y}_1^2 + \dot{y}_2^2}
\end{aligned}
\tag{2}
$$

We use these flat outputs to propagate the simplified model by declaring a virtual state $q = \begin{bmatrix} x & \dot{x} & z & \dot{z} \end{bmatrix}^T$ and a synthetic input $v = \begin{bmatrix} \ddot{x} & \ddot{z} \end{bmatrix}^T$, and relating them with the linear ODE $\dot{q} = Aq + Bv$, where the matrices A and B are a simple chain of integrators.

Finally, the original dynamics of the system are preserved by enforcing the following implicit expression of the simplified model (derived from Eq. 1) as a kinematic state constraint:

$$
\begin{aligned}
&T\cos(\alpha) - L\sin(\alpha-\alpha_e) - D\cos(\alpha-\alpha_e) - ma_\top = 0 \\
&T\sin(\alpha) + L\cos(\alpha-\alpha_e) - D\sin(\alpha-\alpha_e) - ma_\perp = 0
\end{aligned}
\tag{3}
$$

where: $\quad a_\top = \frac{\dot{x}\ddot{x} + \dot{z}(\ddot{z}+g)}{\sqrt{\dot{x}^2+\dot{z}^2}}, \qquad a_\perp = \frac{\dot{x}(\ddot{z}+g) - \dot{z}\ddot{x}}{\sqrt{\dot{x}^2+\dot{z}^2}}$

Formulation of Optimization Problem for Trajectory Generation through Cluttered Environments: Following the derivation of the differentially flat dynamic model, the optimization problem for trajectory generation through cluttered environments is formulated as the following standard nonlinear problem (NLP)

$$\underset{q,\,v}{\arg\min} \quad J = \int_{t_0}^{t_f} dt, \tag{4a}$$

$$s.t \quad \dot{q} = Aq + Bv, \tag{4b}$$

$$q(t_0) = q_0, \tag{4c}$$

$$q(t_f) = q_f, \tag{4d}$$

$$g_x(q,v) \in \chi, g_u(q,v) \in \mu, \tag{4e}$$

$$g_x(q,v), \notin p \in \chi \tag{4f}$$

where Eq. 4a is the objective function (chosen to generate minimum time flight paths), Eq. 4b is the dynamic constraint of the problem (represented by the differentially flat model described previously), Eqs. 4c and 4d are the boundary constraints that enforce the desired initial and terminal states of the problem, and Eq. 4e are the constraints placed on the virtual state and synthetic input that constrict the vehicle to act within its flight envelope. Eq. 4f are the constraints on the state space χ that define the obstacle set p that the vehicle cannot enter. Once discretized, this optimization problem is numerically solved in real time using the NLP solver IPOPT to generate the QRBP flight path and feedforward signals.

Obstacle Avoidance Constraints The no-flight zones are interpreted as a constraint on the allowable states within the feasible state-space χ. These constraints are defined as a set of circular regions $p = \{p_1, p_2, p_3, \ldots, p_n\}$ within χ where the vehicle cannot enter as it completes the specified flight mission. Each no-flight zone expressed as:

$$p_i = (x - x_{O_i})^2 + (z - z_{O_i})^2 - (r_{O_i} + r_{C_i})^2, \quad i = 1,2,3,\ldots,n \tag{5}$$

where the point (x_{O_i}, z_{O_i}) represents the center of mass of each obstacle, r_{O_i} is the distance from each obstacle center of mass to its longest edge, and r_{C_i} is a clearance space between each obstacle edge and the boundary of p_i, implemented to artificially inflate the no-flight zone to account for position tracking error. Once each no-flight zone is defined, the state constraint $g_x(q,v) \notin p = p_1 \cup p_2 \cup \ldots p_n \in \chi$ is enforced, requiring the trajectory generator to plan the optimal path through the set of no-flight zones within the bounded state space.

CONTROLLER DESIGN

Outer Loop Position Controller: Once the optimal path planner solves the obstacle avoidance problem, the outer loop position controller takes in the optimal flight path P_d^* (along with its first and second derivatives) and the aerodynamic feedforward signals $(L^*, D^*, \gamma^*, \alpha^*, \alpha_e^*)$ to generate the thrust command T_c and the desired attitude Ψ_d necessary to steer the vehicle along the flight path. These signals are generated by way of dynamic inversion of the 2D point-mass model used for trajectory generation shown in Eq. 1, expressed in the inertial frame, assuming that the prescribed aerodynamic forces approximate the actual aerodynamic forces for all time. The outer loop assumes that any thrust or attitude command (within a bounded set) is instantaneously attainable.

To compute T_c and Ψ_d, we invert Eq. 6 below, which depicts the 2D point-mass model for trajectory generation in the inertial frame:

$$\begin{bmatrix} T\cos\phi \\ T\sin\phi \end{bmatrix} - \underbrace{\begin{bmatrix} L^*\sin(\gamma^*+\alpha^*-\alpha_e^*) + D^*\cos(\gamma^*+\alpha^*-\alpha_e^*) \\ L^*\cos(\gamma^*+\alpha^*-\alpha_e^*) - D^*\sin(\gamma^*+\alpha^*-\alpha_e^*) \end{bmatrix}}_{\begin{bmatrix} F_{A_y}^* \\ F_{A_z}^* \end{bmatrix}} + \begin{bmatrix} 0 \\ mg \end{bmatrix} = m \begin{bmatrix} \ddot{y} \\ \dot{z} \end{bmatrix}. \tag{6}$$

Inverting Eq. 6, assuming $F_A^* \approx F_A$, provides the following expressions for T_c and ϕ_c:

$$\begin{aligned} T_c &= \sqrt{(F_{A_y}^* + m\ddot{y}_c)^2 + (F_{A_z}^* + m(\ddot{z}_c - g)^2} \\ \phi_d &= \tan^{-1}\left[\frac{F_{A_z}^* + m(\ddot{z}_c - g)}{F_{A_y}^* + m\ddot{y}_c}\right], \end{aligned} \tag{7}$$

where $\ddot{y}_c$ and $\ddot{z}_c$ are the inertial acceleration commands for longitudinal motion and altitude, respectively. To generate $\ddot{y}_c$ and $\ddot{z}_c$, we define a control law in terms of the desired second order position error dynamics for the vehicle in the inertial frame, such that

$$\ddot{X}_c = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{z}_c \end{bmatrix} = \begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \\ \ddot{z}_d \end{bmatrix} + \mathbf{K_{D_X}} \begin{bmatrix} \dot{x}_d - \dot{x} \\ \dot{y}_d - \dot{y} \\ \dot{z}_d - \dot{z} \end{bmatrix} + \mathbf{K_{P_X}} \begin{bmatrix} x_d - x \\ y_d - y \\ z_d - z \end{bmatrix}, \tag{8}$$

where $\mathbf{K_{P_X}}$ and $\mathbf{K_{D_X}}$ are the proportional and derivative gain matrices for the control law, respectively. *Remark:* Note that the outer loop (like the trajectory planner) assumes the QRBP is limited to planar motion, generating only a thrust command T_c and the desired pitch angle ϕ_d for position correction. Thus, we schedule the remaining desired attitude profiles θ_d (roll) and ψ_d (yaw), as well as the desired lateral inertial position x_d to zero for all time. Feeding the inertial acceleration command $\ddot{X}_c$ into Eq. 7 generates the T_c and Ψ_d for the vehicle, which are then sent to the control allocator and the inner loop controller, respectively.

Inner Loop Attitude Controller: The inner loop attitude controller takes in the desired attitude Ψ_d prescribed by the position controller and produces the required control moments M to correct for attitude error. The attitude controller is derived using the dynamic inversion approach described in (Ref. 7), where the rotational dynamics of the vehicle are simplified to that of a quadrotor. Second order (attitude) error dynamics are used to generate the required vehicle rotational acceleration:

$$\ddot{\Psi}_c = \begin{bmatrix} \ddot{\phi}_c \\ \ddot{\theta}_c \\ \ddot{\psi}_c \end{bmatrix} = \begin{bmatrix} \ddot{\phi}_d \\ \ddot{\theta}_d \\ \ddot{\psi}_d \end{bmatrix} + \mathbf{K_{D_\psi}} \begin{bmatrix} \dot{\phi}_d - \dot{\phi} \\ \dot{\theta}_d - \dot{\theta} \\ \dot{\psi}_d - \dot{\psi} \end{bmatrix} + \mathbf{K_{P_\psi}} \begin{bmatrix} \phi_d - \phi \\ \theta_d - \theta \\ \psi_d - \psi \end{bmatrix}, \tag{9}$$

where $\mathbf{K_{P_\psi}}$ and $\mathbf{K_{D_\psi}}$ are diagonal matrices containing the controller gains for attitude and attitude rate. Note that $\ddot{\Psi}_d$ and $\dot{\Psi}_d$ are calculated in real time by, numerically differentiating and filtering Ψ_d from the outer loop (as in (Ref. 7)), and $\dot{\Psi}$, Ψ are feedback signals. The required control M moments are then calculated from $\ddot{\Psi}_c$ using

$$\begin{bmatrix} \dot{p}_c \\ \dot{q}_c \\ \dot{r}_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \ddot{\phi}_c \\ \ddot{\theta}_c \\ \ddot{\psi}_c \end{bmatrix} + \begin{bmatrix} 0 & 0 & c_\theta \dot{\theta} \\ 0 & s_\phi \dot{\phi} & -s_\phi s_\theta \dot{\theta} - c_\phi c_\theta \dot{\phi} \\ 0 & c_\phi \dot{\phi} & -s_\theta c_\phi \dot{\theta} + s_\phi c_\theta \dot{\phi} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{10}$$

$$\mathbf{M} = \begin{bmatrix} \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{p}_c + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q}_c + (I_{xx} - I_{zz})pr \\ I_{zz}\dot{r}_c + (I_{yy} - I_{xx})pq \end{bmatrix}. \tag{11}$$

Control Allocator: The thrust command T_c and control moments M are converted to the control input $\vec{u}$ using the following allocation:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ -d_L k_T & -d_L k_t & d_L k_t & d_L k_t \\ k_q & -k_q & k_q & -k_q \\ -d_N k_T & d_N k_T & d_N k_T & -d_N k_T \end{bmatrix}^{-1} \begin{bmatrix} T_c \\ \mathscr{L} \\ \mathscr{M} \\ \mathscr{N} \end{bmatrix}, \tag{12}$$

where $k_T = \rho \pi R^4 C_T$, $k_Q = \rho \pi R^5 C_T$, ρ is the atmospheric air density, R is the rotor radius, C_T and C_Q are the rotor thrust and rotor hub torque coefficients, respectively. d_L and d_N are the longitudinal and lateral moment arms, respectively.

RESULTS

This section describes the results pertaining to the computational efficiency of the trajectory planner with the proposed obstacle avoidance methodology and the performance of the proposed control architecture. We present the optimal trajectories and the tracking performance for two flight missions in the following sections: (1) a transition from 3kt (1.54 m/s) vertical ascent to 25kt (12.86 m/s) forward flight ($H \rightarrow FF$) through a field of 3 no-flight zones, and (2) a 25kt forward flight to hover ($FF \rightarrow H$) transition with 2 no-flight zones.

Hover to Forward Flight with 3 No-Flight Zones

For the 3kt ascent to 25kt forward flight mission, the vehicle is tasked with flying through 3 circular no-flight zones, the geometries of which are detailed in Table 1.

Transition Flight Path and Aerodynamic Feedforward Profiles Figure 4 shows the resulting time-optimal flight path generated by the path planner compared to a feasible initial guess generated without consideration of obstacle avoidance.

Table 1. $H \rightarrow FF$ Mission No-Flight Zone Geometries

	x_{obs}, z_{obs} (m)	r_{obs} (m)	r_c (m)
NFZ 1	(6,3)	0.5	0.5
NFZ 2	(8,8)	0.5	0.5
NFZ 3	(2,4)	0.5	0.5

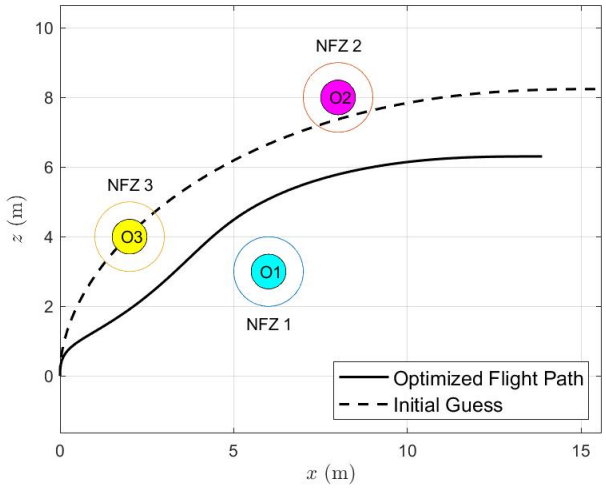

Figure 4. QRBP $H \rightarrow FF$ Optimal Obstacle Avoidance Flight Path

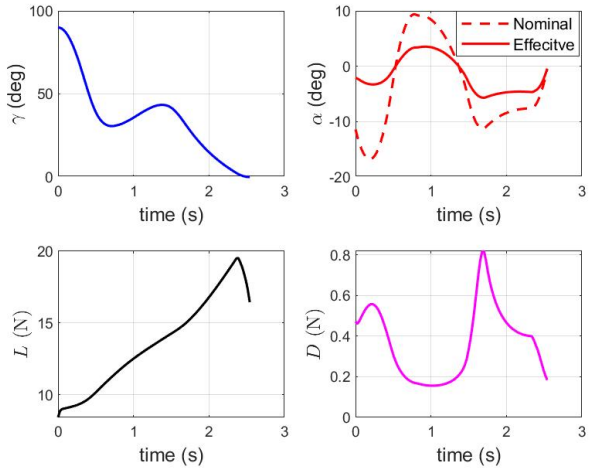

Figure 5. QRBP $H \rightarrow FF$ Mission Aerodynamic Feedforward Profiles

Figure 5 shows the corresponding the aerodynamic feedforward profiles for dynamic inversion.

The reported average computational time for this trajectory is 0.231s, while the prescribed time of flight is 2.51s. Observing Figure 4, it can be seen that the optimized flight path avoids each no-flight zone by a significant margin (by at least 2 meters).

Tracking Performance Figure 6 shows the flight path tracking for the obstacle avoidance $H \rightarrow FF$ flight mission. Figures 7 and 8 show the tracking of the inertial position and inertial velocity, respectively. Figure 9 shows the pitch angle tracking and the thrust command T_c into the vehicle (compared to the projected feedforward T^*).

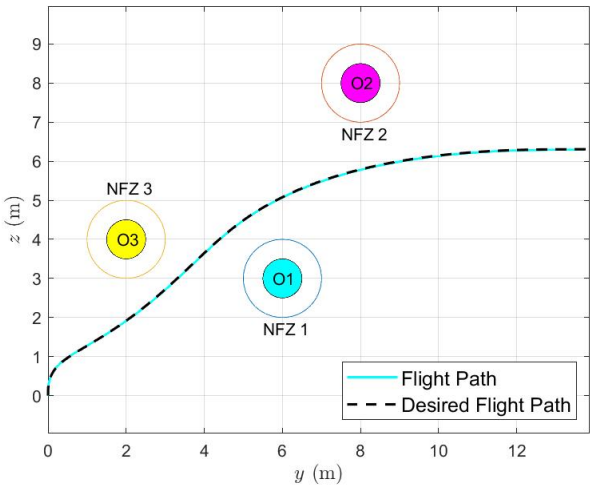

Figure 6. QRBP $H \rightarrow FF$ Mission Flight Path Tracking Performance

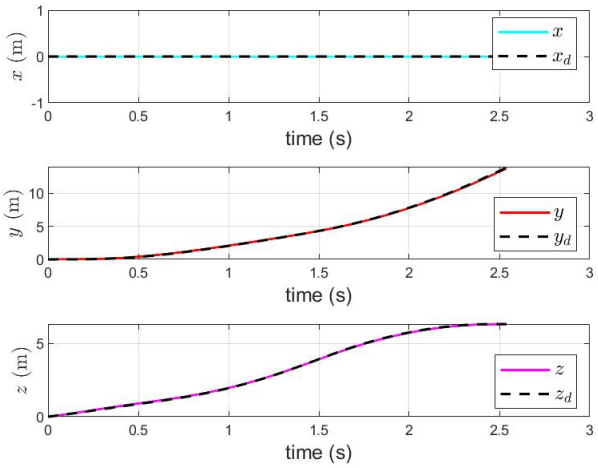

Figure 7. QRBP $H \rightarrow FF$ Mission Inertial Position Tracking

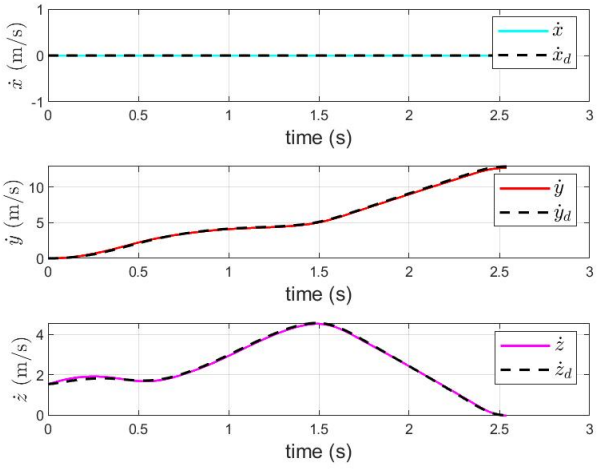

Figure 8. QRBP $H \rightarrow FF$ Mission Inertial Velocity Tracking

Overall tracking performance of the dynamic inversion controller is shown to be adequate for the $H \rightarrow FF$ flight mission, with a maximum tracking error of 0.11 m, 0.15 m/s, and 1.7^o for position, velocity, and attitude tracking, respectively.

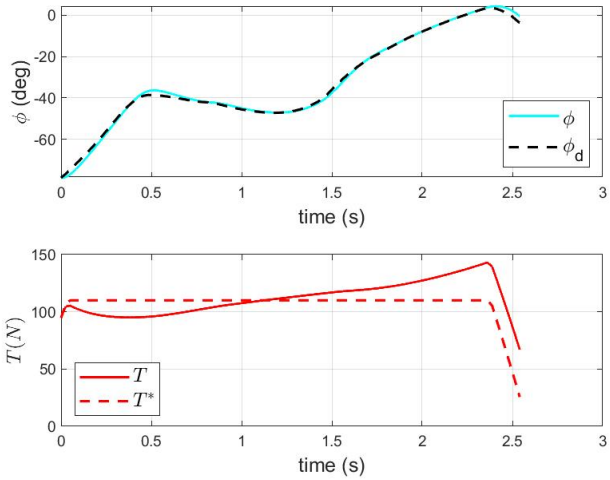

Figure 9. QRBP $H \rightarrow FF$ Mission Attitude Tracking and Thrust Input

Forward Flight to Hover with 2 No-Flight Zones

For the 25kt forward flight to hover mission, the vehicle must fly through 2 circular no-flight zones, the geometries of which, with respect to the obstacle limit, are detailed in Table 2. Note that for this transition, the terminal altitude is constrained to be the same as the initial altitude, requiring the vehicle to complete the $FF \rightarrow H$ transition without gaining altitude.

Table 2. $FF \rightarrow H$ Mission No-Flight Zone Geometries

	x_{obs}, z_{obs} (m)	r_{obs} (m)	r_c (m)
NFZ 1	(20,12.5)	1.5	1.5
NFZ 2	(40,12.5)	1.5	1.5

Transition Flight Path and Aerodynamic Feedforward Profiles As with the previous mission, we show the optimal flight path for the $FF \rightarrow H$ obstacle avoidance mission (compared to a feasible initial guess that ignores the obstacles) in Figure 10. Figure 11 shows the corresponding aerodynamic feedforward profiles for the control architecture. The reported average computational time for this case is 1.021s, while the prescribed time of flight is 5.912s.

Comparing the optimal trajectory of the $FF \rightarrow H$ obstacle avoidance case to the $H \rightarrow FF$ trajectory, it can be seen that the trajectory planner has allotted much less clearance between the no-flight zones in the former compared to the latter. This is to be expected, the vehicle's control authority in γ is lower due to the high initial forward flight speed. Thus, when assessing tracking performance, the controller must be carefully tuned so as to avoid collision with the first obstacle (should the vehicle enter the no-flight zone).

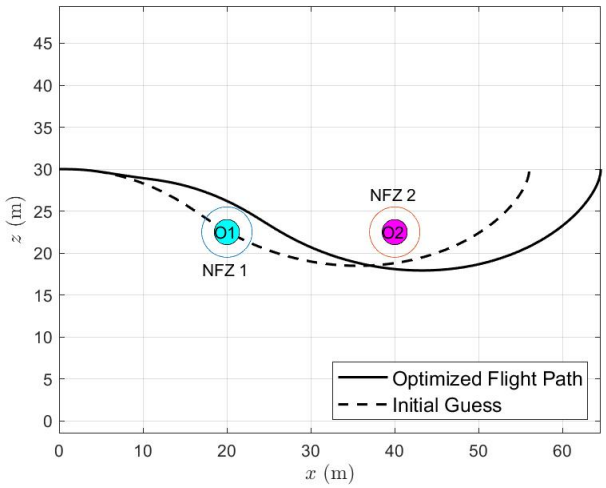

Figure 10. QRBP $FF \rightarrow H$ Optimal Obstacle Avoidance Flight Path

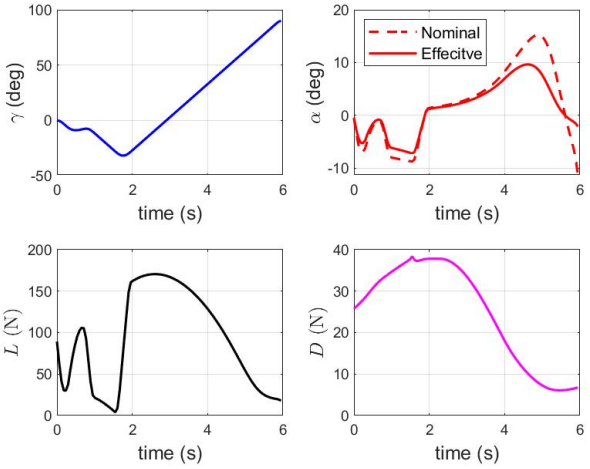

Figure 11. QRBP $FF \rightarrow H$ Mission Aerodynamic Feedforward Profiles

Tracking Performance We show the vehicle's flight path tracking performance in Figure 12. Figures 13 and 14 show the tracking of the inertial position and velocity, respectively. Figure 15 shows the pitch angle tracking and the thrust command into the vehicle.

Based on the tracking performance for the $FF \rightarrow H$ flight mission, we note that, despite completing the mission, controller performance has degraded compared to the $H \rightarrow FF$ case, with a maximum tracking error of 5.68 m, 3.96 m/s, and 6.2^o for position, velocity, and attitude tracking, respectively. However, noting the flight path tracking in Figure 12, we see that while the vehicle enters NFZ 1, it still manages to avoid the obstacle inside of it, with the closest approach between the vehicle and the obstacle being 1.31m. While the control architecture executed the obstacle avoidance mission, re-tuning is necessary in order to eliminate the lag response in the inertial velocity channel on the outer loop.

CONCLUSIONS

This paper proposed an optimal trajectory generation methodology for the guidance of a quadrotor biplane tailsitter through obstacle cluttered environments. The trajectory generation and path tracking of two flight missions ($H \rightarrow FF$, $FF \rightarrow H$

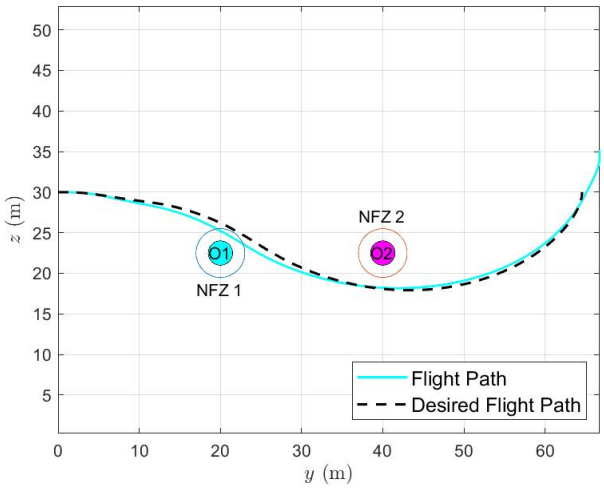

Figure 12. QRBP $FF \rightarrow H$ Mission Flight Path Tracking Performance

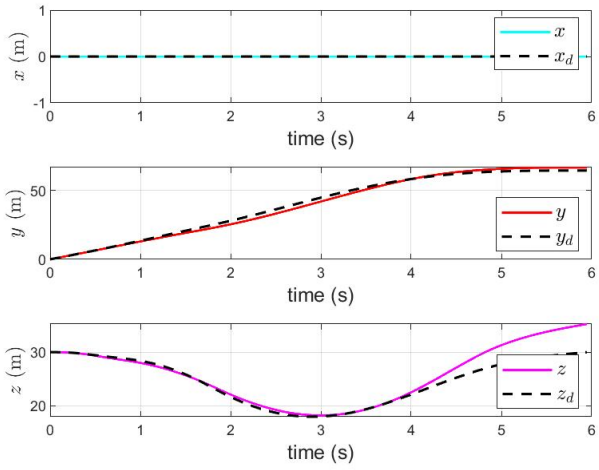

Figure 13. QRBP $FF \rightarrow H$ Mission Inertial Position Tracking

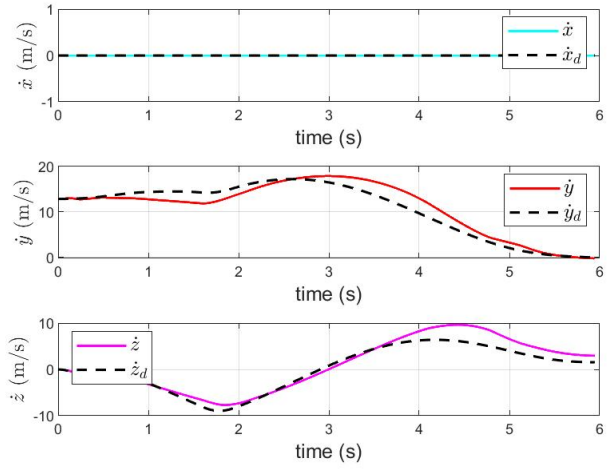

Figure 14. QRBP $FF \rightarrow H$ Mission Inertial Velocity Tracking

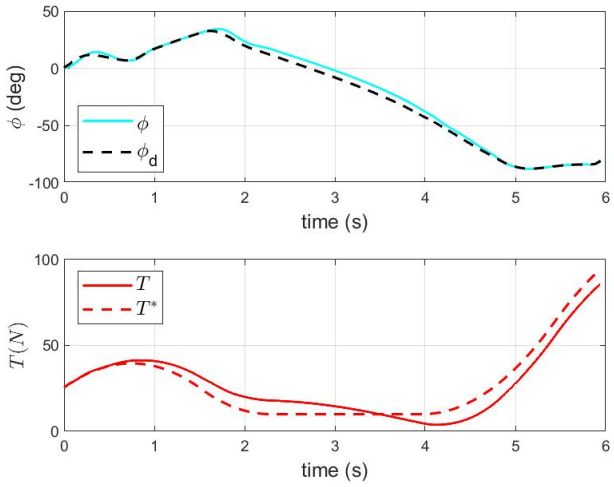

Figure 15. QRBP $FF \rightarrow H$ Mission Attitude Tracking and Thrust Input

transitions) were shown and analyzed for computational efficiency and tracking performance. The following findings are establish and reported: (1) The proposed control architecture is capable of generating transition trajectories through cluttered environments with reasonable computational efficiency, and (2) the control architecture is capable of following the prescribed trajectories tightly enough to avoid the specified obstacles. Based on the analysis of the results presented in this paper, there are several open questions that must be addressed to improve the proposed architecture, particularly improving the computational efficiency of the architecture for more complex $FF \rightarrow H$ missions and improving the performance of the proposed controller for more robust tracking.

ACKNOWLEDGMENTS

This work is collaborative research between the DEVCOM Army Research Laboratory and Rensselaer Polytechnic Institute Center for Mobility and Vertical Lift (MOVE). The work was sponsored in part by the Office of Naval Research (ONR), under contract number N00014-16-1-2705, and in part under the Army/Navy/NASA Vertical Lift Research Center of Excellence (VLRCOE) Program, grant number W911W61120012, with Dr. Mahendra Bhagwat.

Author contact:
Kristoff McIntosh, mcintk3@rpi.edu;
Jean-Paul Reddinger, jean-paul.f.reddinger.civ@army.mil;
Jae Woo Kim, kimj49@rpi.edu;
Sandipan Mishra, mishrs2@rpi.edu

REFERENCES

1. Narayan, P., Wu, P., Campbell, D., and Walker, R., "An Intelligent Control Architecture for Unmanned Aerial Systems (UAS) in the National Aerospace System (NAS)," 2nd Australian Unmanned Air Vehicle Systems Conference, Melbourne, Australia, March 2007.

2. Zaludin, Z., and Harituddin, A. S. M., "Challenges and Trends of Changing from Hover to Forward Flight for a Converted Hybrid Fixed Wing VTOL UAS from Automatic Flight Control System Perspective," IEEE 9th International Conference on System Engineering and Technology, Shah Alam, Malaysia, October 2019.

3. Hrishikeshavan, V., Bawek, D., Rand, O., and Chopra, I., "Control of a Quad Rotor-Biplane Micro Air Vehicle in Transition from Hover to Forward Flight," American Helicopter Society Specialists' Meeting for Unmanned Rotorcraft, January 2013.

4. Hrishikeshavan, V., Yeo, D., , and Chopra, I., "Onboard Flow Sensing in a Quadrotor Biplane Micro Air Vehicle for Transition between Hover and Steady-Level Flight," American Helicopter Society Specialists' Meeting for Unmanned Rotorcraft, January 2015.

5. Swarnkar, S., Parwana, H., Kothari, M., and Abhishek, "Development of Flight Dynamics Model and Control of Biplane-Quadrotor UAV," 2018 AIAA Guidance, Navigation, and Control Conference, January 2018.

6. Flores, A., Montes, O., and Flores, G., "A simple controller for the transition maneuver of a tail-sitter drone," 2018 IEEE Conference on Decision and Control, Miami Beach, FL, December 2018.

7. Swarnkar, S., Parwana, H., Kothari, M., and Abhishek, A., "Biplane-Quadrotor Tail-Sitter UAV: Flight Dynamics and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 41, January 2018, pp. 1–19. DOI: 10.2514/1.G003201

8. Li, Z., Zhou, W., and Hao, L., "Robust Controller Design for a Tail-sitter UAV in Flight Mode Transitions," 2018 IEEE 14th International Conference on Control and Automation, Anchorage, AK, June 2018.

9. Reddinger, J. P., McIntosh, K., Zhao, D., and Mishra, S., "Modeling and Trajectory Control of a Transitioning Quadrotor Biplane Tailsitter," Vertical Flight Society 75th Annual Forum Proceedings, May 2019.

10. Mechali, O., Xu, L., Wei, M., Benkhaddra, I., Guo, F., and Senouchi., A., "A Rectified RTT* with Efficient Obstacles Avoidance Method for UAV in 3D Environment," IEEE International Conference over CYBER Technology in Automation, Control, and Intelligent Systems, July 2019.

11. Budiyanto, A., Cahyadi, A., Adji, T. B., and Wahyunggoro., O., "UAV Obstacle Avoidance Using Potential Field under Dynamic Environment," International Conference on Control, Electronics, Renewable Energy and Communications, August 2015.

12. Jayaweera, H. M. P. C., and Hanoun, S., "UAV Path Planning for Reconnaissance and Look-Ahead Coverage Support for Mobile Ground Vehicles," *Sensors*, Vol. 21, 2021 2021, pp. 4595–4620. DOI: https://doi.org/10.3390/s21134595

13. Dabadie, C., Kaynama, S., and Tomlin, C. J., "A practical reachability-based collision avoidance algorithm for sampeld-data systems: Application to ground robots," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014. DOI: 10.1109/IROS.2014.6943149

14. Herbert, S. L., Bansal, S., Ghosh, S., and Tomlin, C. J., "Reachability-Based Safety Guarantees using Efficient Initializations," 2019 IEEE 58th Conference on Decision and Control (CDC), 2019. DOI: 10.1109/CDC40024.2019.9020575

15. McIntosh, K., Reddinger, J.-P., Zhao, D., and Mishra., S., "Optimal Trajectory Generation of a Quadrotor Biplane Tailsitter," Vertical Flight Society 76th Annual Forum Proceedings, October 2020.

16. McIntosh, K., Reddinger, J. P., Zhao, D., and Mishra., S., "Optimal Trajectory Generation for Transitioning Quadrotor Biplane Tailsitter Using Differential Flatness," Vertical Flight Society 77th Annual Forum Proceedings, May 2021.

17. Weitz, L. A., "Derivation for a Point-Mass Aircraft Model Used for Fast-Time Simulation," MITRE Technical Report, November 1995.