

Optimal Trajectory Generation for Transitioning Quadrotor Biplane Tailsitter Using Differential Flatness

Kristoff McIntosh

Graduate Student

Rensselaer Polytechnic Institute
Troy, NY, USA

Jean-Paul Reddinger

Aerospace Engineer

DEVCOM Army Research Laboratory
Aberdeen Proving Ground, MD, USA

Di Zhao

Ph.D. Candidate

Rensselaer Polytechnic Institute
Troy, NY, USA

Sandipan Mishra

Associate Professor

Rensselaer Polytechnic Institute
Troy, NY, USA

ABSTRACT

This paper presents the development of an optimization-based trajectory planner for the autonomous transition of a quadrotor biplane tailsitter (QRBP) between the flight modes of hover to forward flight and forward flight to hover. The trajectory planner is formulated as an optimization problem with an embedded dynamic model of the QRBP, vehicle design constraints (e.g. power), physical constraints (e.g. stall) and initial/terminal states for the transition. A differentially flat reformulation is employed to reduce the computational cost of the trajectory planner for on-board mission planning. The solution of this problem yields time-optimal state and input trajectories for transition. Using this trajectory planner, we generate trajectories for various transition flight missions (from hover to forward flight and vice versa) under various constraints. Further, we demonstrate how the proposed algorithm can also be used to assess the agility of a vehicle in terms of minimum space required to perform a specific maneuver or transition, given physical design constraints (such as maximum power). Finally, we demonstrate trajectory tracking on a high fidelity simulation of a QRBP with a cascaded dynamic-inversion based controller with a control blending strategy between the quadcopter and forward-flight control modes, for hover to forward flight.

NOTATION

x, y	inertial position
z	inertial altitude
V	inertial velocity
γ	flight path angle
V_w	rotor wake velocity
V_a	airspeed
α	geometric angle of attack (wing)
α_e	effective angle of attack (wing)
ϕ	pitch angle
L	lift force on wing
D	drag force on wing
T	thrust
ρ	air density (sea level)
R	rotor disk radius
C_{L_0}	lift curve intercept
C_{L_a}	lift curve slope
C_{D_0}	drag curve intercept
S_l	aerodynamic wing area
S_d	aerodynamic fuselage area
m	mass of transitioning-UAS

INTRODUCTION

Transitioning Unmanned Aerial Systems (UAS) are a class of aerial vehicles capable of operating in and transitioning between the vertical takeoff and landing and fixed wing flight regimes. These hybrid aircraft are specifically designed to capitalize on the strengths of these flight modes; combining the ability of electrically powered vertical take-off and landing (e-VTOL) vehicles to hover and climb, with the efficiency of fixed wing aircraft in forward flight. These vehicles leverage the benefits of both flight modes: high maneuverability, reduced take-off and landing footprint, increased endurance in hover, and larger capacity for payload delivery. Because of these advantages, transitioning UAS are seeing growing interest in both civilian (Ref. 1) and military applications (Ref. 2).

One of the primary challenges in transitioning UAS design is the development of effective and robust guidance-navigation-control (GNC) architectures for the critical transition phase between flight regimes. As a result, there has been significant interest in designing controllers for transitioning UAS (specifically those of the tail-sitter design). In Refs. 4 and 5, a control law was developed and implemented on a micro quadrotor biplane (QRBP) tailsitter that used a quaternion-based framework to express vehicle rotation, specifically to address the issue of gimbal-lock. Ref. 6 proposed a model-based transi-



Figure 1. 20 lb Common Research Configuration (CRC-20) QRBP (Ref. 3)

tion controller for a tailsitter drone based on Lyapunov stability. The controller was then simulated on a 2-dimensional representation of the vehicle dynamics based on body-frame variables. In Refs. 7 and 8, a control architecture based on the principles of nonlinear dynamic inversion and state feedback was implemented on tailsitter drones for aggressive, robust inner-loop feedback. For navigation and sensing, Ref. 9 developed an inner loop control strategy for a micro QRBP using an on-board flow sensor instrumentation for sensing attitude.

The efforts described above have concentrated primarily on effective inner-loop control methodologies for transitions from hover to forward flight and vice versa. However, literature on effective trajectory design and outer-loop control for autonomous transition is relatively sparse. Most guidance through transition is determined either by a human pilot (Ref. 1) or transition trajectories are obtained by heuristically generating simple profiles of the position and velocity variables (Refs. 3, 10). In Ref. 11, an approach for generation of transition trajectories was developed for a quadrotor biplane tailsitter by solving an optimization problem with dynamic constraints derived from a simplified reduced order dynamic model and explicit physical constraints on maximum allowed power, torque, and stall angle. However, while the proposed approach was proven effective in generating optimal transition trajectories for the QRBP, the transition dynamics were expressed in a strongly coupled non-linear form, resulting in increased computational cost for trajectory generation. This naturally makes on-the-fly implementation very difficult. Furthermore, in some cases, the trajectories generated exhibited poor numerical conditioning, leading to oscillations in the planned trajectory.

Prior literature has shown that the computational burden of optimization-based path planning problems (in robotics as well as UASs) can be reduced significantly by suitably reformulating the problem using differential flatness. For example, in Refs. 12 and 13, Chamseddine used a differentially flat dynamic model for the optimal trajectory planning of a quadrotor UAS, which was shown to generate time-optimal trajectories at a tenth of the computational cost compared to optimization using the standard non-linear model. In Ref. 14, Zhao leveraged differential flatness to compute time-optimal trajectories for a helicopter for shipboard landing, which also reduced computational time by $7\times$ as compared to using the

standard non-linear model directly in the optimizer. These applications suggest that a flatness-based approach to trajectory generation can allow for effective on-board outer-loop control of UAS through the reduction of computational time, while guaranteeing both optimality and feasibility of planned trajectories. Thus, this paper extends the previous research in Ref. 11 by developing a differentially flat model of the QRBP transition dynamics, which is then used in a numerical optimizer to generate optimal trajectories. We evaluate the performance of the proposed differential flatness based trajectory generation approach in terms of computational performance (for real-time implementation and on-board planning), quality of trajectories obtained and finally tracking performance in a high-fidelity simulation of a CRC-20 (Fig. 1).

PROBLEM STATEMENT

The general GNC architecture for a transitioning UAS is shown in Fig. 2. Typical control architectures for transitioning UAS consist of an outer loop position controller and an inner loop attitude controller (Note: position $P \triangleq [x \ y \ z]^T$, and attitude $\Psi \triangleq [\phi \ \theta \ \psi]^T$, respectively). Reference state trajectories $\mathbf{X}^*(\cdot)$ produced by the trajectory planner are fed into the outer loop controller, which corrects for error in inertial position by generating the desired attitude angles ϕ (pitch), θ (roll), ψ (yaw) to be tracked by the inner loop, as well as the required control thrust T needed to correct for position. The inner loop corrects for error in attitude by generating the required control moments ($M \triangleq [L \ M \ N]^T$) necessary for attitude correction. The control force and moments are then allocated to the required control input u given to the vehicle. For further details, we refer the reader to the *Controller Design* section below.

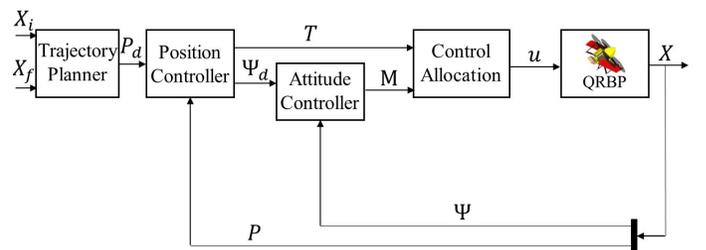


Figure 2. Cascaded Control Architecture for the QRBP.

Given a set of initial and terminal flight conditions \mathbf{x}_0 , \mathbf{x}_f (e.g. hover to forward flight), the objective of the trajectory planner is to generate the necessary state and input trajectories ($\mathbf{x}^*(t)$ and $\mathbf{u}^*(t)$, respectively), that will allow a QRBP to transition between \mathbf{x}_0 and \mathbf{x}_f , subject to the vehicle dynamics and physical constraints. The planned path for the trajectories is assumed to be in the 2-dimensional vertical plane, while the out-of-plane *planned variables* are held constant. A schematic detailing a visual representation of a transition trajectory can be seen in Fig. 3.

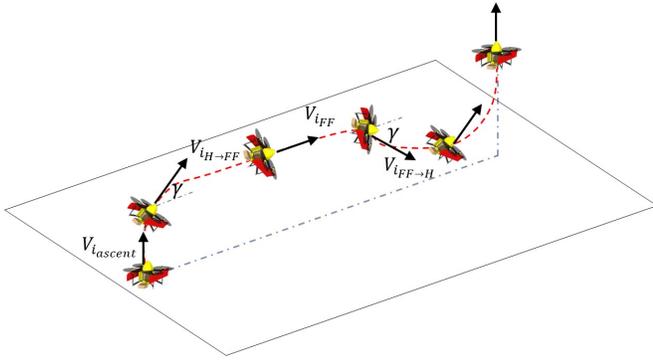


Figure 3. Visualization of Transition Flight Paths.

In Ref. 11, the trajectory planner is posed as an optimization problem designed around a simplified dynamic model for trajectory generation. The goal of the trajectory planner is to minimize the flight time of the transition trajectory, although alternative cost functions such as fuel, etc. may also be considered in the same framework. The constraints for the problem are governed by the QRBP dynamics, physical limitations, etc. In general, this problem can be expressed the an optimization problem described in Problem 1 below.

$$\begin{aligned}
 \arg \min_{x, u} \quad & J = f(x, u, t), \quad \text{cost function} \\
 \text{s.t.} \quad & \dot{x} = \mathbf{f}(x, u), \quad \text{dynamics constraints} \\
 & x(t_0) = x_0, \quad \text{initial state constraint} \\
 & x(t_f) = x_f, \quad \text{terminal state constraint} \\
 & x \in \mathbf{X} \quad u \in \mathbf{U}, \quad \text{state and input constraints}
 \end{aligned} \tag{1}$$

Problem 1 is a free end time optimization problem with constraints, which cannot be solved analytically. To obtain a numerical solution, Problem 1 is discretized (in time) into a nonlinear programming (NLP) problem. However, the nonlinear and coupled dynamics in Problem 1 results in significant computational cost (and in some cases, failure to converge) of the trajectory planner as well as numerical oscillations in the solution. In this paper, we propose a differentially flat representation of the simplified transition dynamics that can be used in the optimization to reduce the computation time, while improving the quality of the transition trajectory. Upon generation, the trajectories must be trackable by the *full-scale vehicle* using the GNC architecture described in Fig. 2.

Flight Dynamics Simulation Model: To validate the trajectory generation approach, a flight dynamics simulation model of the CRC-20 QRBP designed by DEVCOM Army Research Laboratory (Ref. 3) will be used. The flight dynamics simulation of the CRC-20 is a 6-DOF model consisting of 12 states and 4 control inputs. The state vector $\mathbf{x} = [x \ y \ z \ \phi \ \psi \ u \ v \ w \ p \ q \ r]^T$, is defined in the conventional aerodynamic sense (right handed inertial and body frames), with the exception that the nose of the vehicle is defined by y , while the left wing is defined by x , changing the corresponding definitions of ϕ (pitch angle), θ (roll angle), u (leftward body velocity), v (noseward velocity), p (pitch rate), and q (roll

rate). This is done to avoid gimbal lock during planning and control. The input vector $\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4]$ is the rotational speeds of each individual rotor. The model uses momentum theory and blade element analysis (BEA) to model rotor wake over the biplane wings and aerodynamics forces/moments acting on the vehicle.

SIMPLIFIED DYNAMICAL MODEL FOR TRAJECTORY PLANNING

To generate transition trajectories in the optimizer a reduced order (simplified) model must be used, as in Ref. 11. We briefly review this model below:

Simplified QRBP Dynamics for Trajectory Planning: For the purposes of trajectory planning for the transition problem, a simplified model was proposed in Ref. 11 for computational efficiency while capturing the key characteristics of the longitudinal dynamics of the vehicle. We briefly describe this model here. This model only takes into account the longitudinal motion of the QRBP (i.e. translation in the vertical $x-z$ plane of the inertial frame and rotation (pitch) about the body frame y -axis). 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) are assumed to be stabilized by the inner loop control.

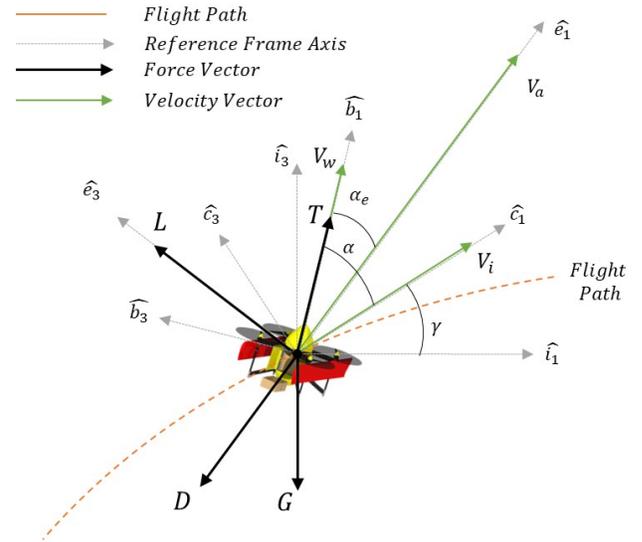


Figure 4. Reference Frame Construction of QRBP Model

Figure 4 shows the free body diagram of the simplified dynamics of the QRBP during the transition flight. Similar to the 3-DOF dynamics of the conventional fixed-wing airplane (Ref. 15), four state variables (i.e. 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. The rotor thrust T and geometric angle of attack α are employed to steer the vehicle. The simplified vehicle dynamics in the transition plane are:

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

where:

$$\begin{aligned}
\alpha &= \phi - \gamma \\
V_w &= 1.2 \sqrt{\frac{T}{8\rho\pi R}} \\
V_a &= \sqrt{V_i^2 + V_w^2 + 2V_i V_w \cos \alpha} \\
V_a \sin \alpha_e &= V_i \sin \alpha \\
L &= 0.5\rho(C_{L0} + C_{L\alpha} \alpha_e) S_l V_a^2 \\
D &= 0.5\rho C_{D0} S_d V_a^2
\end{aligned} \tag{2}$$

where L and D are the aerodynamic lift and drag forces respectively, 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 the rotor wake (α_e represents the angle between the vehicle longitudinal axis and the airspeed V_a , and thus determines magnitude and direction of L and D). Note that the longitudinal dynamics are represented such that the state $\mathbf{x} = [x, z, V_i, \gamma]^T$ and the input $\mathbf{u} = [T, \alpha]^T$. In this paper, the differential flatness remodeling is based on this original simplified model.

TRAJECTORY OPTIMIZATION: DIFFERENTIALLY FLAT FORMULATION

Differentially Flat Model Reformulation: While the above model was shown to generate reasonable transition trajectory by solving Problem 1, the computation times required to obtain these were of the order of 10s of seconds. To address this issue, we leverage the differential flatness property of the original nonlinear dynamics represented in Eq. 2, so that they can be transformed into an equivalent linear system, whose state and input variables are tied to the original ones by defining a set of endogenous mappings. We choose the inertial position of the vehicle as flat outputs ($y_1 = x$, $y_2 = z$) and express the state variables in terms of these flat outputs, such that:

$$\begin{aligned}
x &= y_1 & \dot{x} &= \dot{y}_1 \\
z &= y_2 & \dot{z} &= \dot{y}_2 \\
V &= \sqrt{\dot{y}_1^2 + \dot{y}_2^2} & \dot{V}_i &= \frac{\dot{y}_1 \dot{y}_1 + \dot{y}_2 \dot{y}_2}{\sqrt{\dot{y}_1^2 + \dot{y}_2^2}} \\
\gamma &= \tan^{-1}\left(\frac{\dot{y}_2}{\dot{y}_1}\right) & \dot{\gamma} &= \frac{\dot{y}_1 \dot{y}_2 - \dot{y}_2 \dot{y}_1}{\dot{y}_1^2 + \dot{y}_2^2}
\end{aligned} \tag{3}$$

This formulation is used to propagate the system using the following chains of integrators:

$$\begin{aligned}
\dot{\mathbf{q}} &= \mathbf{A}\mathbf{q} + \mathbf{B}\mathbf{v}, \text{ where } \mathbf{A} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{J}_2 \\ \mathbf{J}_3 \end{bmatrix}, \\
\mathbf{J}_1 &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{J}_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \mathbf{J}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{4}$$

where we define the virtual state $\mathbf{q} = [x \ \dot{x} \ z \ \dot{z}]^T$ and the synthetic input $\mathbf{v} = [\ddot{x} \ \ddot{z}]^T$. The initial and terminal boundary constraints change accordingly ($\mathbf{q}(t_0) = \mathbf{q}_0$, $\mathbf{q}(t_f) = \mathbf{q}_f$), such that we maintain the desired state before and after transition. We preserve the original nonlinear dynamics of the system by enforcing the expression (derived from Eq. 1) below:

$$\begin{aligned}
T \cos(\alpha) - L \sin(\alpha - \alpha_e) - D \cos(\alpha - \alpha_e) - ma_{\top} &= 0 \\
T \sin(\alpha) + L \cos(\alpha - \alpha_e) - D \sin(\alpha - \alpha_e) - ma_{\perp} &= 0
\end{aligned} \tag{5}$$

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

Using the differentially flat dynamics model, we reformulate the optimization problem to the form represented by Eq. 6 with the cost function representing time optimization, ($J = t_f - t_0$), the dynamic constraints represented by the chain of integrators described in Eq. 4, and the decision variables being the virtual state and synthetic input (\mathbf{q}, \mathbf{v}). The coupling of the plant inputs ($\mathbf{u} = [T, \alpha]^T$) with the differentially flat outputs \mathbf{q} (and their derivatives) and the synthetic input \mathbf{v} is enforced using the implicit relationships for T and α , such that $g_x(\mathbf{q}, \mathbf{v}) \in \mathcal{X}$, and $g_u(\mathbf{q}, \mathbf{v}) \in \mathcal{U}$. Finally, we add constraints that limit the vehicle's thrust output ($T \leq T_{MAX}$) to coincide with the maximum power output of the vehicle ($P_{MAX} = 1500W$). We also constrain the effective angle attack such that it remains within a specified tolerance of the stall angle ($\alpha_e \leq \alpha_{STALL} - \epsilon_{TOLL}$). With the establishment of these constraints and cost function, we re-formulate the optimization problem as follows:

$$\begin{aligned}
\arg \min_{x, u} \quad & J = \int_{t_0}^{t_f} dt, \quad \text{cost function} \\
s.t. \quad & \dot{\mathbf{q}} = \mathbf{A}\mathbf{q} + \mathbf{B}\mathbf{u}, \quad \text{dynamic constraints} \\
& \mathbf{q}(t_0) = \mathbf{q}_0, \quad \text{initial state constraint} \\
& \mathbf{q}(t_f) = \mathbf{q}_f, \quad \text{terminal state constraint} \\
& g_x(\mathbf{q}, \mathbf{v}) \in \mathbf{X} \quad g_u(\mathbf{q}, \mathbf{v}) \in \mathbf{U}, \quad \text{state and input constraints}
\end{aligned} \tag{6}$$

This differentially flat optimal trajectory planner can be solved using a standard NLP solver (such as IPOPT) to generate trajectories for transition.

CONTROLLER DESIGN

For trajectory tracking, we use the cascaded nonlinear dynamic inversion control architecture described in Ref. 5. For addressing the two flight regimes (hover/ascent and fixed wing flight), two controllers are designed for QRBP that run in parallel: the first approximates the vehicle dynamics as a pure quadrotor, while the latter approximates the vehicle as a pure fixed-wing aircraft. The dynamics of each case are derived with the appropriate simplifying assumptions (i.e., neglect aerodynamic forces for the quadrotor mode controller, include them for the fixed-wing mode controller). The strategy is to then switch between the two controllers based on the pitch angle of the vehicle during transition. Figure 5 shows a diagram

of the control architecture for control of the CRC-20. We modify the controller in Ref. 5 by using a controller blender, where the control inputs of each separate controller is gradually blended in transition. We briefly describe the control blender in the next section.

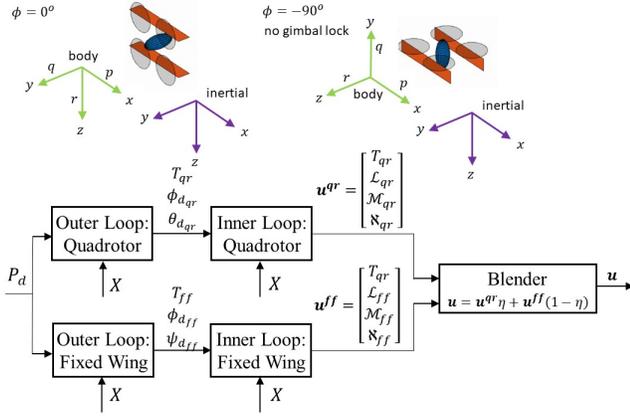


Figure 5. Blended Control Architecture.

Blending Strategy: As shown in Fig. 5, the control input \mathbf{u} is obtained from a control blender driven by the flight path angle γ of the following form:

$$\mathbf{u} = \eta(\gamma)\mathbf{u}^{qr} + (1 - \eta(\gamma))\mathbf{u}^{ff} \quad (7)$$

where \mathbf{u}^{qr} , and \mathbf{u}^{ff} are the outputs of the quadrotor controller and fixed wing controller, respectively, and η is the weighting determined by the flight path angle γ . We propose η as a linear function of γ as shown below:

$$\eta = \begin{cases} \frac{\gamma - \gamma_{\min}}{\gamma_{\max} - \gamma_{\min}} & \text{if } \gamma \in [\gamma_{\min}, \gamma_{\max}] \\ 0 & \text{if } \gamma > \gamma_{\max} \\ 1 & \text{if } \gamma < \gamma_{\min} \end{cases} \quad (8)$$

where γ_{\min} and γ_{\max} are determined to be the start and end of the blending regime. Thus, we enforce the integrity of the blender such that the quadrotor controller is in full control if $\gamma < \gamma_{\min}$, the fixed wing controller is in full control if $\gamma > \gamma_{\max}$, and the control input is blended if $\gamma_{\min} \leq \gamma \leq \gamma_{\max}$.

Note: We assume there is perfect allocation of the control forces and moments generated by the controller to the concatenated input $\bar{\mathbf{u}}$ to the CRC-20 described in the Problem Statement.

RESULTS

This section describes the demonstration and an analysis of the differentially flat trajectory planning and tracking algorithm for transition. First, we present sample trajectories for two flight cases: hover to forward flight ($H \rightarrow FF$), and vice versa ($FF \rightarrow H$). Next, a performance analysis of the trajectory planner is presented, detailing both a spatial feasibility

Constraint	Limits
Power \mathbf{P}	$\in [0, 1500]W$
Effective angle of attack α_e	$\pm 10^\circ$
Terminal position x_f, z_f	unconstrained
Maximum flight path angle rate $\dot{\gamma}$	$\pm \{30, 40, 60\} \frac{deg}{s}$
Forward flight velocity V_i	$\{25, 30, 35\}$ kts

Table 1. Key Path and Boundary Constraints

analysis (an analysis of feasibility based on changing boundary conditions in x and z) of the trajectory planner, and an analysis of average computational time for both flight cases. Finally, tracking performance for representative flight cases for hover to forward flight ($H \rightarrow FF$) is presented.

Trajectory Planner Sample Profiles: In this section, we present inertial-position time trajectories (t vs. $x(t)$, t vs. $z(t)$) and the corresponding flight paths ($x(t)$ vs. $z(t)$) generated by the differentially flat trajectory planner for the $H \rightarrow FF$ and the $FF \rightarrow H$ transition cases. We command the optimizer to generate time-optimal transition trajectories for these flight cases under various key path constraints to see how the inertial position profiles change. We tabulate the constraints of interest in Table 1.

Hover to Forward Flight: For this analysis, we evaluate the effect of the constraint on rate of change of flight path angle $\dot{\gamma}$, while keeping all other key constraints (α_e , P , etc.) constant between flight cases. Figure 6 shows various cases of the $H \rightarrow FF$ flight mission as the $\dot{\gamma}$ constraint varies.

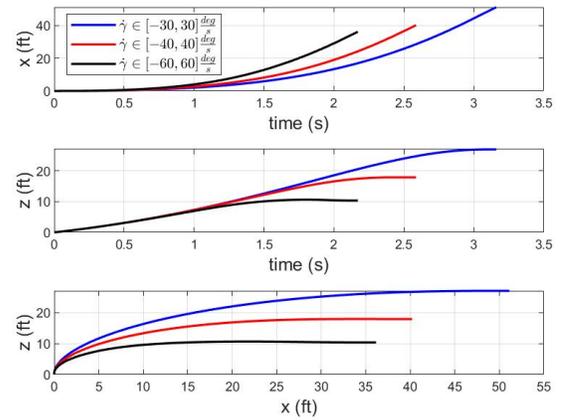


Figure 6. Differentially Flat Time Optimal Trajectories for Hover to Forward Flight Case ($\dot{\gamma}$ constraint varied, all other constraints constant)

From Fig. 6, we note that as the bound on $\dot{\gamma}$ is increased (from $\pm 30 \frac{deg}{s}$ to $\pm 60 \frac{deg}{s}$), the vehicle requires both less time and less space in x and z to complete the transition maneuver. While this is consistent with intuition, we verify the feasibility implied by these results in the *Feasibility Analysis* and *Tracking Performance* sections.

Forward Flight to Hover: For analysis of the trajectories for $FF \rightarrow H$, we evaluate the effect of two constraints: varying

initial forward flight velocity V_i (leaving terminal position unconstrained) and the terminal altitude z_f of the transition (with $V_i = 25$ kts). Figures 7 and 8 show the inertial position profiles of various $FF \rightarrow H$ flight cases as we vary the boundary constraint on V_i and z_f , respectively.

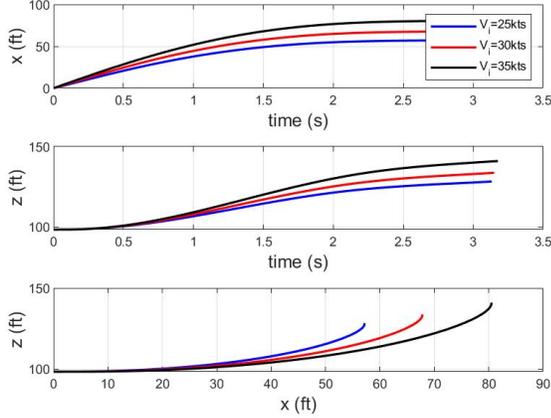


Figure 7. Differentially Flat Time Optimal Trajectories for Forward Flight to Hover Case (initial V_i varied, all other constraints constant)

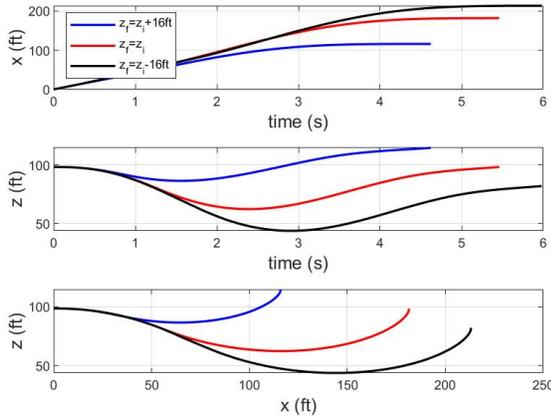


Figure 8. Differentially Flat Time Optimal Trajectories for Forward Flight to Hover Case (terminal altitude z_f constraint varied, all other constraints constant)

From Fig. 7, we observe that increasing the initial forward flight velocity (with no final position constraint) does not significantly affect the time of flight (likely due to keeping the constraint on $\dot{\gamma}$ constant at $\pm 30 \frac{deg}{s}$). However, there is an increased demand on the lateral distance and altitude required to complete the transition. Figure 8 illustrates that the opposite case (i.e., keeping V_i constant at 25 kts while constraining the terminal altitude) has a similar effect. However, in the case where z_f is constrained, the vehicle requires significantly more longitudinal draw distance in x to complete the transition, with more needed as the z_f constraint varies from

$$z_f = z_i + 5m \text{ to } z_f = z_i - 5m.$$

Trajectory Planner Studies

In this section, we detail two studies on the differentially flat trajectory planner. First, the trajectory planning algorithm is used to determine the capability of the transitioning UAS for maneuvers, for example in a tight space. As a representative study, we use the planner to determine the x - z space for which feasible trajectories can be generated given different maximum flight path angle rates. Next, for demonstrating real-time application of the trajectory planner, we perform a study of typical computational times required to solve the trajectory planning problem.

Determining Feasibility of Transitions in Tight Spaces:

The trajectory planner can be used to determine the *minimum region in the x - z plane* within which the UAS can perform a transition maneuver. To obtain this, we change the constraints on x and z : $x_{\min} \leq x \leq x_{\max}$ and $z_{\min} \leq z \leq z_{\max}$ in the differentially flat optimization problem (Eq. 6) and ascertain whether the solver can generate a feasible trajectory. We thus determine the values of x_{\max} and z_{\max} for which the trajectory planner generates feasible trajectories (keeping x_{\min} and z_{\min} fixed). This analysis was conducted for the $H \rightarrow FF$ and $FF \rightarrow H$ cases with varying constraints as described above (varying maximum $\dot{\gamma}$ for the $H \rightarrow FF$ case,; varying initial velocity V_i and terminal height z_f for the $FF \rightarrow H$ case, keeping all other constraints the same).

Figure 9 shows the feasibility contours for the $H \rightarrow FF$ case for different maximum allowable flight path angle rates, $\dot{\gamma}$, while Fig. 10 shows the same for the $FF \rightarrow H$ case for different initial velocity V_i . For each instance, the initial position of the vehicle (x_i, z_i) is located at the origin $(0,0)$. The lower limit on x and z was also kept consistent between each instance ($x_{\min} = z_{\min} = -30m$). The optimization problem was solved for a range of x_{\max} and z_{\max} , with an upper limit of $40m$ for $H \rightarrow FF$ and an upper limit of $70m$ for $FF \rightarrow H$ (with a resolution of $1m$ for both).

From Fig. 9, for the $H \rightarrow FF$ case we observe that the feasibility region increases (both w.r.t. x_{\max} and z_{\max}) with larger allowable $\dot{\gamma}$. Further, based on the trend of the set boundary, the limit of each profile approaches a positive, non-zero value. This indicates that the constraint on $\dot{\gamma}$ results in a corresponding limit on the minimum allowable space within which the $H \rightarrow FF$ transition maneuver can be performed (i.e., $x_{\max} \approx 15m, z_{\max} \approx 5m$ for $|\dot{\gamma}| \leq 30 \text{ deg/s}$).

Based on Fig. 10, for the $FF \rightarrow H$ case we note that the feasibility set shrinks as we increase initial V_i . We also notice a similar result regarding the limits on x_{\max} and z_{\max} as in the $H \rightarrow FF$ feasibility contours. Of particular note, however, is the region of each profile where there is a linear change in z_{\max} as x_{\max} increases, converging on a z_{\max} limit of zero. Recall that the minimum limit on altitude $z_{\min} = -30m$, while the vehicle begins the maneuver at the origin $(0,0)$. Thus, as z_{\max} approaches zero, the trajectory planner is commanding the vehicle to take advantage of the clearance below the

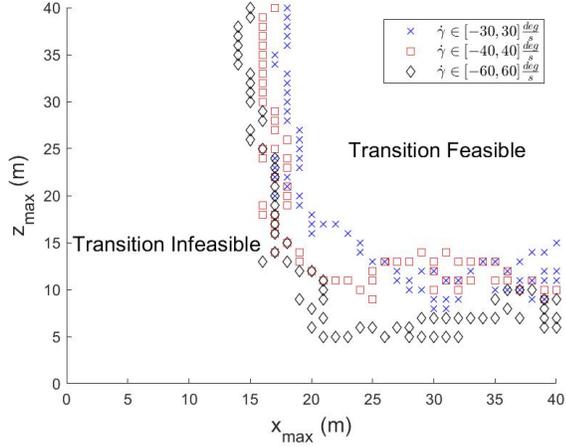


Figure 9. Hover to Forward Flight Feasibility Contours for different bounds on $\dot{\gamma}$

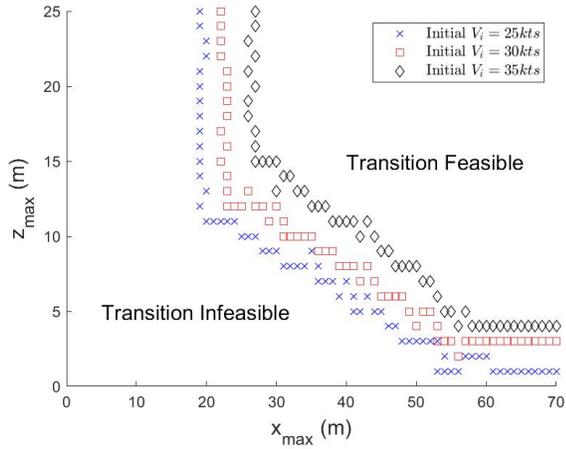


Figure 10. Feasibility Contours (in x-z plane) for Hover to Forward Flight for different initial velocities V_i

starting point to complete the $FF \rightarrow H$ transition, despite the fact that the terminal position of the vehicle remains unconstrained.

Computational Time Analysis: To evaluate the feasibility of using the planner in real-time, a study of the typical computational times required to solve the trajectory optimization problem was performed. Figure 11 shows the histogram detailing the statistical distribution of computational times needed to solve the trajectory optimization problem, for trajectories generated for the $H \rightarrow FF$ (varying $\dot{\gamma}$ constraint) and $FF \rightarrow H$ (varying initial V_i) cases generated for the feasibility analysis study in the previous section. The distribution shown by Figure 11 is indicative of the consistency of the differentially flat trajectory planner’s computational performance. The typical computational time for the $H \rightarrow FF$ trajectories require around 0.25s to solve, while that for the $FF \rightarrow H$ case is around 0.3s. Note that these computational times correspond to the base level performance of an *Intel Core i7* 7th generation 2.8 GHz processor.

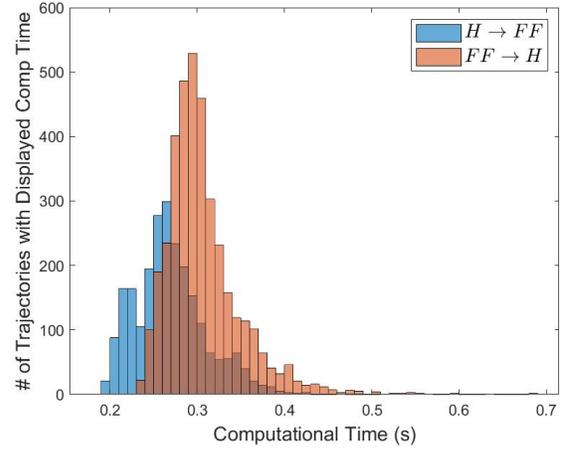


Figure 11. Histogram of Computational Time for Feasible $H \rightarrow FF$ and $FF \rightarrow H$ Trajectories

Simulation Results

Hover to Forward Flight: We implement the control architecture described in Ref. 5 (with blending operation described in the *Controller Design* section) to evaluate the trackability of the trajectories generated for the $H \rightarrow FF$ case. Note that for initialization of the control blender, $\gamma_{\min} = -8^\circ$, and $\gamma_{\max} = -5^\circ$. Figures 12 and 13 show the outer loop tracking performance of the time profiles for inertial position and the flight path, respectively, for the $H \rightarrow FF$ case where $\dot{\gamma}$ constrained to $\pm 30 \frac{\text{deg}}{\text{s}}$. Figure 14 shows the corresponding tracking performance for inertial velocity V_i and flight path angle γ , as well as the inner loop tracking of the pitch angle ϕ . Similarly, Figs. 15, 16 and 17 show the same for the $H \rightarrow FF$ case where the constraint on $\dot{\gamma}$ is relaxed to $\pm 60 \frac{\text{deg}}{\text{s}}$.

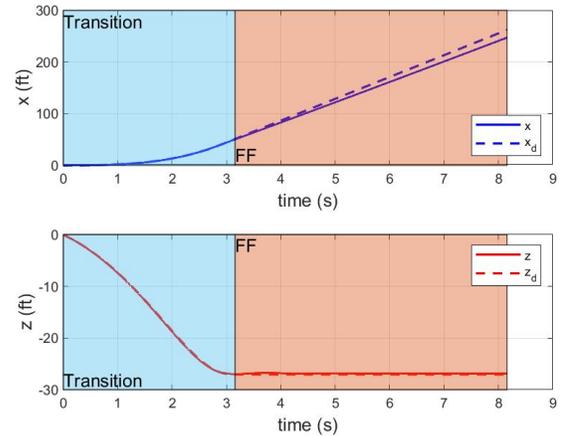


Figure 12. Hover to Forward Flight Outer Loop Tracking Results: $\dot{\gamma} \in [-30, 30] \frac{\text{deg}}{\text{s}}$

As illustrated in Figs. 12, 13 and 14, the controller demonstrates adequate tracking quality for inertial x and z position during the transition, however there is increasing error in inertial x tracking during forward flight due to the steady state

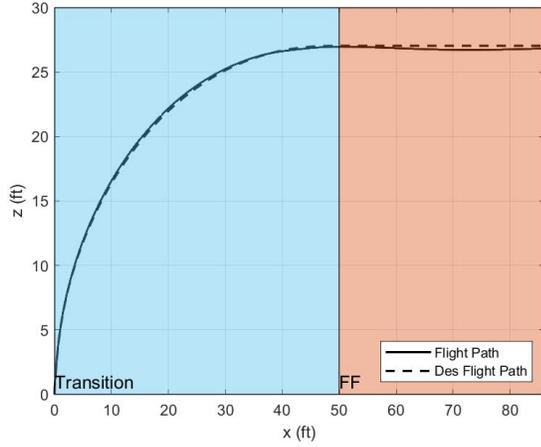


Figure 13. Hover to Forward Flight Flight Path Tracking
Results: $\dot{\gamma} \in [-30, 30] \frac{deg}{s}$

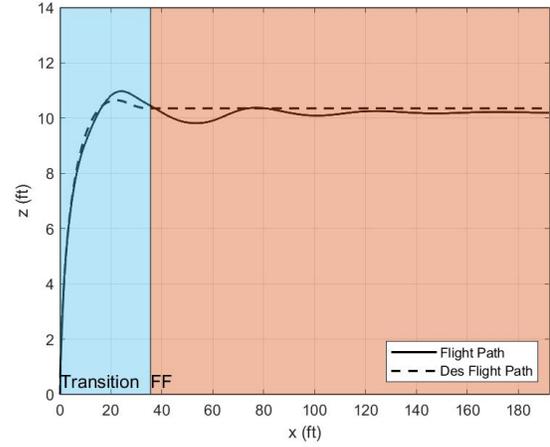


Figure 16. Hover to Forward Flight Flight Path Tracking
Results: $\dot{\gamma} \in [-60, 60] \frac{deg}{s}$

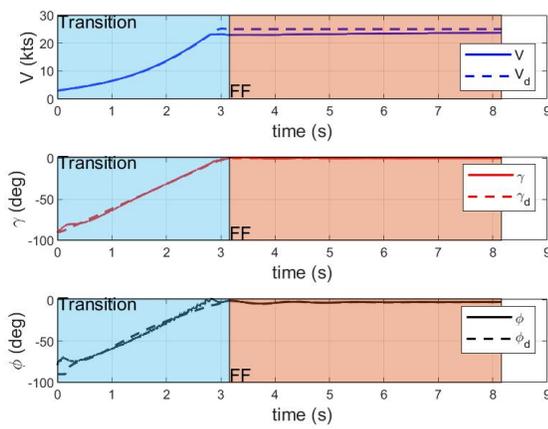


Figure 14. Hover to Forward Flight Inner Loop Tracking
Results: $\dot{\gamma} \in [-30, 30] \frac{deg}{s}$

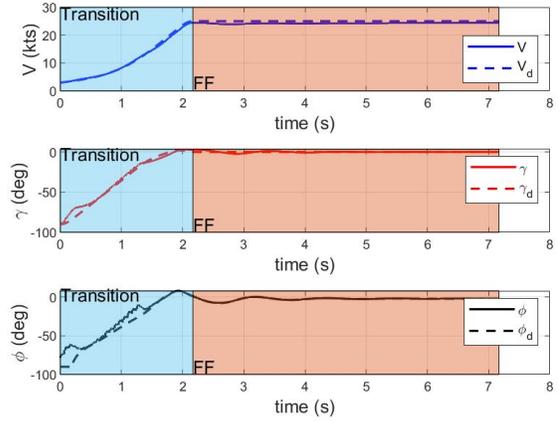


Figure 17. Hover to Forward Flight Inner Loop Tracking
Results: $\dot{\gamma} \in [-60, 60] \frac{deg}{s}$

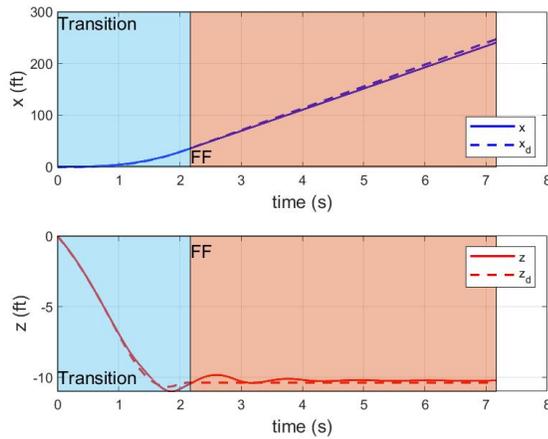


Figure 15. Hover to Forward Flight Outer Loop Tracking
Results: $\dot{\gamma} \in [-60, 60] \frac{deg}{s}$

error in V_i (the fact that no steady state error is present in γ or ϕ tracking in forward flight verifies that the error exists

specifically in the \dot{x} component of V_i as opposed to the \dot{z} component). Note that this error coincides with the transition from the quadrotor controller to the fixed-wing controller, as the error in \dot{y} begins to occur the moment $\gamma \approx -8^\circ$. This indicates the forward flight controller as a potential source of the error. This remains an open issue that will be addressed in future work.

Note the $H \rightarrow FF$ tracking case where the constraint on $\dot{\gamma} = \pm 60 \frac{deg}{s}$, shown in Figs. 15, 16 and 17. While the vehicle still successfully transitions from hover and stabilizes in forward flight (indicated by successfully outer loop position tracking), the flight path angle profile γ strays slightly from the desired γ during transition (specifically when the quadrotor controller has full control authority over the vehicle). This suggests the quadrotor controller is less effective for more aggressive transitions.

CONCLUSIONS

This paper studied an optimization-based trajectory planning approach for autonomous transition of a quadrotor biplane

tailsitter (QRBP) between the flight modes of hover to forward flight and vice versa. We established and reported the following findings: (1) The proposed differential-flatness based trajectory planner can generate transition trajectories for the QRBP (for both the $H \rightarrow FF$ and $FF \rightarrow H$ cases) with computational solution times that are suitable for on-board path planning; (2) The path planner can be used to determine the agility of the transitioning UAS (e.g., the space needed to perform a specific maneuver or transition), given specific vehicle constraints such as power; (3) the optimal transition trajectories (for the $H \rightarrow FF$) can be tracked using a cascaded dynamic inversion approach with control blending.

Based on this initial study, there are several open questions for future investigation; including validating the trajectory tracking performance for the $FF \rightarrow H$ case and more complex maneuvers in general, the design of a robust tracking control allocation methodology and finally the demonstration of *real-time* trajectory generation and 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 (VLRCE) Program, grant number W911W61120012, with Dr. Mahendra Bhagwat.

REFERENCES

- 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.
- 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.
- 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, Philadelphia, PA, May 2019.
- 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, Scottsdale, AZ, January 2013.
- 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, Kissimmee, FL, January 2018.
- 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.
- Deyuan, L., Hao, L., and Lewis, F. L., "Robust trajectory tracking control for unmanned tail-sitters in aggressive flight mode transitions," *International Journal of Robust Nonlinear Control*, Vol. 29, December 2018, pp. 4270–4284. DOI: 10.1002/nrc.4626
- 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.
- 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, Chandler, AZ, January 2015.
- Todeschini, D., Fagiano, L., Micheli, C., and Cattano, A., "Control of vertical take off, dynamic flight and landing of hybrid drones for airborne wind energy systems," American Control Conference (ACC), Philadelphia, PA, July 2019.
- 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.
- Chamseddine, A., Rabbath, C. A., and Join, C., "Flatness-Based Fault Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, October 2012, pp. 2832–2850.
- Chamseddine, A., Li, T., Zhang, Y., Rabbath, C. A., and Theilliol, D., "Flatness-based Trajectory Planning for a Quadrotor Unmanned Aerial Vehicle Test-bed Considering Actuator and System Constraints," 2012 American Control Conference (ACC), Montreal, Canada, June 2012.
- Zhao, D., Krishnamurthi, J., Mishra, S., and Gandhi, F., "A Trajectory Generation Method for Time-Optimal Helicopter Shipboard Landing," American Helicopter Society 74th Annual Forum Proceedings, Fort Worth, TX, May 2018.
- Weitz, L. A., "Derivation for a Point-Mass Aircraft Model Used for Fast-Time Simulation," MITRE Technical Report, November 1995.