2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Differential flatness based trajectory generation for time-optimal helicopter shipboard landing

Di Zhao[1] Jayanth Krishnamurthi[1] Farhan Gandhi[1] and Sandipan Mishra[1]

Abstract— **In this paper, a time-optimal trajectory generation algorithm is proposed for helicopter shipboard landing. The algorithm utilizes a simplified model of the helicopter's dynamics and exploits the differential flatness of the model to formulate a nonlinear programming problem, whose solution provides time-optimal reference approach/landing trajectories. The trajectories are then tracked by an inner-loop linear dynamic inversion (LDI) controller to generate the actual inputs that steer the full-state nonlinear helicopter model. The proposed algorithm reduces approach/landing flight time and enables a higher degree of maneuverability (such as obstacle avoidance and large heading angle change) in comparison to typical state of the art methods of trajectory generation. Because of its computational efficiency, the path planner can also be used in real-time, i.e., through iterative recalculation of the remaining trajectory to account for deviations from the planned flight path. High fidelity simulations have been conducted on a verified UH-60A Black Hawk model, which show the effectiveness of the proposed method.**

I. Introduction

Shipboard landing is among the most challenging helicopter flight operations, because of (1) the limited landing time, (2) the stringent safety constraints, (3) the rough sea conditions and turbulent shipboard motion , and (4) the complex ship-airwake-helicopter interactions during the landing maneuver. Hence, there is significant pilot workload during the shipboard landing operation, which makes computer-assisted and/or fully autonomous control strategies imperative.

There has been significant interest in designing effective control solutions for autonomous landing. In [1], a novel control strategy was developed based on optical flow theory. A parameter *tau* inspired from human pilot behavior was defined, which leads to a 'natural' and smooth autonomous landing. In [2], a vision-based control system combining image-based visual servoing (IBVS) controller and a translational rate command (TRC) controller was implemented to generate control inputs based on the translational motion information obtained by the camera, which enables the helicopter to track and land onto the deck. In [3], a model predictive control (MPC) algorithm was proposed to deal with ship airwake and rough sea conditions during the landing process. Nonlinear simulations proved the performance of the controller. In addition, the application of the dynamic inversion (DI) controller for helicopter landing was documented in [4], [5]. These controllers eliminated the need for gain scheduling and enforcing the controlled variables to follow commanded responses [6], [7]. Human piloted and autopiloted nonlinear simulations were conducted in [4] and [5] respectively, proving the effectiveness of the DI controller.

In addition to trajectory tracking control laws for helicopter shipboard landing problem, there has also been progress in creating time-optimal trajectory generation strategies for quadrotors in the past few years [8]–[11]. Specially, a new trajectory generation algorithm exploiting the differential flatness of the quadrotor dynamics was introduced in [11]. Such methods proved to be computationally efficient in solving for time-optimal reference trajectories that guide the quadrotor to the moving platform and lend themselves to iterative real-time solution (in a model predictive shrinking horizon framework, for example), which makes the class of differential flatness based optimization algorithms a potential candidate for trajectory planning for helicopter maneuvers in general, and landing in particular.

In contrast to the large body of research on trajectory generation for quadrotors, current trajectory generation methods for helicopter flight typically parameterize the reference trajectory with geometric or kinematic variables. In general, these trajectories do not guarantee time-optimality and lack lateral maneuverability, leading to the conservative design of flight missions requiring responsiveness, obstacle avoidance and large heading maneuvers.

Motivated by the need for trajectory planning methods for the helicopter shipboard landing problem, this paper presents a new time-optimal trajectory generation algorithm compatible with the standard cascade control structure [5] for helicopters. The helicopter outer-loop dynamics are converted into an equivalent (and computationally efficient) model by exploiting its differential flatness. This model is then used in the formulation of the trajectory optimization problem, which can be solved for a time-optimal reference trajectory that is also capable of necessary obstacle avoidance and lateral maneuver. In addition, the corresponding (virtual) feedforward control input to be fed into the inner-loop controller is also generated by the proposed algorithm. A standard linear dynamic inversion (LDI) controller is used for inner-loop tracking control, which produces the actual control inputs to the helicopter based on the command information from the outer-loop. Simulations conducted on the full-state nonlinear helicopter model validate the feasibility and effectiveness of the proposed algorithm.

[1] Di Zhao, Jayanth Krishnamurthi, Farhan Gandhi and Sandipan Mishra are with the department of Mechanical, Aerospace and Nuclear Engineering of Rensselaer Polytechnic Institute, 110 8th street, Troy, NY, 12180 {zhaod3,krishj,gandhf,mishs2}@rpi.edu

978-1-5386-5428-6/$31.00 ©2018 AACC

II. HELICOPTER SHIPBOARD LANDING

A. Problem Statement

A schematic illustrating the helicopter shipboard landing problem is shown in Fig.1. In order to succinctly describe the helicopter shipboard landing problem, the helicopter/shipboard system is separated into four subsystems including helicopter dynamic model, shipboard kinematic model, inner-loop linear dynamic inversion (LDI) controller and outer-loop path planner, organized by the cascade control architecture shown in Fig.2.

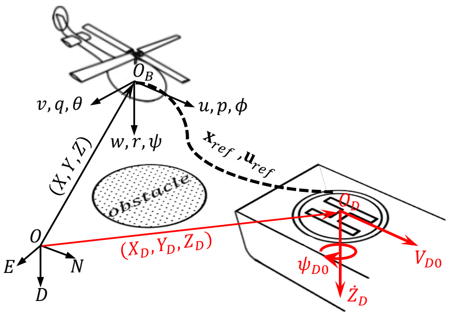

Fig. 1: Schematic of helicopter shipboard landing problem. Based on the measurement of both the fuselage and the shipboard, the outer-loop path planner solves and stores the reference time-optimal trajectory and the corresponding virtual input, while the inner-loop controller generates the actual input that tracks the reference trajectory by using the information from the outer-loop. The helicopter is then steered to circumvent the obstacle, approach and land on the shipboard safely with minimum impact.

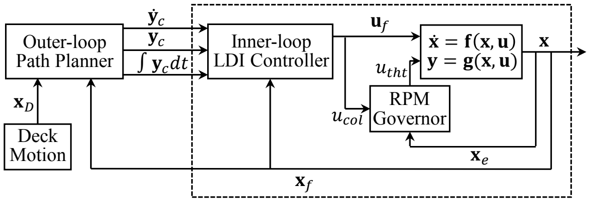

Fig. 2: Overall control architecture for helicopter shipboard landing. The outer-loop path planner triggers the optimization algorithm when is necessary to generate a reference trajectory based on current deck state $\mathbf{x}_D$ and fuselage state $\mathbf{x}_f$. Then, the above trajectory is stored so that the real time commands $\dot{\mathbf{y}}_c$, $\mathbf{y}_c$ and $\int \mathbf{y}_c dt$ can be scheduled from it and be provided to the inner-loop controller, which applies a LDI control law to track the above commands. Subsequently, the fuselage input $\mathbf{u}_f$ and the throttle input u_{tht} are generated by the inner-loop controller and RPM governor respectively, which steer the helicopter to the shipboard.

Next, we present a brief description of the helicopter dynamics, the shipboard kinematics and the inner-loop controller. Then, a simplified outer-loop dynamic model is presented and its differential flatness verified.

B. Helicopter dynamic model

We will use a UH-60A Black Hawk model developed by [12], which is a derivative of Sikorsky's GenHel model [13]. The dynamics, in general, are given by:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x},\mathbf{u}) \end{aligned} \tag{1}$$

where, $\mathbf{y}$ is the measurement of the states used in the controller, $\mathbf{x}$ is the state vector :

$$\mathbf{x} = \left[\mathbf{x}_f^T, \mathbf{x}_r^T, \mathbf{x}_t^T, \mathbf{x}_e^T\right]^T \tag{2}$$

$$\begin{aligned} \text{where} \quad \mathbf{x}_f &= \left[u,v,w,p,q,r,\phi,\theta,\psi,X,Y,Z\right]^T \\ \mathbf{x}_r &= \left[\beta_0,\beta_{1s},\beta_{1c},\beta_d,\dot{\beta}_0,\dot{\beta}_{1s},\dot{\beta}_{1c},\dot{\beta}_d,\lambda_0,\lambda_{1s},\lambda_{1c}\right]^T \\ \mathbf{x}_t &= \lambda_{0TR}, \quad \mathbf{x}_e = \left[\Omega,\chi_f,Q_e\right]^T \end{aligned} \tag{3}$$

$\mathbf{x}_f$ denotes 12 fuselage rigid body states, $\mathbf{x}_r$ denotes 8 blade flapping states and 3 inflow states of the main rotor, $\mathbf{x}_t$ denotes the tail rotor inflow state, $\mathbf{x}_e$ denotes 3 engine states.

The control input $\mathbf{u}$ is given by:

$$\mathbf{u} = \left[u_{lat}, u_{long}, u_{col}, u_{ped}, u_{tht}\right]^T, \tag{4}$$

which consists of lateral, longitudinal, collective joystick input to the main rotor, pedal input to the tail rotor, and throttle input to the engine. Note that fuselage input $\mathbf{u}_f = \left[u_{lat}, u_{long}, u_{col}, u_{ped}\right]^T$ is comprised of the input channels governing the fuselage motion.

C. Shipboard (deck) kinematic model

There have been several models proposed for capturing shipboard (deck) motion, such as [3], [5]. We now present a simplified shipboard kinematic model that is assumed to be known to the outer-loop path planner.

The horizontal motion of the deck is defined by a constant velocity V_{D0} ($20 knots$ or $10.289 m/s$) with a constant heading angle ψ_{D0}, with the heaving of the shipboard approximated by a sinusoidal function of time. while this is an example of a typical scenario, the approach developed in this paper can be extended in a straightforward manner to any *explicit* time-varying deck motion as long as the planner has a suitable forecast. Starting at (X_{D0}, Y_{D0}), the North-East-Down coordinate and the heading angle of the shipboard is given by:

$$\begin{aligned} X_D(t) &= X_{D0} + V_{D0} \cdot t \cos \psi_{D0} \\ Y_D(t) &= Y_{D0} + V_{D0} \cdot t \sin \psi_{D0} \\ Z_D(t) &= Z_{D0} + A_D \sin(2\pi t \big/ T_D + \phi_D) \\ \psi_D(t) &= \psi_{D0} \end{aligned} \tag{5}$$

where t is the simulation time, $Z_{D0} = -6m$ is the mean downward coordinate of the deck and A_D, T_D and ϕ_D denote amplitude, period and phase of the deck oscillation.

Remark: While the scopte of this paper is limited to simplistic ship deck motion, a more realistic shipboard motion model based on the Systematic Characterization of the Naval Environment (SCONE) database [14], together with the corresponding ship-airwake-helicopter interaction model can be used to verify the robustness of the proposed algorithm to noise and disturbances.

D. Inner-loop LDI controller

The design of the inner-loop controller refers to a previous work [12] and is based on the well-known model-following LDI control law.

Fig.3 shows the structure of the controller, where the state space matrices $\mathbf{A}_f \in \mathbb{R}^{12\times12}$, $\mathbf{B}_f \in \mathbb{R}^{12\times4}$ are obtained by the linearization and order-reduction of the nonlinear model at different trim speeds. $\delta\mathbf{x}_f$ and $\delta\mathbf{u}_f$ are small perturbation from $\mathbf{x}_{f,trim}$ and $\mathbf{u}_{f,trim}$ respectively. Moreover, the commanded output vector here is chosen as $\mathbf{y}_c = \left[\dot{\phi}_c, \dot{\theta}_c, -\dot{Z}_c, \dot{\psi}_c\right]^T$, which denotes the rates of roll angle and pitch angle, the vertical velocity and the rate of yaw angle respectively.

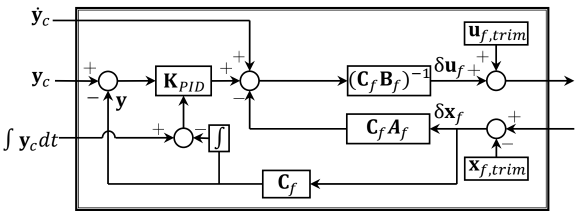

Fig. 3: Inner-loop LDI controller structure. The commands $\mathbf{y}_c$ and $\int \mathbf{y}_c dt$ are used for error compensation, while $\dot{\mathbf{y}}_c$ provides the feed-forward information. They together constitute the virtual input and then go through the dynamic inversion process, which generates the actual input perturbation $\delta\mathbf{u}_f$. The final input to the helicopter is thus $\mathbf{u}_f = \mathbf{u}_{f,trim} + \delta\mathbf{u}_f$. With the properly selected feedback gain matrix $\mathbf{K}_{PID}$, the actual fuselage output $\mathbf{y}$ and $\int \mathbf{y}dt$ can be enforced to converge to $\mathbf{y}_c$ and $\int \mathbf{y}_c dt$.

E. General framework for the outer-loop planner

The outer-loop path planner is organized by the structure shown in Fig.4.

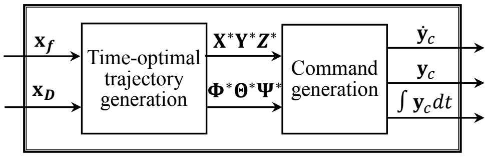
$\mathbf{x}_f$
$\mathbf{x}_D$
Time-optimal trajectory generation
$\mathbf{X}^*\mathbf{Y}^*\mathbf{Z}^*$
$\mathbf{\Phi}^*\mathbf{\Theta}^*\mathbf{\Psi}^*$
Command generation
$\dot{\mathbf{y}}_c$
$\mathbf{y}_c$
$\int \mathbf{y}_c dt$

Fig. 4: Outer-loop path planner structure. Upon activation, the fuselage state $\mathbf{x}_f$ and the shipboard state $\mathbf{x}_D$ are fed to the outer-loop path planner. The time-optimal trajectory generation algorithm then calculates for a feasible reference trajectory containing the reference translational motion ($\mathbf{X}^*, \mathbf{Y}^*, \mathbf{Z}^*$) and rotational motion (Φ^*, Θ^*, Ψ^*) (including up to second order time derivative informations). Then, real time virtual input $\dot{\mathbf{y}}_c$ and corresponding reference $\mathbf{y}_c$, $\int \mathbf{y}_c dt$ for the inner-loop can be scheduled respect to the current flight time from the reference trajectory, until a new one is updated.

F. Simplified dynamic model for the outer-loop

With the inner-loop controller being responsive and robust, the subsystems in the dashed line box in Fig.(2) can actually be incorporated into one simplified dynamic model for the trajectory planner. These simplified dynamics can be represented by:

$$\dot{\mathbf{x}}_{ol} = \mathbf{f}_{ol}(\mathbf{x}_{ol}, \mathbf{u}_{ol}) \tag{6}$$

where $\mathbf{x}_{ol} = \left[\phi, \theta, \psi, X, Y, Z, \dot{\phi}, \dot{\theta}, \dot{\psi}, \dot{X}, \dot{Y}, \dot{Z}\right]^T$ is essentially the fuselage state $\mathbf{x}_f$ with all elements transformed into North-East-Down (NED) coordinate frame. On the other hand, the control input $\mathbf{u}_{ol}$ is chosen to be identical to the virtual input $\dot{\mathbf{y}}$ of the inner-loop LDI controller. Hence:

$$\ddot{\phi} = u_1, \;\; \ddot{\theta} = u_2, \;\; \ddot{Z} = -u_3, \;\; \ddot{\psi} = u_4 \tag{7}$$

Note that Eq.(7) directly provides the dynamic model of the helicopter attitude and vertical motion, leaving the dynamics of the horizontal motion to be determined next.

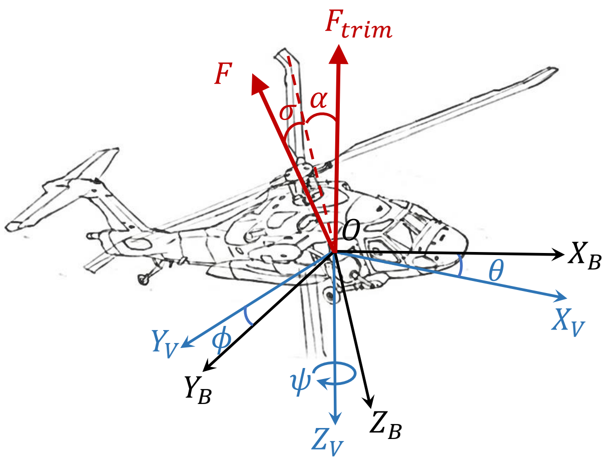

Fig. 5: Schematic of the helicopter coordinate systems. $OX_VY_VZ_V$ is the inertial frame with OX_V being the aircraft longitudinal axis's projection on the horizontal plane and OZ_V aligning with gravitational acceleration. $OX_BY_BZ_B$ is the aircraft body fixed frame. Notice that $OX_VY_VZ_V$ is transformed from North-East-Down(NED) inertial frame $OX_NY_EZ_D$ by $T_3(\psi)$ and $OX_BY_BZ_B$ is transformed from $OX_VY_VZ_V$ by $T_1(\phi)T_2(\theta)$.

As shown in Fig.5, the non-gravitational resultant force F_{trim} at trim attitude $(\phi_{trim}, \theta_{trim})$ points upward vertically and has the same magnitude of the gravitational force. With the control input perturbed from the trim condition, the helicopter reaches a new attitude (ϕ, θ), which leads to the rotation of the resultant force from F_{trim} to F. The tilting of F can be described by the rotation $T_1(\sigma)T_2(\alpha)T_3(\psi)$ of the NED frame $OX_NY_EZ_D$. Therefore, the dynamics of the helicopter's translational motion in the NED frame are:

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} T_{11} & T_{12} & \begin{matrix} \cos\sigma\sin\alpha\cos\psi \\ +\sin\sigma\sin\psi \end{matrix} \\ T_{21} & T_{22} & \begin{matrix} \cos\sigma\sin\alpha\sin\psi \\ -\sin\sigma\cos\psi \end{matrix} \\ T_{31} & T_{32} & \cos\sigma\cos\alpha \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix} \tag{8}$$

We make the following assumptions to simplify the model:

1) The helicopter's vertical acceleration is small compared to gravitation acceleration, i.e. $\ddot{Z} \ll g$.
2) The trim values of roll and pitch angles ϕ_{trim} and θ_{trim} are small (within 5 degrees) throughout the flight.
3) The rotor thrust remains the dominant component of the resultant force F, of which the direction in the body fix frame $OX_BY_BZ_B$ stays static.

Thus, with assumption 2) and 3), it is valid to claim $\alpha \approx \theta - \theta_{trim}$ and $\sigma \approx \phi - \phi_{trim}$ and related terms in Eq.(8) can then be substituted. On the other hand, applying assumption 1) to the third equation expanded from Eq.(8), the resultant force F becomes:

$$F = -\frac{mg}{\cos(\phi - \phi_{trim})\cos(\theta - \theta_{trim})} \tag{9}$$

Replacing F in Eq.(8) with Eq.(9), the dynamic equations describing northward and eastward motion are written as:

$$\begin{aligned} \ddot{X} &= -g\left(\tan(\theta-\theta_{trim})\cos\psi + \frac{\tan(\phi-\phi_{trim})}{\cos(\theta-\theta_{trim})}\sin\psi\right) \\ \ddot{Y} &= -g\left(\tan(\theta-\theta_{trim})\sin\psi - \frac{\tan(\phi-\phi_{trim})}{\cos(\theta-\theta_{trim})}\cos\psi\right) \end{aligned} \tag{10}$$

which draws the implicit connection between helicopter horizontal motion and attitude. In summary, Eq.(7) and Eq.(10) together constitute the simplified dynamic model for the outer-loop path planner.

G. Differential flatness of the simplified dynamic model

Instead of using Eq.(6) directly to form the dynamic constraints in the optimization problem, we will invoke the differential flatness property of the model to transform Eq.(6) into an equivalent system, which will prove to be computationally more efficient.

We now recall the definition of the differential flat system, as below [15].

Definition: A system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$ with $\mathbf{x}\in\mathbb{R}^n, \mathbf{u}\in\mathbb{R}^m$ is differential flat if there exists a flat output $\mathbf{y}\in\mathbb{R}^m$ in the form:

$$\begin{aligned} & \mathbf{y} = \Phi(\mathbf{x},\mathbf{u},\dot{\mathbf{u}},\cdots,\mathbf{u}^{(i)}) \\ \text{such that} \quad & \mathbf{x} = \Phi_x(\mathbf{y},\dot{\mathbf{y}},\cdots,\mathbf{y}^{(j-1)}) \\ & \mathbf{u} = \Phi_u(\mathbf{y},\dot{\mathbf{y}},\cdots,\mathbf{y}^{(j-1)},\mathbf{y}^{(j)}) \end{aligned} \tag{11}$$

where Φ, Φ_x and Φ_u are smooth functions.

For the system represented by Eq.(6), the flat output can be chosen as:

$$\mathbf{y}_{ol} = \left[X, Y, Z, \psi\right]^T \tag{12}$$

Upon the selection of flat output $\mathbf{y}_{ol}$ in Eq.(12), the mappings from $\mathbf{y}_{ol}$, $\dot{\mathbf{y}}_{ol}$ and $\ddot{\mathbf{y}}_{ol}$ to $\mathbf{x}_{ol}^{sub} = \left[\psi, X, Y, Z, \dot{\psi}, \dot{X}, \dot{Y}, \dot{Z}\right]^T$ and $\mathbf{u}_{ol}^{sub} = \left[u_3, u_4\right]^T$ are directly found, leaving the mappings to ϕ, θ and their time derivatives to be determined next.

Notice that Eq.(10) can be transformed into the form:

$$\begin{aligned} \theta &= -\arctan\left(\frac{\ddot{X}}{g}\cos\psi + \frac{\ddot{Y}}{g}\sin\psi\right) + \theta_{trim} \\ \phi &= \arctan\left(\frac{-\ddot{X}\sin\psi + \ddot{Y}\cos\psi}{\sqrt{g^2 + (\ddot{X}\cos\psi + \ddot{Y}\sin\psi)^2}}\right) + \phi_{trim} \end{aligned} \tag{13}$$

Hence, the mappings to $\dot{\theta}$, $\dot{\phi}$, $\ddot{\theta}$ and $\ddot{\phi}$ can also be obtained by differentiating Eq.(13) once and twice respectively. Note that both X and Y are fourth-differentiable functions of time.

So far, the equivalent system of the one represented by Eq.(6) can be written explicitly in Brunovsky form [15]:

$$\dot{\mathbf{q}}_{ol} = \begin{bmatrix} \mathbf{J}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_2 \end{bmatrix} \mathbf{q}_{ol} + \begin{bmatrix} \mathbf{L}_4 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{L}_2 \end{bmatrix} \mathbf{v}_{ol} \tag{14}$$

where the state vector $\mathbf{q}_{ol} = [X, \dot{X}, \ddot{X}, X^{(3)}, Y, \dot{Y}, \ddot{Y}, Y^{(3)}, Z, \dot{Z}, \psi, \dot{\psi}]^T$ and the input vector $\mathbf{v}_{ol} = \left[X^{(4)}, Y^{(4)}, \ddot{Z}, \ddot{\psi}\right]^T$, and:

$$\mathbf{J}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{L}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{J}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{L}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{15}$$

The input vector $\mathbf{u}_{ol}$ of the original system can be recovered from the input vector $\mathbf{v}_{ol}$ of the equivalent system by:

$$\begin{aligned} u_1 &= f_1(\mathbf{q}_{ol}) + g_{11}(\mathbf{q}_{ol})v_1 + g_{12}(\mathbf{q}_{ol})v_2 \\ u_2 &= f_2(\mathbf{q}_{ol}) + g_{21}(\mathbf{q}_{ol})v_1 + g_{22}(\mathbf{q}_{ol})v_2 \\ u_3 &= -v_3 \\ u_4 &= v_4 \end{aligned} \tag{16}$$

where f_1, f_2, g_{11}, g_{12}, g_{21} and g_{22} in Eq.(16) can be determined by the second differentiation of Eq.(13).

Remark: By representing the system in the form of Eq.(14), the original nonlinear dynamics in Eq.(6) are transformed into an equivalent set of linear kinematics, which is well-suited for trajectory planning. On the other hand, the nonlinear terms in the original dynamics Eq.(10) are transferred into Eq.(13), which are enforced by inequality constraints in the optimization problem.

III. Time-optimal trajectory generation algorithm for outer-loop path planner

So far, a linear kinematic model that captures crucial characteristics of helicopter outer-loop motion has been developed in Section II-G. In this part, based on the above model, corresponding cost function and multiple constraints are stated and formulated into an optimization problem, of which the solution is the time-optimal reference trajectory for helicopter shipboard landing.

A. Cost function

The objective function of the optimization problem is the total approach time or landing flight time based on the specific scenarios, defined by:

$$\min \quad J = t_f - t_0, \tag{17}$$

where t_0 is the time at which the optimization algorithm is triggered, and t_f is the time instant (to be determined) when the helicopter reaches the terminal state (either end of landing or end of approach).

B. Discretized kinematics

For computational tractability, the trajectory being optimized is discretized into N segments which is a preselected constant. Hence, the sampling time t_s is:

$$t_s = \frac{t_f - t_0}{N-1} \tag{18}$$

Then, by assuming the input $\mathbf{v}_{ol}$ to be zero-order hold (ZOH) signal, the continuous kinematics in Eq.(14) can be discretized by:

$$\mathbf{q}(k+1) = \mathbf{F}\mathbf{q}(k) + \mathbf{G}\mathbf{v}(k) \tag{19}$$

where $\mathbf{q}(k) = [X(k), X_v(k), X_a(k), X_j(k), Y(k), Y_v(k), Y_a(k), Y_j(k), Z(k), Z_v(k), \psi(k), \psi_v(k)]^T$ is the discretized state vector and $\mathbf{v}(k) = [X_s(k), Y_s(k), Z_a(k), \psi_a(k)]^T$ is the discretized input vector at the k^{th} node. The subscripts v, a, j and s represent velocity, acceleration, jerk and snap respectively.

Overall, the optimization variables are comprised of t_f, $\mathbf{q}(k)$ and $\mathbf{v}(k)$ for all $k \in [1,N] \subseteq \mathbb{N}$, of which the concatenated vector belongs to $\mathbb{R}^{(16N+1)\times 1}$.

C. Optimization constraints

The optimization constraints used by the optimizer are listed entirely below, of which some may be activated for either approach or landing trajectory generation only.

1) Discretized kinematics constraints: The kinematics in Eq.(19) governing the forward propagation of the trajectory should be always satisfied during the optimization process, that is:

$$\mathbf{q}(k+1) - \mathbf{F}\mathbf{q}(k) - \mathbf{G}\mathbf{v}(k) = \mathbf{0}, \ \forall k \in [1, N-1] \subset \mathbb{N} \quad (20)$$

2) Boundary condition constraints: The initial and terminal condition constraints have the generalized form:

$$\mathbf{q}(1) = \mathbf{q}_0, \ \mathbf{q}(N) = \mathbf{q}_f \quad (21)$$

The initial condition $\mathbf{q}_0$ is decided by the fuselage state $\mathbf{x}_f$ at t_0 when the algorithm is executed. All terminal conditions in $\mathbf{q}_f$ except for vertical motion terms are dictated by the deck motion state $\mathbf{x}_D$ and its time derivatives at the terminal time t_f. Note that high-order time derivatives including X_{j0}, Y_{j0}, X_{jf} and Y_{jf} in $\mathbf{q}_0$ and $\mathbf{q}_f$ are set free.

On the other hand, Z_f and Z_{vf} are chosen differently for the approach and landing scenarios. For the approach trajectory, the helicopter ends up hovering at some predetermined altitude, which makes:

$$Z_f = Z_{D0} + Z_{HOV}, \ Z_{vf} = 0 \quad (22)$$

where Z_{HOV} is set to $-6m$ for all approach trajectories in this paper. For the landing trajectory, the helicopter is required to touch down on the deck with zero vertical relative velocity, which makes:

$$\begin{aligned} Z_f &= Z_{TOL} + Z_{D0} + A_D \sin(2\pi t_f \big/ T_D + \phi_D), \\ Z_{vf} &= 2\pi A_D \big/ T_D \cos(2\pi t_f \big/ T_D + \phi_D) \end{aligned} \quad (23)$$

where Z_{TOL} is the landing safety tolerance accounting for helicopter size (i.e. distance from the center of gravity to the landing gear plane) and other safety redundancies.

Remark: By splitting the entire flight trajectory into approach/landing phases and setting different terminal constraints for vertical motion, the optimizer is freed from solving for the deck terminal height $Z_D(t_f)$ and vertical velocity $\dot{Z}_D(t_f)$ during the approach trajectory planning, which saves computational cost.

3) Input constraints: Ideally, the input constraints should be implemented by setting upper and lower bounds directly for Eq.(16); however, the computational cost of iteratively solving for u_1 and u_2 in Eq.(16) is enormous because of the complexity of these functions. On the other hand, it is easy to show that u_1 and u_2 can be bounded by simply constraining $\ddot{X}$, $X^{(3)}$, $X^{(4)}$, $\ddot{Y}$, $Y^{(3)}$, $Y^{(4)}$, $\dot{\psi}$ and $\ddot{\psi}$. Hence, the constraints of u_1 and u_2 are transferred to the constraints of other states and inputs, leaving the input constraint to be:

$$\left|\mathbf{v}^{sub}(k)\right| \leq \mathbf{v}^{sub}_{\max}, \forall k \in [1,N] \subset \mathbb{N}. \quad (24)$$

where $\mathbf{v}^{sub}(k) = \left[Z_a(k), \psi_a(k)\right]^T$.

4) State constraints: The state constraints are written as:

$$\left|\mathbf{q}^{sub}(k)\right| \leq \mathbf{q}^{sub}_{\max}, \forall k \in [1,N] \subset \mathbb{N}. \quad (25)$$

where $\mathbf{q}^{sub}(k) = \left[Z_v(k), \psi_v(k), X_j(k), Y_j(k), X_s(k), X_s(k)\right]^T$. Except for the above states, the horizontal velocity $V(k)$, the roll angle $\phi(k)$ and pitch angle $\theta(k)$ are also constrained by:

$$\mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{q}(k)) \leq \mathbf{h}_{\max}, \forall k \in [1,N] \subset \mathbb{N}. \quad (26)$$

where $\mathbf{h}(\mathbf{q}(k))$ includes $V(k) = \sqrt{X_v^2(k) + Y_v^2(k)}$ and Eq.(13).

5) Additional constraints: Aside from the above common constraints, scenario specific constraints must also be enforced for the different flight phases.

The horizontal airspeed is comparatively high during approach, so that the side-slip angle is constrained by:

$$\begin{aligned} \beta(k) = \psi(k) - \arctan 2(Y_v(k), (X_v(k)) = 0, \\ \forall k \in [1,N] \subset \mathbb{N}. \end{aligned} \quad (27)$$

In some case, the approach trajectory also needs to avoid certain obstacles with the boundary function being $\gamma(X,Y,Z) = 0$, with the relative constraint becoming:

$$\gamma(X(k), Y(k), Z(k)) \geq 0, \forall k \in [1,N] \subset \mathbb{N}. \quad (28)$$

Moreover, there may exist aggressive horizontal maneuvers that make the snaps X_s and Y_s noisy during approach, hence the finite impulse response (FIR) filter is used to regulate the infinite norm of the high frequency signals:

$$\begin{gathered} \left|\frac{X_s(k+1) + X_s(k-1)}{2} - X_s(k)\right| \leq \tilde{X}_{s\max}, \\ \left|\frac{Y_s(k+1) + Y_s(k-1)}{2} - Y_s(k)\right| \leq \tilde{Y}_{s\max}, \\ \forall k \in [2, N-1] \subset \mathbb{N}. \end{gathered} \quad (29)$$

On the other hand, to improve the landing safety, additional constraints are enforced on landing trajectory. Anti-collision constraint is implemented by:

$$Z(k) \leq Z_{TOL} + Z_D(k), \forall k \in [1,N] \subset \mathbb{N}. \quad (30)$$

where $Z_D(k)$ is the deck position sampled at the k^{th} time step.

Due to model inaccuracy, actuator delay and possible deck rolling/pitching, there will inevitably be some relative velocity between the helicopter and the deck at touch down. Such situations can be remedied by picking a window time when the deck motion is relatively quiescent for the helicopter to land. This is achieved by constraining the absolute velocity of deck heaving at the touch down time instant:

$$\left|2\pi A_D \Big/ T_D \cos(2\pi t_f \Big/ T_D + \phi_D)\right| \leq V_{D\max} \tag{31}$$

where $V_{D\max}$ is the threshold constant that can be tuned for different conditions.

D. Formulation of the optimization problem

With the cost function and optimization constraints listed above, the optimization problem can be written as:

$$\begin{aligned} \underset{t_f,\mathbf{q}(k),\mathbf{v}(k)}{\arg\min} \quad & J = t_f - t_0 \\ s.t \quad & \Gamma_{\min} \leq \Gamma(t_f,\mathbf{q}(k),\mathbf{v}(k)) \leq \Gamma_{\max}, \\ & \Gamma^{ap}_{\min} \leq \Gamma^{ap}(t_f,\mathbf{q}(k),\mathbf{v}(k)) \leq \Gamma^{ap}_{\max}, \\ & \text{ if approach.} \\ & \Gamma^{ld}_{\min} \leq \Gamma^{ld}(t_f,\mathbf{q}(k),\mathbf{v}(k)) \leq \Gamma^{ld}_{\max}, \\ & \text{ if landing.} \end{aligned} \tag{32}$$

where Γ is the constraint function with Eq.(20), Eq.(21) and Eq.(24)-Eq.(26) stacked up together, Γ^{ap} is comprised of Eq.(27)-Eq.(29), Γ^{ld} is comprised of Eq.(30) and Eq.(31). Note that ”$\leq$” denotes element-wise inequality and 0s are used for both the lower and upper bounds of the equality constraints in Eq.(32).

In this work, the optimization problem Eq.(32) is solved using numerical optimization tool CasADi in MATLAB, with the ‘IPOPT’ solver. The corresponding solution is then stored as the reference trajectory, with which the commands for the inner-loop are scheduled with the current flight time at each time step.

IV. Simulation Results

In this section, we present results from simulation of a variety of flight landing scenarios and the performance of the proposed algorithm under such scenarios. For brevity, we have included salient features of the simulations in the captions of the figures instead of the text.

A. Flight scenarios

1) Approach phase: Three different cases of flight approach scenarios listed in Table I have been used for testing the approach trajectory generation capability of the proposed algorithm. Note that the helicopter starts all three flights at (0m,0m,-70m) in the NED frame with a trim condition at 80 knots (41.156m/s).

TABLE I: Different flight scenarios for approach

case #	shipboard initial position (m,m)	shipboard heading (deg)	obstacle position and radius (m,m,m)
1	(500,350)	0	-
2	(350,350)	30	-
3	(500,350)	30	(500,150,50)

TABLE II: Different deck heaving scenarios

case #	amplitude(m)	period(s)	phase(deg)
1	1.7	8	0
2	2.5	5.5	0

2) Landing phase: Two different cases of deck heave aggressiveness in Table II have been used for testing the landing trajectory generation capability of the proposed algorithm. Note that all landing trajectories start at the terminal state of the approach trajectory in case 3 of Table I.

B. Trajectories generated by the proposed algorithm

1) Approach phase: Fig.6 and Fig.7 show the approach trajectories generated by the proposed algorithm in all three cases. Fig.8 shows the corresponding helicopter attitudes during the approach. Fig.9 shows the comparison between the computational time for the proposed algorithm against the algorithm that does not invoke differential flatness property. The result is obtained by solving the trajectory for case 2 using randomized initial guesses, on a laptop with Intel Core i5-6200U 2.40 GHz.

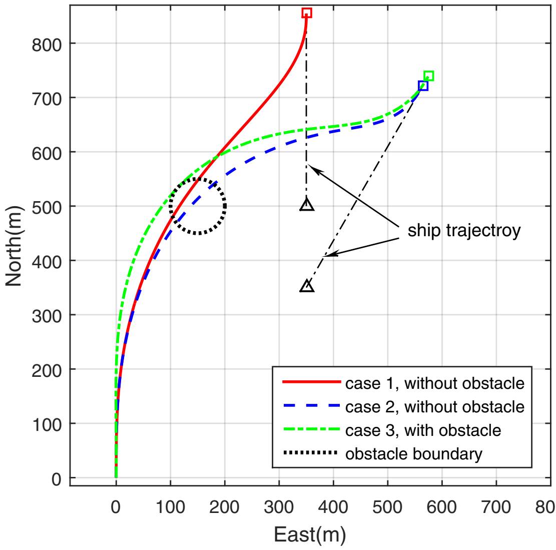

Fig. 6: Ground tracks of the approach trajectories. In all three cases, the helicopter has a successful rendezvous with the ship with the heading angle aligned. The trajectory in case 3 (green dash-dot line) successfully avoids the obstacle. The ability of the proposed algorithm to plan for different flight scenarios with high degree of flexibility is thus demonstrated.

2) Landing phase: Fig.10 shows the trajectories generated by the proposed algorithm for two different cases of deck heaving aggressiveness. For both cases, the trajectories with and without the constraint Eq.(31) activated are shown. For all the landing cases, we use $N = 101$ trajectory segments, leading to an average computational time of 0.25 seconds, which illustrates that the proposed algorithm can be used in real-time iteratively.

C. Implementation on the nonlinear full-state model

1) Approach phase: All three reference approach trajectories generated in Section IV-B.1 have been implemented on the nonlinear helicopter model (with the inner-outer loop

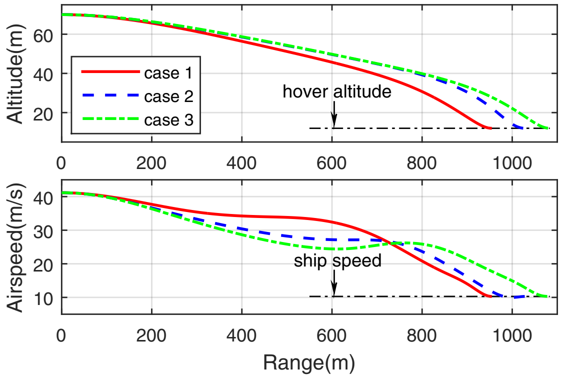

Fig. 7: Altitude and airspeed versus range profiles during approach. In all three cases, the helicopter descends to the prescribed hover altitude, while decelerating to the same speed as the ship at the end of the approach phase. Note that the proposed algorithm has different deceleration times for different flight scenarios so that the flight time can be minimized, while satisfying all the constraints.

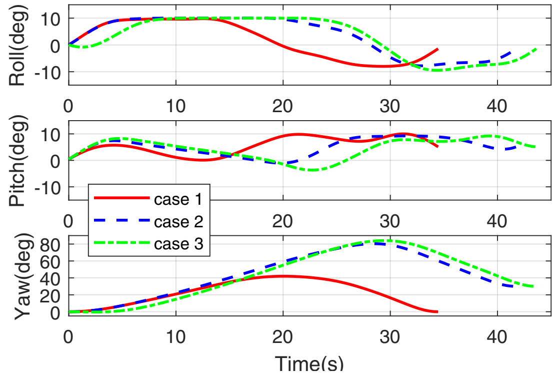

Fig. 8: Helicopter attitude time histories during approach. The pitch and roll angle are well constrained by Eq.(26), which proves the effectiveness of exploiting the differential flatness of the model. Moreover, the anti-sideslipping constraint Eq.(27) enforces the yaw angle to align with the flight azimuth angle throughout the approach.

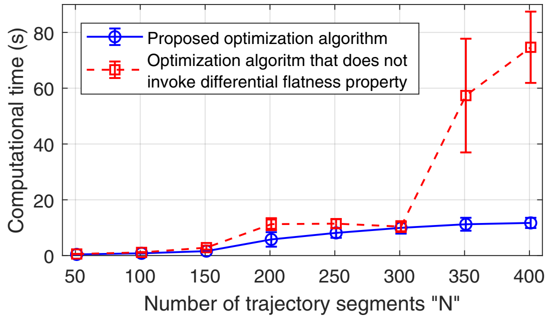

Fig. 9: Computational time comparison between the proposed algorithm and the one that does not invoke differential flatness property. Overall, the proposed algorithm has higher computational efficiency. As the number of trajectory segments N increases, the computational time taken by the proposed algorithm increases gradually. On the other hand, the other algorithm (i.e. the one that does not invoke differential flatness) takes enormous computational effort to converge for large N, while the computational time taken is very sensitive to different initial guesses.

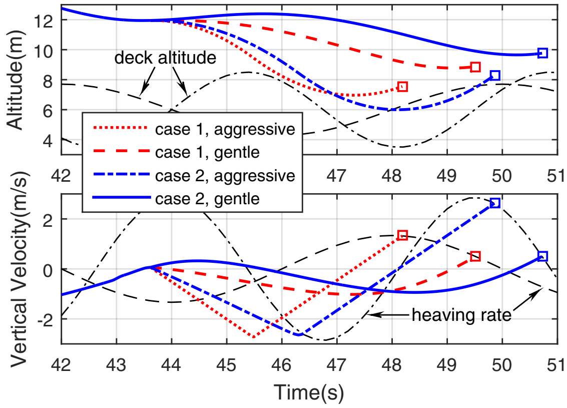

Fig. 10: Vertical motions of the reference trajectories for landing. Although all trajectories satisfied terminal position and velocity constraints, the two trajectories without constraints on the terminal deck heaving rate appear to be much more aggressive, which corresponds with minimum landing time. On the other hand, the two gentler trajectories hover longer and seek to land when the deck heaving is more quiescent.

structure described in Section II-A). However, for clarity, Fig.11 only shows the simulation result from case 3, while Fig.12 shows the actual inputs fed into the nonlinear helicopter model. On the other hand, Table III lists the terminal errors of actual approach trajectories for all three cases.

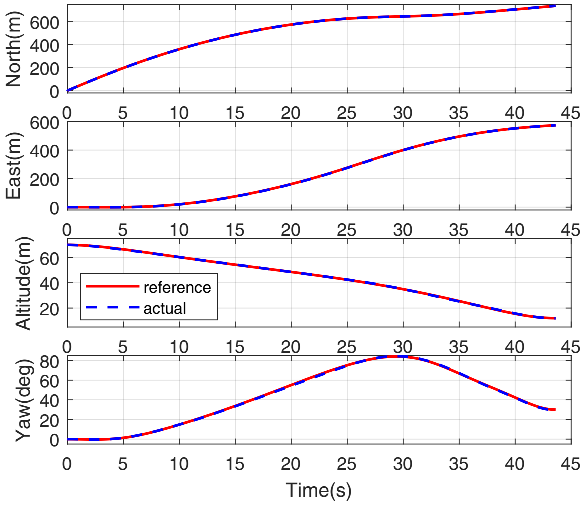

Fig. 11: Tracking history of the flat outputs during approach. Throughout the nonlinear simulation, all the actual flat outputs tightly track the references generated by the proposed algorithm. This is benefited by both the high resolution provided by the simplified outer-loop model and the responsive inner-loop LDI controller.

2) Landing phase: All four reference landing trajectories generated in Section IV-B.2 have been implemented on the nonlinear helicopter model. Similarly, Fig.13 only shows the simulation results of the two trajectories in case 2 for clarity, while Table IV lists the terminal errors of all four actual landing trajectories.

V. Conclusions

In this paper, a time-optimal trajectory generation algorithm for helicopter shipboard landing is proposed. By utilizing the equivalent model converted from the simplified helicopter dynamics via differential flatness, together with the introduction of corresponding cost function and constraints, the algorithm formulates a standard optimization

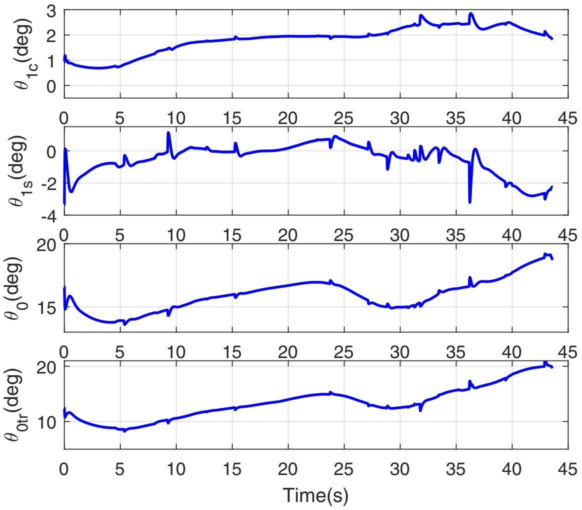

Fig. 12: History of the actual inputs fed into the nonlinear helicopter model during approach.

TABLE III: Terminal errors of the approach trajectories. In all three cases, the actual approach trajectories end very close to the reference trajectory. Hence, despite the difference of the flight scenarios, the actual fuselage state of helicopter relative to the ship is basically static at the end of the approach, which simplifies the landing trajectory planning hugely.

case #	horizontal position (m)	vertical position (m)	horizontal velocity (m/s)	vertical velocity (m/s)	heading angle (deg)
1	0.9179	-0.0212	0.1376	0.0029	0.067
2	0.4116	0.2154	0.044	-0.0484	1.2136
3	0.4574	-0.0514	0.0597	0.0632	-0.7372

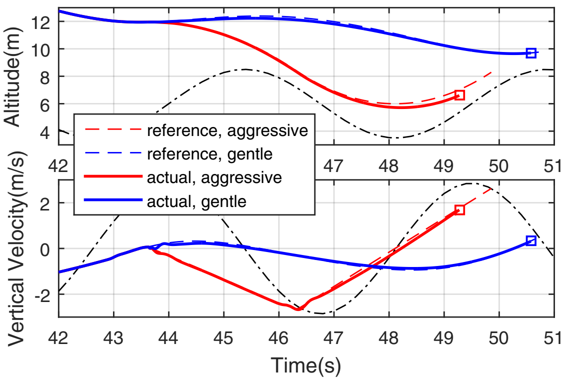

Fig. 13: Tracking history of vertical motion during lading. Due to the factors such as actuator delay, the helicopter cannot accelerate and decelerate as swiftly as planned by the proposed algorithm. Therefore, it is more difficult for helicopter to follow the more aggressive trajectory, which leads to a touch down ahead of schedule and a comparatively larger relative normal velocity to the deck. On the other hand, the implementation of the gentler trajectory leads to a much softer landing, with only 1.3 second increase in flight time

problem, which can be solved efficiently for the time-optimal reference trajectories capable of obstacle avoidance and large heading maneuvers. The proposed algorithm is compatible with the standard LDI controller, which is used to track the references. Simulations conducted on a full-state nonlinear helicopter model show the effectiveness of the proposed method.

TABLE IV: Terminal errors of the landing trajectories. Despite a larger relative position errors, the relative normal velocity, which is of more concern, are obviously decreased by the implementation of the gentler reference trajectories.

case #	landing strategy	horizontal position (m)	tangential velocity (m/s)	normal velocity (m/s)
1	aggressive	0.4932	0.2812	1.1343
1	gentle	1.0421	0.3055	0.7909
2	aggressive	0.83	0.3796	1.0584
2	gentle	1.3602	0.3430	0.6384

VI. Acknowledgments

This work was sponsored by the Office of Naval Research (ONR), under contract number N00014-16-1-2705.

References

[1] M. Voskuijl, G. Padfield, D. Walker, B. Manimala, and A. Gubbels, “Simulation of automatic helicopter deck landings using nature inspired flight control,” *The Aeronautical Journal*, vol. 38, no. 12, pp. 25–34, Aug 2010.

[2] Q. Truong, T. Rakotomamonjy, A. Taghizad, and J. Biannic, “Vision-based control for helicopter ship landing with handling qualities constraints,” *20th IFAC Symposium on Automatic Control in AerospaceACA 2016 Sherbrooke, Quebec, Canada*, vol. 49, pp. 118–123, Aug 2016.

[3] T. D. Ngo and C. Sultan, “Model predictive control for helicopter shipboard operations in the ship airwakes,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 574–589, 2016.

[4] G. Soneson, J. Horn, J. Yang, and A. Zheng, “Simulation testing of advanced response types for ship-based rotorcraft,” *Journal of the American Helicopter Society*, vol. 61, no. 3, pp. 1–13, July 2016.

[5] J. Horn, J. Yang, C. He, D. Lee, and J. Tritschler, “Autonomous ship approach and landing using dynamic inversion control with deck motion prediction,” *41st European Rotorcraft Forum 2015, Munich, Germany*, vol. 2, pp. 864–877, Sep 2015.

[6] G. Meyer, R. Su, and L. Hunt, “Application of nonlinear transformations to automatic flight control,” *Automatica*, vol. 20, no. 1, pp. 103–107, 1984.

[7] S. Lane and R. Stengel, “Flight control using nonlinear inverse dynamics,” *Automatica*, vol. 24, pp. 471–483, 1988.

[8] A. Chamseddine, T. Li, Y. Zhang, C. A. Rabbath, and D. Theilliol, “Flatness-based trajectory planning for a quadrotor unmanned aerial vehicle test-bed considering actuator and system constraints,” in *2012 American Control Conference (ACC)*, June 2012, pp. 920–925.

[9] W. Van Loock, G. Pipeleers, and J. Swevers, “Time-optimal quadrotor flight,” in *Proc. of the 2013 European Control Conference (ECC)*, 2013, pp. 1788–1792.

[10] M. Hehn and R. DAndrea, “Real-time trajectory generation for quadrocopters,” *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 877–892, 2015.

[11] B. Hu and S. Mishra, “A time-optimal trajectory generation algorithm for quadrotor landing onto a moving platform,” *American Control Conference (ACC), Seattle, USA*, pp. 4183–4188, May 2017.

[12] J. Krishnamurthi and F. Gandhi, “Flight simulation and control of a helicopter undergoing rotor span morphing,” *Journal of the American Helicopter Society*, vol. In Press, Aug 2017.

[13] J. Howlett, “Uh-60a black hawk engineering simulation program: Volume i - mathematical model,” *NASA CR-166309*, 1981.

[14] A. Schwartz, “Systematic characterization of the naval environment (scone) standard deck motion data for a generic surface combatant,” *Memorandum from Office of Naval Research and Naval Surface Warfare Center, Carderock Division*, May 2015.

[15] G. Rigatos, “Nonlinear control and filtering using differential flatness approaches: applications to electromechanical systems,” *Springer*, vol. 25, 2015.