Real-time path planning for time-optimal helicopter shipboard landing via trajectory parametrization

Di Zhao
PhD Student

Sandipan Mishra
Associate Professor

Farhan Gandhi
Redfern Chair in
Aerospace Engineering

Center for Mobility with Vertical Lift
Rensselaer Polytechnic Institute
Troy, NY, United States

ABSTRACT

A real-time path planning algorithm is developed to generate time-optimal trajectory for helicopter shipboard landing. The trajectory optimization problem is translated to the lower dimensional flat output space by exploiting the differential flatness property of the simplified helicopter model. Then, the flat outputs are parameterized using piecewise spline functions with adjustable coefficients, which are used to shape the trajectory and approximate the optimal solution. Further, by allowing the flexible selection of each spline segment's time-duration and enforcing additional path constraints, the time-optimality of the planned trajectory is largely preserved without violation of state and input bounds. Compared to pure temporal discretization methods, the proposed algorithm employs considerably less decision variables and significantly reduces the computational time by 75%, which only leads to a 0.5% growth in the optimal flight time as the trade-off. The improvement in computational efficiency enables the real-time recalculation of the time-optimal trajectories on-the-fly if there are unforeseen deviations from the planned flight path.

INTRODUCTION

Autonomous and pilot-assistive landing is an area of active research because of turbulent shipboard motion, the strong wind over deck (WOD) and the degraded visual environment caused by rough sea states. Moreover, landing often needs to be performed in very limited time under stringent safety constraints. As a result, shipboard recovery continues to be one of the most challenging rotorcraft flight operations. Consequently, the development of the autonomous control strategies for landing becomes imperative.

The autonomous landing problem can be posed in a standard guidance, navigation and control (GNC) framework. In (Refs. 1–3), several state estimation and control strategies have been developed to enable rotorcrafts to successfully land onto moving decks by using onboard computer vision systems. These works focus on incorporating the vision based algorithms into the feedback control loop, where real-time state estimations of both the rotorcraft and the deck are needed in order to execute the flight commands. On the other hand, various feedback controllers applicable for shipboard operations have also been investigated (Refs. 4–7). These control techniques range from dynamic inversion (DI), sliding mode control to backstepping control. Among them, the application of the DI controller on the ship-based helicopter represents an especially well-developed category (Refs. 4, 5), as the effectiveness of the controller has been proved by both human piloted and auto-piloted nonlinear simulations. In addition, the applications of model predictive control (MPC) on helicopter shipboard landing are presented in (Refs. 8, 9). Such control strategies differs from the conventional feedback control as they relies on an internal model to predict and optimize over the landing trajectory for the future time horizon.

Compared to the large body of research on navigation and control methods mentioned above, the development of the guidance algorithms remain quite limited. Current guidance laws typically plan for the reference trajectory by parameterizing single or multiple geometric and/or kinematic variables (Refs. 4, 10, 11). while there have been some approaches aimed towards optimizing the path parameters (Ref. 12), these trajectories normally do not guarantee optimality and lack flexibility.

In contrast to the conventional guidance methods, a recent research presented a new trajectory generation method for time-optimal helicopter shipboard landing (Ref. 13). This approach formulates an optimization problem based on a differential flat model of the simplified helicopter dynamics. Due to the computational efficiency of the formulation, the trajectory generation method is able to provide precise time-optimal reference trajectories with comparatively low computational cost. However, the viability of this method still relies on solving a nonlinear programming (NLP) problem in real-time, whose convergence is significantly influenced by both the problem size and the initial guess of the solution. This

Presented at the Vertical Flight Society 75th Annual Forum & Technology Display, Philadelphia, Pennsylvania, May 13–16, 2019.

hinders the generation of the real-time solution of the reference trajectory, which is critical for landing in the dynamically changing environments. As a result, this paper aims to transform the original optimization problem presented in (Ref. 13) into a computationally efficient one by applying trajectory parameterization techniques. The feasibility of such a scheme comes from the differentially flat nature of the original problem, as only a small subset of the optimization variables needs to be parameterized independently. Ultimately, this computational efficiency in solving the trajectory generation problem empowers the guidance system to work in an MPC-like framework and provide real-time reference trajectory generation during shipboard landing.

HELICOPTER SHIPBOARD LANDING

Problem Statement and Scope

Fig.1 illustrates the autonomous landing problem in the typical GNC framework, where the navigation subsystem measures and estimates the states of both the helicopter and the ship for a given period of time; the guidance subsystem then plans for the desired input and reference trajectories that lead to the final touchdown on the basis of these navigation information; and the control subsystem finally executes these guidance commands by a low-level feedback control scheme.

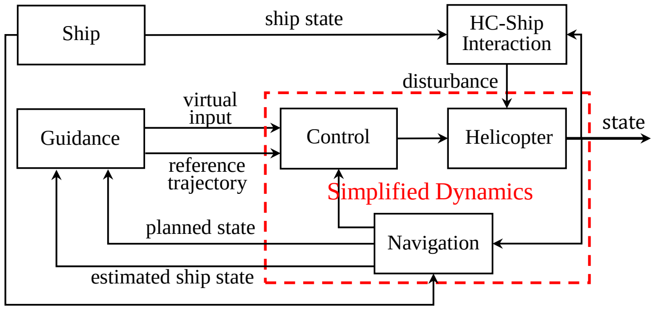

Fig. 1: Overall control architecture for helicopter shipboard landing.

As can be seen in Fig.1, the stochastic nature of the ship motion not only results in a state estimation problem, but also induces disturbances on the helicopter dynamic behavior because of helicopter-ship interaction. Hence, the accuracy of the navigation and the robustness of the control are both necessary requirements for a successful landing. Nonetheless, for the purposes of this paper we assume that the navigation information is accurate throughout the flight, and the controller is effective in regulating the closed-loop dynamics and rejecting the disturbance.

The primary investigation in this paper is on the development of the path planning algorithms for the guidance subsystem for *time-optimal* landing. Since the ship motion cannot be predicted far into the future, the guidance subsystem must achieve real-time (re-)generation of the reference trajectory based on updates of the ship and helicopter states, which eventually enables the safe and time-optimal landing under the stringent constraints in a dynamic environment.

Simplified Helicopter Dynamics

Following the work presented in (Ref. 13), the inner-loop shown in the red dash-line box of Fig.1, formed by the nonlinear helicopter dynamics, the navigation and control subsystems, can be described as a simplified nonlinear model:

$$\dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}(t), \mathbf{v}(t)) \tag{1}$$

where the planned state $\mathbf{p} = [X, Y, Z, \phi, \theta, \psi, \dot{X}, \dot{Y}, \dot{Z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ consists of the helicopter position (north, east, down) in the inertial frame, attitude (roll, pitch, yaw) and their first order derivatives; the virtual input $\mathbf{v} = [v_1, v_2, v_3, v_4]^T$. Specifically, the dynamics in Eq.(1) are determined by:

$$\begin{aligned}
\ddot{X} &= -g\left(\tan(\theta-\bar{\theta})\cos\psi + \frac{\tan(\phi-\bar{\phi})}{\cos(\theta-\bar{\theta})}\sin\psi\right)\\
\ddot{Y} &= -g\left(\tan(\theta-\bar{\theta})\sin\psi - \frac{\tan(\phi-\bar{\phi})}{\cos(\theta-\bar{\theta})}\cos\psi\right)\\
\ddot{Z} &= -v_3\\
\ddot{\phi} &= \quad v_1\\
\ddot{\theta} &= \quad v_2\\
\ddot{\psi} &= \quad v_4
\end{aligned} \tag{2}$$

where g is the gravitational acceleration; $\bar{\phi}$ and $\bar{\theta}$ are trim value for roll and pitch angles at given airspeed. Essentially, the virtual input $\mathbf{v}$ planned by the guidance subsystem directly generates the accelerations of heaving and rotational motion; the change of the helicopter attitude subsequently tilts the thrust and creates the acceleration in the horizontal plane.

Ship Motion

As far as the helicopter shipboard landing problem is concerned, the kinematic information of the ship state required by the guidance subsystem is included in:

$$\mathbf{x}_D = \left[X_D, Y_D, Z_D, \phi_D, \theta_D, \psi_D, \dot{X}_D, \dot{Y}_D, \dot{Z}_D, \dot{\phi}_D, \dot{\theta}_D, \dot{\psi}_D\right]^T \tag{3}$$

While the high-fidelity ship motion data can be directly extracted from the Systematic Characterization of the Naval Environment (SCONE) database (Ref. 14), the guidance subsystem demands the analytic expression of the ship motion, so that the trajectory optimization problem can be solved using gradient descent based method. For this purpose, the time-varying ship motion for a given time period is approximated by explicit sinusoidal functions of the form:

$$\hat{(\cdot)}(t) = \bar{(\cdot)} + \sum_k A_{(\cdot),k} \sin(k\omega_{(\cdot)}t + \phi_{(\cdot),k}) \tag{4}$$

where $\bar{(\cdot)}$ is the time-average, $\omega_{(\cdot)}$ is the frequency, $A_{(\cdot),k}$ and $\phi_{(\cdot),k}$ are the amplitude and phase lag related to the k^{th} harmonic, and $(\cdot)$ denotes the elements in $\mathbf{x}_D$ excluding X_D, (i.e. ship position in surge (forward) direction), as the mean forward velocity $\bar{\dot{X}}_D$ of the ship is non-trivial. Nonetheless, the analytic expression $\hat{X}_D(t)$ can be obtained by integrating

$\hat{\dot{X}}_D(t)$. Overall, this makes the approximated ship state being $\hat{\mathbf{x}}_D = [\hat{X}_D, \hat{Y}_D, \hat{Z}_D, \hat{\phi}_D, \hat{\theta}_D, \hat{\psi}_D, \hat{\dot{X}}_D, \hat{\dot{Y}}_D, \hat{\dot{Z}}_D, \hat{\dot{\phi}}_D, \hat{\dot{\theta}}_D, \hat{\dot{\psi}}_D]^T$.

In this work, the ship motion is assumed to be always known. Therefore, the information in Eq.(4) is accessible to the outer-loop path planner at arbitrary flight time.

FORMULATION OF THE TIME-OPTIMAL TRAJECTORY OPTIMIZATION PROBLEM

General Form of the Trajectory Optimization Problem

With the simplified helicopter dynamics and ship motion defined, a general form of time-optimal trajectory optimization problem can be formulated as:

$$\mathbf{P1:}\quad \underset{t_f^*, \mathbf{p}^*(t), \mathbf{v}^*(t)}{\arg\min} \quad J = \int_{t_0}^{t_f} 1 dt \tag{5a}$$

$$\text{s.t.}\quad \dot{\mathbf{p}} = \mathbf{f}(\mathbf{p}, \mathbf{v}) \tag{5b}$$

$$\mathbf{p}(t_0) = \mathbf{K}_0(\mathbf{p}_0) \tag{5c}$$

$$\mathbf{p}(t_f) = \mathbf{K}_f(\hat{\mathbf{x}}_D(t_f)) \tag{5d}$$

$$\mathbf{L}(\mathbf{p}(t), \mathbf{v}(t)) \leq 0 \tag{5e}$$

where Eq.(5a) represents the minimization of total flight time $t_f - t_0$ through the optimal state $\mathbf{p}^*(t)$ and input $\mathbf{v}^*(t)$; Eq.(5b) constitute the dynamic constraints given by the simplified helicopter dynamics in Eq.(1); Eq.(5c,5d) are the initial and terminal boundary constraints respectively; Eq.(5e) contains the path constraints, which encompass all necessary restrictions on state and input variables during the flight. The concrete form of $\mathbf{K}_0$, $\mathbf{K}_f$ and $\mathbf{L}$ will be discussed in a later section.

Reformulation of the Problem via Differential Flatness

P1 is an infinite-dimensional nonlinear optimization problem, finding the solution for which can be quite expensive, if not impractical. In particular, the nonlinear dynamic constraints can add enormous computational cost to the solving process. Hence, it is preferable to reformulate Problem 1 into an equivalent but more computationally efficient form. In (Ref. 13), it was shown that the system represented in Eq.(1) is differential flat (Ref. 13), with the flat output being:

$$\mathbf{o} = \left[X, Y, Z, \psi\right]^T \tag{6}$$

By defining the new state $\mathbf{q} = [X, \dot{X}, \ddot{X}, X^{(3)}, Y, \dot{Y}, \ddot{Y}, Y^{(3)}, Z, \dot{Z}, \psi, \dot{\psi}]^T$ and the new input $\mathbf{w} = [w_1, w_2, w_3, w_4]^T = [X^{(4)}, Y^{(4)}, \ddot{Z}, \ddot{\psi}]^T$, which belong to the set $\{\mathbf{o}, \dot{\mathbf{o}}, \cdots, \mathbf{o}^{(j)}\}$, the smooth functions Φ_p and Φ_v can be determined, so that:

$$\begin{aligned} \mathbf{p} &= \Phi_p(\mathbf{q}) \\ \mathbf{v} &= \Phi_v(\mathbf{q}, \mathbf{w}) \end{aligned} \tag{7}$$

which map $\mathbf{q}$ and $\mathbf{w}$ to the original planned state $\mathbf{p}$ and virtual input $\mathbf{v}$.

Note that the only non-identity functions in Φ_p and Φ_v are those which map to the roll and pitch motions, namely Φ_ϕ, Φ_θ, $\Phi_{\dot{\phi}}$, $\Phi_{\dot{\theta}}$, $\Phi_{\ddot{\phi}}$ and $\Phi_{\ddot{\theta}}$. Specifically, we have:

$$\begin{aligned} \phi &= \Phi_\phi(\ddot{X}, \ddot{Y}, \psi) \\ &= \arctan\left(\frac{-\ddot{X}\sin\psi + \ddot{Y}\cos\psi}{\sqrt{g^2 + (\ddot{X}\cos\psi + \ddot{Y}\sin\psi)^2}}\right) + \bar{\phi} \\ \theta &= \Phi_\theta(\ddot{X}, \ddot{Y}, \psi) \\ &= -\arctan\left(\frac{\ddot{X}}{g}\cos\psi + \frac{\ddot{Y}}{g}\sin\psi\right) + \bar{\theta} \end{aligned} \tag{8}$$

Subsequently, by differentiating Eq.(8) once and twice respectively, we have:

$$\begin{aligned} \dot{\phi} &= \Phi_{\dot{\phi}}(\ddot{X}, X^{(3)}, \ddot{Y}, Y^{(3)}, \psi, \dot{\psi}) \\ \dot{\theta} &= \Phi_{\dot{\theta}}(\ddot{X}, X^{(3)}, \ddot{Y}, Y^{(3)}, \psi, \dot{\psi}) \\ \ddot{\phi} &= \Phi_{\ddot{\phi}}(\ddot{X}, X^{(3)}, X^{(4)}, \ddot{Y}, Y^{(3)}, Y^{(4)}, \psi, \dot{\psi}, \ddot{\psi}) \\ \ddot{\theta} &= \Phi_{\ddot{\theta}}(\ddot{X}, X^{(3)}, X^{(4)}, \ddot{Y}, Y^{(3)}, Y^{(4)}, \psi, \dot{\psi}, \ddot{\psi}) \end{aligned} \tag{9}$$

Further, the dynamics governing the evolution of $\mathbf{q}$ and $\mathbf{w}$ take the form:

$$\dot{\mathbf{q}}(t) = \mathbf{F}\mathbf{q}(t) + \mathbf{G}\mathbf{w}(t) \tag{10}$$

which essentially consists only of a bank of pure integrators.

Consequently, with the mapping in Eq.(7) and the dynamics in Eq.(8) determined, Problem 1 can be reformulated into an equivalent form:

$$\mathbf{P2:}\quad \underset{t_f^*, \mathbf{q}^*(t), \mathbf{w}^*(t)}{\arg\min} \quad J = \int_{t_0}^{t_f} 1 dt \tag{11a}$$

$$\text{s.t.}\quad \dot{\mathbf{q}} = \mathbf{F}\mathbf{q} + \mathbf{G}\mathbf{w} \tag{11b}$$

$$\Phi_p(\mathbf{q}(t_0)) = \mathbf{K}_0(\mathbf{p}_0) \tag{11c}$$

$$\Phi_p(\mathbf{q}(t_f)) = \mathbf{K}_f(\mathbf{x}_D(t_f)) \tag{11d}$$

$$\mathbf{L}(\Phi_p(\mathbf{q}), \Phi_v(\mathbf{q}, \mathbf{w})) \leq 0 \tag{11e}$$

Note that the original nonlinear dynamic constraint in the P1 is replaced by a linear counterpart in P2, while the nonlinearity is transferred into the algebraic boundary and path constraints. Since the algebraic constraints are generally easier to handle in the optimization process, the reformulated P2 has better computational performance than the original P1.

Discretization of the Problem

In (Ref. 13), problem P2 is discretized into a finite-dimensional nonlinear programming (NLP) problem in order to circumvent solving the infinite-dimensional problem and achieve numerical tractability. In practice, the time domain from t_0 to t_f is divided into "$N-1$" equal intervals, making the time step:

$$t_s = \frac{t_f - t_0}{N - 1} \tag{12}$$

where the node number N is fixed in the optimization process to avoid integer programming problem, and t_s essentially becomes the optimization variable. Accordingly, the continuous dynamics in Eq.(10) can be transformed into its discrete counterpart:

$$\mathbf{q}_d(k+1) = \mathbf{F}_d\mathbf{q}_d(k) + \mathbf{G}_d\mathbf{w}_d(k) \tag{13}$$

where $\mathbf{q}_d(k)$ and $\mathbf{w}_d(k)$ are the discrete state and input variable at each node $k \in [1,N]$; and the matrices are computed with:

$$\mathbf{F}_d(t_s) = e^{\mathbf{F}t_s},\ \mathbf{G}_d(t_s) = \int_0^{t_s} e^{\mathbf{F}\tau}\mathbf{G}\, d\tau \tag{14}$$

by using the state transition matrix of $\mathbf{F}$ and assuming zero-order hold of the input signal $\mathbf{w}_d(k)$ over each time interval.

Consequently the discretized problem takes the form:

$$\textbf{P2a:}\quad \underset{\substack{t_s^*,\mathbf{q}_d^*(k),\mathbf{w}_d^*(k)\\ k\in[1,N]}}{\arg\min}\quad J = (N-1)*t_s \tag{15a}$$

$$\text{s.t.}\quad \mathbf{q}_d(k+1) = \mathbf{F}_d\mathbf{q}_d(k) + \mathbf{G}_d\mathbf{w}_d(k) \tag{15b}$$

$$\Phi_p(\mathbf{q}_d(1)) = \mathbf{K}_0(\mathbf{p}_0) \tag{15c}$$

$$\Phi_p(\mathbf{q}_d(N)) = \mathbf{K}_f(\mathbf{x}_D(t_f)) \tag{15d}$$

$$\mathbf{L}(\Phi_p(\mathbf{q}_d(k)),\Phi_v(\mathbf{q}_d(k),\mathbf{w}_d(k)) \leq 0 \tag{15e}$$

Being finite dimensional, P2a takes the form of the nonlinear programming problem and can be solved by interior-point (IpOpt) methods using numerical optimization software packages such as CasADi.

While the local optimum of P2a can be computed easily by using the numerical optimization toolbox, this algorithm still has two disadvantages:

1) The minimization of flight time in P2a is effective only when the node number N in Eq.(12) is relatively large, as it allows the input $\mathbf{w}_d(k)$ to switch frequently during the flight and hence provides adequate degree of freedom to shape the trajectory. However, with the trajectory evenly discretized across the time domain, the input variables at most of the nodes remain the same, as the path constraints are active and no excessive maneuver is required. Therefore, unnecessary computational cost is spent on solving for the information of these trivial nodes through Eq.(13).

2) The solution to P2a only contains the information of optimal state $\mathbf{q}_d^*(k)$ and $\mathbf{w}_d^*(k)$ at the finite N nodes. In the practical implementation, for any flight time $t \in ((k-1)t_s^*, kt_s^*)$ that falls between the k^{th} and $(k+1)^{th}$ node, the actual commanded state $\mathbf{q}_c(t)$ are typically obtained by linearly scheduling the discrete optimal trajectory:

$$\mathbf{q}_c(t) = \mathbf{q}_d^*(k) + \frac{t-(k-1)t_s^*}{t_s^*}(\mathbf{q}_d^*(k+1) - \mathbf{q}_d^*(k)) \tag{16}$$

However, with the commanded input being $\mathbf{w}_c(t) = \mathbf{w}_d^*(k), t \in [(k-1)t_s^*, kt_s^*]$, the exact state $\mathbf{q}(t)$ that satisfy the original continuous dynamics in Eq.(10) should be calculated by:

$$\mathbf{q}(t) = e^{\mathbf{F}(t-(k-1)t_s^*)}\mathbf{q}_d^*(k) + \int_{(k-1)t_s}^{t} e^{\mathbf{F}(t-\tau)}\ \mathbf{G}\mathbf{w}_d^*(k)d\tau \tag{17}$$

Since the magnitude of $||\mathbf{q}_c(t) - \mathbf{q}(t)||$ increases as the time step t_s becomes larger (i.e. node number N becomes smaller), the size of node number N needs to be large enough to bound these deviations, so that the implemented reference trajectory does not have significant discrepancy from the original designed trajectory. On the other hand, although one can obtain the commanded state $\mathbf{q}_c(t)$ by directly using Eq.(17), the computed command could then violate the path constraints as Eq.(15e) is only enforced at the N discrete nodes and there is no guarantee for $\mathbf{q}_c(t)$ to lie within the bounds *in-between* these sample points.

REFORMULATION OF THE TRAJECTORY OPTIMIZATION PROBLEM USING PARAMETERIZATION

Parameterization of the Trajectory

From the discussion above, we know the feasibility of the optimal solution to P2a as a time-optimal trajectory greatly depends on the selection of the discrete node number N. However, the increase in N leads to a larger problem size and invokes considerable computational cost, which adverse the real-time generation of the time-optimal trajectory.

On the other hand, parameterization of the solution using a set of basis functions, rather than discretization, is an alternative way for transcribing P2 into a numerically more tractable formulation. The idea behind such an approach is to approximate the optimal state $\mathbf{q}^*(t)$ and input $\mathbf{w}^*(t)$ in the optimization problem P2 by a set of specific basis functions parameterized by unknown coefficients, in the form:

$$(\cdot)(t) = \sum_{k=0}^{n} a_{(\cdot)k}\varphi_{(\cdot)k}(t) \tag{18}$$

where $(\cdot)$ denotes the elements in the state vector $\mathbf{q}$ and input vector $\mathbf{w}$; $\varphi_k^{(\cdot)}(t)$ represents the k^{th} basis function, with $a_k^{(\cdot)}$ being the corresponding coefficients.

For a general optimization problem, all the elements in the state and input vectors need to be parameterized respectively by Eq.(18), in order to approximate the optimal solution. As for reformulating problem P2, one only needs to form the parametric functions of flat output $\mathbf{o} = [X,Y,Z,\psi]^T$, while the higher order time derivatives in $\mathbf{q}$ and $\mathbf{w}$ get automatically determined. In this paper, the variables in $\mathbf{o}$ are parameterized with M piece-wise polynomial functions (splines):

$$(\cdot)(t) = \sum_{k=0}^{n_{(\cdot)}} a_{(\cdot)k,i}(t-t_i)^k, \forall t \in [t_i, t_{i+1}], i \in [0, M-1] \tag{19}$$

where t_i marks both the end of the i^{th} spline and the beginning of the $(i+1)^{th}$ spline, $n_{(\cdot)}$ determine the highest order of the corresponding variable. To preserve the continuity of the elements in state vector $\mathbf{q}$, $n_X = n_Y = 4$ and $n_Z = n_\psi = 2$ are chosen for the rest of this paper.

Once the parametric functions in Eq.(19) are defined, each independent flat output variable, together with its time derivatives, can be written in the compact form:

$$(\bullet)(t) = \mathbf{\Phi}_{(\cdot)}(t-t_i)\mathbf{a}_{(\cdot),i}, \forall t \in [t_i, t_{i+1}], i \in [0, M-1] \tag{20}$$

where $(\bullet)$ represents $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{\Psi}$, which are vectors containing the related flat output variable and the time derivatives in the state $\mathbf{q}$, i.e. $(\bullet) = \left[(\cdot), \dfrac{d}{dt}(\cdot), \cdots, \dfrac{d^{(n_{(\cdot)}-1)}}{dt}(\cdot) \right]^T$; $\mathbf{a}_{(\cdot),i} = [a_{(\cdot)0,i}, a_{(\cdot)1,i}, \cdots, a_{(\cdot)n_{(\cdot)},i}]^T$ represents the coefficient vector; and the matrix $\mathbf{\Phi}_{(\cdot)}(t-t_i)$ consists of the basis functions and their derivatives:

$$\underbrace{\begin{bmatrix} 1 & t-t_i & \cdots & (t-t_i)^{n_{(\cdot)}-1} & (t-t_i)^{n_{(\cdot)}} \\ 0 & 1 & \cdots & (n_{(\cdot)}-1)(t-t_i)^{n_{(\cdot)}-2} & n_{(\cdot)}(t-t_i)^{n_{(\cdot)}-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & (n_{(\cdot)}-1)! & n_{(\cdot)}!(t-t_i) \end{bmatrix}}_{\mathbf{\Phi}_{(\cdot)}(t-t_i)} \tag{21}$$

Altogether, the state vector $\mathbf{q}$ can be represented by:

$$\mathbf{q}(t) = \mathbf{\Phi}(t-t_i)\mathbf{a}_i, \forall t \in [t_i, t_{i+1}], i \in [0, M-1] \tag{22}$$

where $\mathbf{a}_i = [\mathbf{a}_{X,i}, \mathbf{a}_{Y,i}, \mathbf{a}_{Z,i}, \mathbf{a}_{\psi,i}]^T$ is the concatenated coefficient vector, and the matrix $\mathbf{\Phi}(t-t_i)$ takes the form:

$$\underbrace{\begin{bmatrix} \mathbf{\Phi}_X(t-t_i) & 0 & 0 & 0 \\ 0 & \mathbf{\Phi}_Y(t-t_i) & 0 & 0 \\ 0 & 0 & \mathbf{\Phi}_Z(t-t_i) & 0 \\ 0 & 0 & 0 & \mathbf{\Phi}_\psi(t-t_i) \end{bmatrix}}_{\mathbf{\Phi}(t-t_i)} \tag{23}$$

On the other hand, due to the choice of the highest order $n_{(\cdot)}$, the input vector remains constant for each segment:

$$\mathbf{w}(t) = \begin{bmatrix} n_X! a_{Xn_X,i} \\ n_Y! a_{Yn_Y,i} \\ n_Z! a_{Zn_Z,i} \\ n_\psi! a_{\psi n_\psi,i} \end{bmatrix}, \forall t \in [t_i, t_{i+1}], i \in [0, M-1] \tag{24}$$

Conditions for Continuity

While Eq.(22) comprises piece-wise class $C^{n_{(\cdot)}-1}$ functions of corresponding state variables within each spline, additional conditions need to be satisfied at the junctions of two neighbouring splines in order to guarantee the global continuity of the trajectory. Hence, the right endpoint of the i^{th} spline and the left endpoint of the $(i+1)^{th}$ spline should be equal :

$$\mathbf{q}(t_i) = \mathbf{\Phi}(t_i - t_{i-1})\mathbf{a}_{i-1} = \mathbf{\Phi}(0)\mathbf{a}_i, \forall i \in [1, M-1] \tag{25}$$

Since the last column of $\mathbf{\Phi}(0)$ is trivial in Eq.(25), the first $n_{(\cdot)} - 1$ coefficients $\mathbf{a}_{(\cdot),i}$ of each spline are actually determined by the state $\mathbf{q}(t_i)$ at the corresponding left endpoint. The highest order coefficients $a_{(\cdot)n_{(\cdot)},i}$ can be chosen freely.

Further, we also have $\mathbf{q}(t_0) = \mathbf{\Phi}(0)\mathbf{a}_0$ from the initial condition of the trajectory. Thus, the endpoints of each spline can be propagated in a recursive way:

$$\mathbf{q}(t_i) = \mathbf{F}_p(\Delta t_i)\mathbf{q}(t_{i-1}) + \mathbf{G}_p(\Delta t_i)\mathbf{w}(t_{i-1}), \forall i \in [1, M] \tag{26}$$

where $\Delta t_i = t_i - t_{i-1}$ denotes the length of the i^{th} spline. The matrices in Eq.(26) can be written as:

$$\mathbf{F}_p = \begin{bmatrix} \mathbf{F}_{Xp} & 0 & 0 & 0 \\ 0 & \mathbf{F}_{Yp} & 0 & 0 \\ 0 & 0 & \mathbf{F}_{Zp} & 0 \\ 0 & 0 & 0 & \mathbf{F}_{\psi p} \end{bmatrix}, \mathbf{G}_p = \begin{bmatrix} \mathbf{G}_{Xp} & 0 & 0 & 0 \\ 0 & \mathbf{G}_{Yp} & 0 & 0 \\ 0 & 0 & \mathbf{G}_{Zp} & 0 \\ 0 & 0 & 0 & \mathbf{G}_{\psi p} \end{bmatrix} \tag{27}$$

where the blocks take the form:

$$\begin{gathered} \mathbf{F}_{(\cdot)p}(\Delta t_i) = \begin{bmatrix} 1 & \Delta t_i & \cdots & \dfrac{\Delta t_i^{n_{(\cdot)}-1}}{(n_{(\cdot)}-1)!} \\ 0 & 1 & \cdots & \dfrac{\Delta t_i^{n_{(\cdot)}-2}}{(n_{(\cdot)}-2)!} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\ \mathbf{G}_{(\cdot)p}(\Delta t_i) = \begin{bmatrix} \dfrac{\Delta t_i^{n_{(\cdot)}}}{(n_{(\cdot)})!} & \dfrac{\Delta t_i^{n_{(\cdot)}-1}}{(n_{(\cdot)}-1)!} & \cdots & \Delta t_i \end{bmatrix}^T \end{gathered} \tag{28}$$

With the endpoint values $\mathbf{q}(t_i)$ determined by Eq.(26), they can replace the coefficient vectors $\mathbf{a}_i$ in Eq.(22) and act as the coefficients of the corresponding spline. To parameterize the trajectory of state $\mathbf{q}(t)$ at time $t \in [t_i, t_{i+1}] \subset [t_0, t_f]$, we have:

$$\mathbf{q}(t) = \mathbf{F}_p(t-t_i)\mathbf{q}(t_i) + \mathbf{G}_p(t-t_i)\mathbf{w}(t_i) \tag{29}$$

Note that the left endpoint input $\mathbf{w}(t_i)$ in Eq.(29) remains constant through the entire segment $t \in [t_i, t_{i+1}]$ and can be adjusted to shape the related spline.

Refomulation of the Problem via Parameterization

With the parametric functions of the state $\mathbf{q}(t)$ and the input $\mathbf{w}(t)$ defined, the optimization problem P2 can be reformulated in the following form:

$$\textbf{P2b:} \quad \underset{\substack{\Delta t_i^*, \mathbf{q}^*(t_i), \mathbf{w}^*(t_{i-1}) \\ i \in [1,M]}}{\arg\min} \quad J = \sum_i \Delta t_i \tag{30a}$$

$$\text{s.t.} \quad \mathbf{q}(t_i) = \mathbf{F}_p(\Delta t_i)\mathbf{q}(t_{i-1}) + \mathbf{G}_p(\Delta t_i)\mathbf{w}(t_{i-1}) \tag{30b}$$

$$\mathbf{q}(t) = \mathbf{F}_p(t-t_{i-1})\mathbf{q}(t_{i-1}) + \mathbf{G}_p(t-t_{i-1})\mathbf{w}(t_{i-1}), t \in [t_{i-1}, t_i] \tag{30c}$$

$$\mathbf{w}(t) = \mathbf{w}(t_{i-1}), t \in [t_{i-1}, t_i] \tag{30d}$$

$$\Phi_p(\mathbf{q}(t_0)) = \mathbf{K}_0(\mathbf{p}_0) \tag{30e}$$

$$\Phi_p(\mathbf{q}(t_M)) = \mathbf{K}_f(\mathbf{x}_D(t_f)) \tag{30f}$$

$$\mathbf{L}(\Phi_p(\mathbf{q}(t)), \Phi_v(\mathbf{q}(t), \mathbf{w}(t)) \leq 0 \tag{30g}$$

As can be seen, P2b seeks to optimize the total flight time by designing the optimal length Δt_i^*, optimal coefficients

$\mathbf{q}^*(t_i)$ and $\mathbf{w}^*(t_{i-1})$ (i.e. the endpoint state and input values) of the M splines. The solution can then be used to parameterize the optimal trajectory.

The parameterization problem P2b differs from the discretization problem P2a in the following two aspects:

1) The choice of each spline length Δt_i is flexible. Hence, the trajectory only transitions from one spline to another when the maneuver is required, which avoids placement of numerous nodes in the discretization method.

2) For the implementation of the optimal trajectory, the commanded state $\mathbf{q}_c(t)$ and input $\mathbf{w}_c(t)$ can be directly parameterized by using the optimal coefficients $\mathbf{q}^*(t_i)$ and $\mathbf{w}^*(t_{i-1})$. Since the parametric functions are $C^{n_{(\cdot)}-1}$ smooth and satisfy the dynamics strictly, there is no deviation of the commanded state from the actual dynamics, unlike the discretization approach in P2a.

Because of these two factors, we can now solve for the time-optimal trajectory with fewer number of splines as compared to the discrete nodes used in P2a (i.e. $M \ll N$). Consequently, the computational efficiency of problem P2b is higher than problem P2a.

As far as the numerical implementation of P2b is concerned, Eq.(30a,30b,30e,30f), which are already in discretized form, can be programmed into the numerical solver in a straightforward manner. Since the complete form of the boundary constraints Eq.(30e,30f) resembles its counterpart in P2a and has been discussed fully in (Ref. 13) already, they will not be reviewed in this paper.

On the other hand, Eq.(30g) enforces the path constraint on the continuous functions of state $\mathbf{q}(t)$ in Eq.(30c) and input $\mathbf{w}(t)$ in Eq.(30d), which is numerically intensive. Hence, one needs to conduct some adjustments on Eq.(30g) to improve numerical tractability.

To enforce the path constraint Eq.(30g) in the numerical solver, we sample the state $\mathbf{q}(t)$ and input $\mathbf{w}(t)$ at k time points with Eq.(30c,30d), and then impose the constraint Eq.(30g) on each of these points. Clearly, this will greatly increase the size of the problem P2b and eliminate the advantage of P2b over P2a, which is not preferable.

On the other hand, it is also possible to satisfy path constraint Eq.(30g) by bounding the extremum of each spline, which can be derived analytically from the endpoint values $\mathbf{q}(t_i)$ and $\mathbf{w}(t_{i-1})$ alone. Without introducing any extra sample time points, such method largely preserves the computational efficiency of problem P2b. We discuss this in detail in the next subsection.

Adaptation of the Path Constraint

In order to adapt the path constraint into a numerically manageable form, we first list the specific inequalities in Eq.(30g).

To prevent excessive high slew rates of the state variables during the flight, the horizontal velocity, vertical velocity and the yaw rate of the helicopter are constrained by:

$$\sqrt{\dot{X}^2(t)+\dot{Y}^2(t)} \leq v_{h\max},\ \forall t \in [t_0,t_f] \tag{31a}$$

$$|\dot{Z}(t)| \leq v_{z\max}, \forall t \in [t_0,t_f] \tag{31b}$$

$$|\dot{\psi}(t)| \leq v_{\psi\max}, \forall t \in [t_0,t_f] \tag{31c}$$

In order to limit the aggressiveness of the maneuvers, the horizontal, vertical and yaw accelerations are respectively constrained by:

$$\sqrt{\ddot{X}^2(t)+\ddot{Y}^2(t)} \leq a_{h\max}, \forall t \in [t_0,t_f] \tag{32a}$$

$$|\ddot{Z}(t)| \leq a_{z\max}, \forall t \in [t_0,t_f] \tag{32b}$$

$$|\ddot{\psi}(t)| \leq a_{\psi\max}, \forall t \in [t_0,t_f] \tag{32c}$$

Finally, the roll and pitch angles are also constrained to limit the fuselage attitude:

$$|\phi(t)| = |\Phi_\phi(\ddot{X}(t),\ddot{Y}(t),\psi(t))| \leq \phi_{\max}, \forall t \in [t_0,t_f] \tag{33a}$$

$$|\theta(t)| = |\Phi_\theta(\ddot{X}(t),\ddot{Y}(t),\psi(t))| \leq \theta_{\max}, \forall t \in [t_0,t_f] \tag{33b}$$

where the functions Φ_ϕ and Φ_θ are determined from Eq.(8).

Ideally, the rate and acceleration of roll and pitch angles shown in Eq.(9) should also be constrained directly. However, due to the complexity of the transformations, enforcing these constraints becomes computationally expensive. On the other hand, there is strong correlation between the jerk terms $X^{(3)}$, $Y^{(3)}$ and roll, pitch rates $\dot{\phi}$, $\dot{\theta}$, as well as the snap terms $X^{(4)}$, $Y^{(4)}$ and roll, pitch rates $\ddot{\phi}$, $\ddot{\theta}$. Hence, the roll/pitch rates and accelerations are bounded by constraining the norm of the jerk and snap:

$$\sqrt{\left(X^{(3)}(t)\right)^2+\left(Y^{(3)}(t)\right)^2} \leq j_{h\max}, \forall t \in [t_0,t_f] \tag{34a}$$

$$\sqrt{\left(X^{(4)}(t)\right)^2+\left(Y^{(4)}(t)\right)^2} \leq s_{h\max}, \forall t \in [t_0,t_f] \tag{34b}$$

Aside from the above common constraints, scenario specific constraints must also be enforced for the different flight phases. The airspeed is comparatively high during approach, so that the side-slip angle is constrained by:

$$\beta(t) = \psi(t) - \arctan 2(\dot{Y}(t),\dot{X}(t)) = 0, \forall t \in [t_0,t_f] \tag{35}$$

Note that Eq.(35) is essentially a equality constraint, which indicates the yaw angle of approach phase can be directly parameterized using the parametric functions of the north and east velocity $\dot{X}(t)$, $\dot{Y}(t)$. Accordingly, for the approach phase, we also have:

$$\dot{\psi}(t) = \Phi_{\dot{\psi}}(\dot{X},\ddot{X},\dot{Y},\ddot{Y}) \tag{36a}$$

$$\ddot{\psi}(t) = \Phi_{\ddot{\psi}}(\dot{X},\ddot{X},X^{(3)},\dot{Y},\ddot{Y},Y^{(3)}) \tag{36b}$$

Hence, instead of being constrained by Eq.(31c) and Eq.(32c), the yaw rate and acceleration in the approach phase gets bounded by constraining the horizontal acceleration and jerk in Eq.(32a) and Eq.(34a).

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

$$Z(t) \leq \hat{Z}_D(t) + Z_{TOL}, \forall t \in [t_0, t_f] \tag{37}$$

where $\hat{Z}_D(t)$ is the estimated deck height and Z_{TOL} is the landing safety tolerance accounting for helicopter size.

With the inequalities in Eq.(31-34) and Eq.(37) defined, we can then substitute the time t with the $M+1$ time points $t_i, i \in [0, M]$. By doing so, we actually enforce the path constraint on the endpoints of each spline and thus put the skeleton of the trajectory within the bounds.

Recall that the highest orders of the splines used for the flat output variables are chosen as $n_X = n_Y = 4$ and $n_Z = n_\psi = 2$. Hence, within each spline, this leads to the third order derivatives of $X(t)$ and $Y(t)$ (jerks) being linear functions of time, as well as the fourth order derivatives (snaps) being constant functions. Similarly, the first and second order derivative (velocities and accelerations) of $Z(t)$ and $\psi(t)$ are linear functions of time and constant functions respectively.

Since the extrema of piece-wise linear functions and constant functions locate at the endpoints, one can bound these functions by solely constraining the endpoint values. In other word, the inequalities in Eq.(31b,31c), Eq.(32b,32c), Eq.(34a,34b) get satisfied automatically for all time $t \in [t_0, t_f]$, as long as they are enforced at the endpoints $t_i, i \in [0, M]$ of the M spline functions.

On the other hand, the accelerations $\ddot{X}(t)$, $\ddot{Y}(t)$ and the altitude $-Z(t)$ are quadratic functions of time, while the velocities $\dot{X}(t)$ and $\dot{Y}(t)$ are cubic. Therefore, imposing the inequalities in Eq.(31a), Eq.(32a), Eq.(33a,33b) and Eq.(37) at the endpoints $t_i, i \in [0, M]$ alone is not sufficient for their satisfaction across the entire time domain $[t_0, t_f]$. This situation is illustrated in Fig.2.

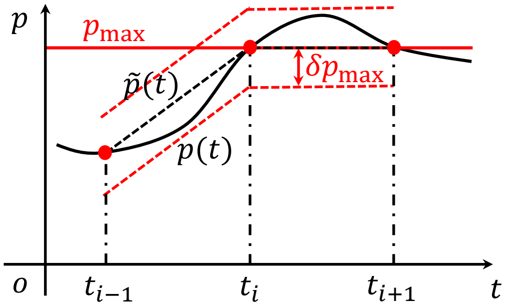

Fig. 2: Illustration of the boundness of the piece-wise quadratic function.

As shown in Fig.2, $p(t)$ is a generic piece-wise quadratic function. It is obvious that constraining $p(t)$ at the endpoints t_i only guarantees the bounds of the related chord line $\tilde{p}(t)$ within the limit $p_{\max}$, while $p(t)$ itself may violate the bound. In addition, for the i^{th} spline of $p(t)$ inside $[t_{i-1}, t_i]$, we have:

$$\begin{aligned} p(t) &= p(t_{i-1}) + v(t_{i-1})t + \frac{1}{2}a(t_{i-1})t^2 \\ \tilde{p}(t) &= p(t_{i-1}) + \left(v(t_{i-1}) + \frac{1}{2}a(t_{i-1})\Delta t_i\right) t \end{aligned} \tag{38}$$

where $v(t)$ and $a(t)$ are the first and second order derivatives of $p(t)$. Since the maximal deviation of the i^{th} quadratic function from its chord line is:

$$\begin{aligned} & \max|p(t) - \tilde{p}(t)| \\ = & \max|\frac{1}{2}a(t_{i-1})\Delta t_i t - \frac{1}{2}a(t_{i-1})t^2| \\ = & \frac{1}{8}a(t_{i-1})\Delta t_i^2 \end{aligned} \tag{39}$$

By enforcing the above deviation to be:

$$\frac{1}{8}a(t_{i-1})\Delta t_i^2 \leq \delta p_{\max}, \forall i = [1, M] \tag{40}$$

we can restrict the piece-wise quadratic function inside the tube with width $\delta p_{\max}$, which is represented by the red dashed lines in Fig.2. Thus, the violation of the bounds can be controlled within a reasonable scope.

For inequality Eq.(32a), we can bound the deviation of $\ddot{X}(t)$ and $\ddot{Y}(t)$ by substituting $a(t)$ in Eq.(40) with $X^{(4)}$ and $Y^{(4)}$ respectively.In addition, assuming the tolerable violation of horizontal acceleration constraint is $\delta a_{h\max}$, the tube width $\delta p_{\max}$ can be selected to be $\frac{\sqrt{2}}{2}\delta a_{h\max}$ for both $\ddot{X}(t)$ and $\ddot{Y}(t)$, which consequently gives:

$$\begin{aligned} |\delta a_h(t)| &= \left| \begin{array}{c} \sqrt{\left(\tilde{\ddot{X}}(t) + \delta\ddot{X}(t)\right)^2 + \left(\tilde{\ddot{Y}}(t) + \delta\ddot{Y}(t)\right)^2} \\ -\sqrt{\tilde{\ddot{X}}^2(t) + \tilde{\ddot{Y}}^2(t)} \end{array} \right| \\ &\leq \sqrt{\delta\ddot{X}^2(t) + \delta\ddot{Y}^2(t)} \\ &\leq \sqrt{2}\max(\delta\ddot{X}^2(t), \delta\ddot{Y}^2(t)) = \delta a_{h\max} \end{aligned} \tag{41}$$

Once the maximal violation of horizontal acceleration constraint is regulated by Eq.(41), the maximal violations of roll and pitch angle constraints are approximately given by:

$$\max|\delta\phi(t)| = \max|\delta\theta(t)| \approx \frac{g\delta a_{h\max}}{g^2 + a_{h\max}} \tag{42}$$

which virtually assures the fulfillment of the inequalities in Eq.(33).

Similarly, concerning the implementation of Eq.(37) during landing, the maximal deviation of the altitude $-Z(t)$ from the corresponding chords can also be bounded by replacing the corresponding variables in Eq.(40). The upper bound of such deviation is selected according to safety factors.

Remark: Although one can use the same strategy discussed above to regulate the maximum violation of the horizontal velocity constraint in Eq.(31a), the cubic form of the velocity

$\dot{X}(t)$ and $\dot{Y}(t)$ leads to a more involved calculation since the deviation of each spline has two extrema that are in a more complex form. At the same time, the constraint on the horizontal velocity is typically less strict than the one on the acceleration in the shipboard landing problem, as the velocity has a general tendency to decrease (to match that of the deck). Hence, for simplicity, velocity constraints Eq.(31a) are only enforced at the endpoints of each spline in this paper.

In addition to execution of the constraint Eq.(40) on the maximal deviation of $\ddot{X}(t)$, $\ddot{Y}(t)$ and $Z(t)$, the flexibility of each spline length is also constrained by:

$$\left| \frac{M\Delta t_i - \sum_{i=1}^{M} \Delta t_i}{\sum_{i=1}^{M} \Delta t_i} \right| \leq F\max, \forall i \in [1, M] \tag{43}$$

Eq.(43) actually limits the difference of each individual spline length from the average one, which avoids the application of extreme long and short splines that impedes the convergence of the optimization problem.

Complete Optimization Problem Formulation

Consequently, with the adaptation of the path constraint in Eq.(30g) discussed in the previous subsection, problem P2b can be reformulated as:

$$\mathbf{P2b}^{\dagger}: \quad \underset{\substack{\Delta t_i^*, \mathbf{q}^*(t_i), \mathbf{w}^*(t_{i-1}) \\ i \in [1,M]}}{\arg\min} \quad J = \sum_i \Delta t_i \tag{44a}$$

$$\text{s.t.} \quad \mathbf{q}(t_i) = \mathbf{F}_p(\Delta t_i)\mathbf{q}(t_{i-1}) + \mathbf{G}_p(\Delta t_i)\mathbf{w}(t_{i-1}) \tag{44b}$$

$$\Phi_p(\mathbf{q}(t_0)) = \mathbf{K}_0(\mathbf{p}_0) \tag{44c}$$

$$\Phi_p(\mathbf{q}(t_M)) = \mathbf{K}_f(\mathbf{x}_D(t_f)) \tag{44d}$$

$$\mathbf{L}^{\dagger}(\Phi_p(\mathbf{q}(t_i)), \Phi_v(\mathbf{q}(t_i), \mathbf{w}(t_{i-1})) \leq 0 \tag{44e}$$

where the adapted path constraint $\mathbf{L}^{\dagger}$ includes the inequalities in Eq.(40) and Eq.(43) in addition to the inequalities that are already contained in the original path constraint $\mathbf{L}$.

Since the optimization problem P2b† is in discrete form, which only consists of algebraic constraints that are executed at the $M+1$ endpoints of the splines, the solution to it (i.e time-optimal trajectory) can be obtained using a numerical optimization solver.

SIMULATION RESULTS

In this section, we present the time-optimal trajectories generated by the proposed algorithm. The flight plan is separated into a approach phase and a landing phase, which will be discussed in the following two subsections, respectively.

Approach Phase

Three cases of approach scenarios have been used for testing the capability of the proposed algorithm. In these cases, the ship travels at a forward speed of 20 knots (10.289 m/s) with different initial positions and heading angles relative to the helicopter, which are listed in Table 1. On the other hand, the helicopter starts at (0m,0m,-70m) in the NED frame with a trim condition at 80 knots (41.156 m/s) in all three cases.

Table 1: Initial ship states in different cases of the approach phase.

case #	North position (m)	East position (m)	Heading angle (deg)
1	500.00	0	0
2	353.55	353.55	0
3	353.55	353.55	30

For all the three cases in Table 1, problem P2b† is solved for the time-optimal trajectories, with $M = 15$ splines used for parameterization. On the other hand, in order to examine the feasibility and optimality of the above trajectories, they are compared with the corresponding benchmark trajectories, which are solved from problem P2a, with $N = 151$ discrete nodes selected.

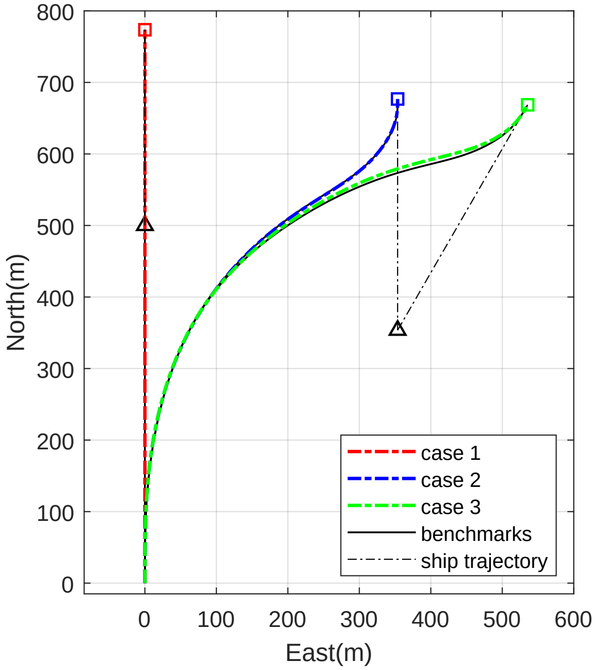

Fig. 3: Ground tracks of the approach trajectories.

Fig.3 shows the ground tracks of the time-optimal approach trajectories solved from P2b† in all three cases. As can be seen, these trajectories resemble their benchmarks well, showing the good approximations to the exact optimal solutions achieved from trajectory parameterization.

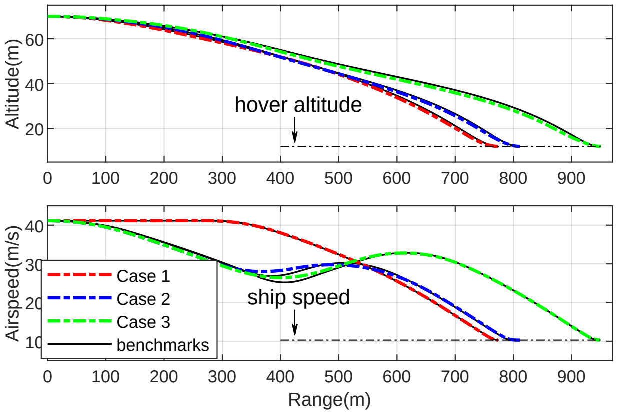

Fig. 4: Altitude and airspeed versus range profiles during approach.

Fig.4 shows the altitude and airspeed profiles of the parameterized trajectories. Similar to the ground tracks, these profiles, shaped by the parametric splines, match the benchmarks closely.

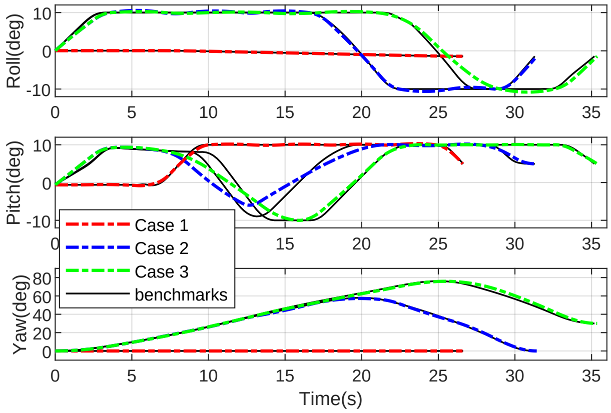

Fig. 5: Helicopter attitude time histories during approach.

Fig.5 shows the corresponding helicopter attitudes time-trace during the approach. Overall, the three Euler angle commands are also well-approximated by the parameterization. There are no notable violations of roll and pitch angle constraints, proving the effectiveness of Eq.(40). On the other hand, small mismatches between the parameterized roll/pitch angles and their benchmarks can be observed in some parts of the trajectories. This is mainly due to the fewer decision variables in the parameterization method compared to the discretization method.

Next, the influence of different selections of the maximal tolerable horizontal acceleration violation $\delta a_{h\max}$, the spline length flexibility $F_{\max}$ and the number of splines M is investigated. For brevity, only the simulation results from case 3 in Table 1 are shown.

Selection of $\delta a_{h\max}$: Fig.6 and Fig.7 show the effect of different choices of the maximal tolerable horizontal acceleration violation, when the flexibility $F_{\max} = 50\%$ and the number of splines $M = 15$.

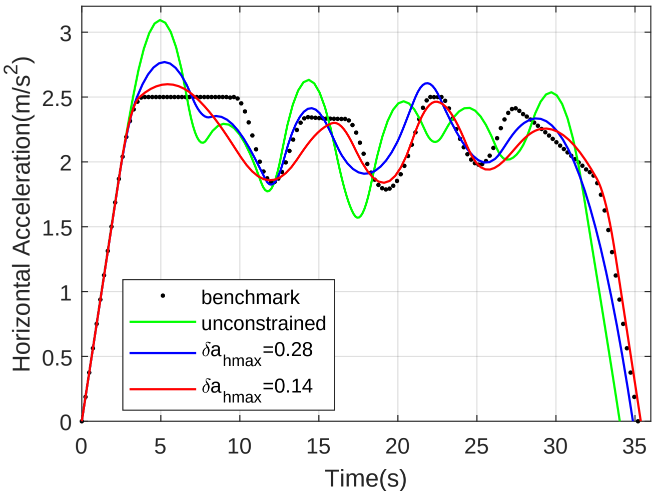

Fig. 6: Helicopter horizontal acceleration with different maximal tolerable horizontal acceleration violation.

As shown in Fig.6, with the maximal tolerable violation $\delta a_{h\max}$ decreasing, the peak horizontal acceleration of the trajectories contract towards the upper bound of the horizontal acceleration $a_{h\max} = 2.5\ \mathrm{m/s^2}$. Accordingly, the flight time increases as the extra acceleration obtained from the violation of the constraints gets restrained.

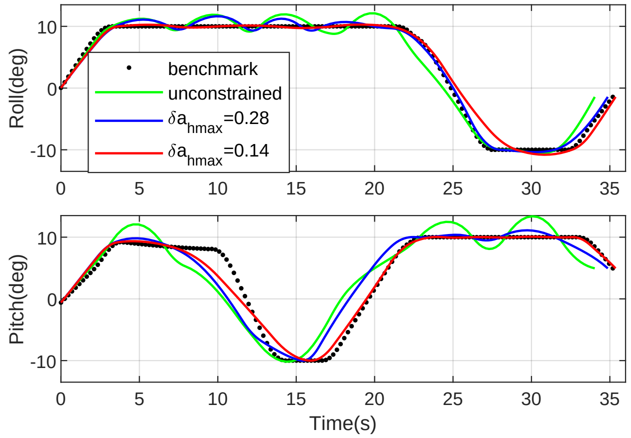

Fig. 7: Helicopter roll and pitch angles with different maximal tolerable horizontal acceleration violation.

Fig.7 shows the similarities between the maximal violations of roll/pitch angle constraints and horizontal acceleration, which is stated in Eq.(42). As can be seen, all notable peaks of the roll and pitch angle profiles are eliminated after $\delta a_{h\max}$ is limited under $0.14\ \mathrm{m/s^2}$, which further results in smoother trajectories of roll and pitch angles.

Selection of Fmax: Fig.8 and Fig.9 show the effect of different choices of spline length flexibility, when the maximal horizontal acceleration violation $\delta a_{h\max} = 0.14\ \mathrm{m/s^2}$ and the number of splines $M = 15$.

As shown in Fig.8, by allowing flexibility of time duration for each spline segment, the algorithm is able to determine better timings for switching the control input by transitioning from one spline to another. This improved distribution of

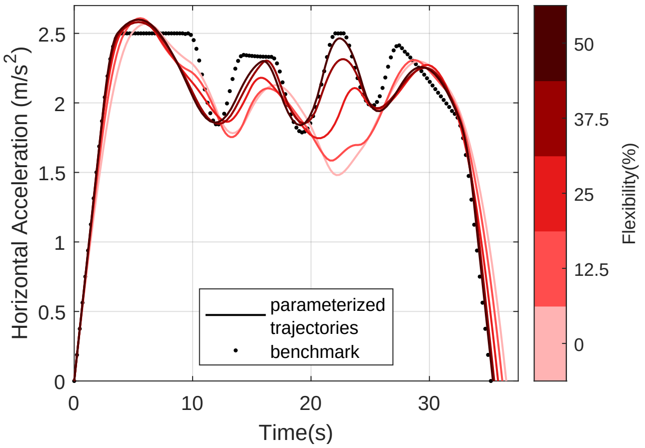

Fig. 8: Helicopter horizontal acceleration with different amounts of flexibility allowed for the spline time durations.

necessary maneuvers leads to a horizontal acceleration profile that fits the benchmark more closely, which relates to the time-optimality of the approach trajectory.

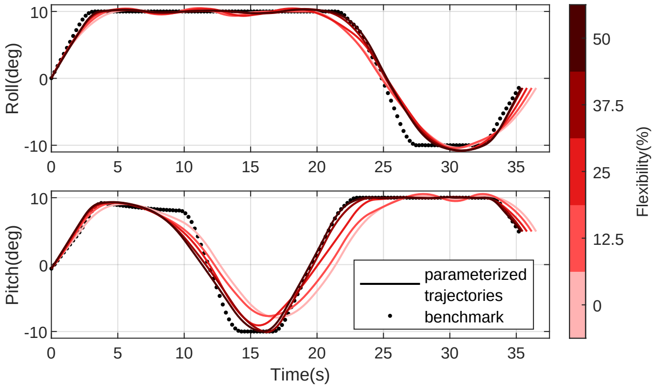

Fig. 9: Helicopter roll and pitch angles with different flexibility of the spline length.

Fig.9 further shows the role of spline length flexibility in shaping the trajectory. As the flexibility increases from 0% (i.e. each spline has the same length) to 50%, the algorithm gains more freedom in placing more splines around the key time points where significant re-directions of roll and pitch angles need to be parameterized with higher accuracy. Contrarily, in the parts where roll and/or pitch angles saturate, the trajectory is approximate with less splines. Consequently, the algorithm is enabled to better locate the optimal timing for turning and decelerating the helicopter during the approach.

Selection of M: Fig.10 and Fig.11 show the effect of different choices of the number of splines, when the maximal horizontal acceleration violation $\delta a_{h\max} = 0.14\ \mathrm{m/s^2}$ and the spline length flexibility $F_{\max} = 50\%$.

Since the number of splines M directly determines the number of decision variables, the corresponding solutions have higher degrees of freedom in shaping the trajectory as we increase M. Therefore, compared to the case in Fig.8 and Fig.9, as M increases from 11 to 15, more intricate maneuvers in trajectories can be observed in Fig.10 and Fig,11. It is interesting to note though, that the total flight time does not significantly reduce.

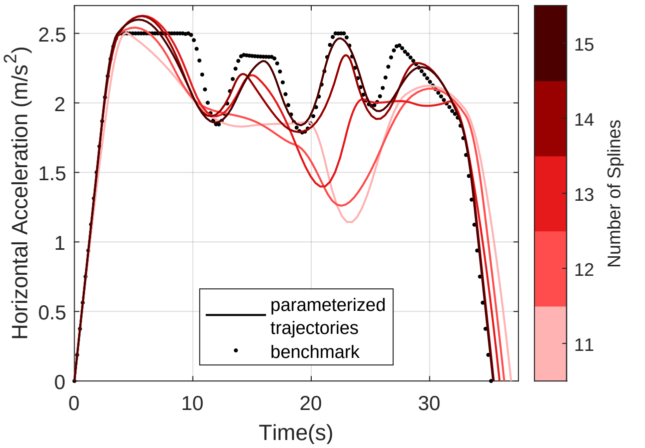

Fig. 10: Helicopter horizontal acceleration with different number of splines.

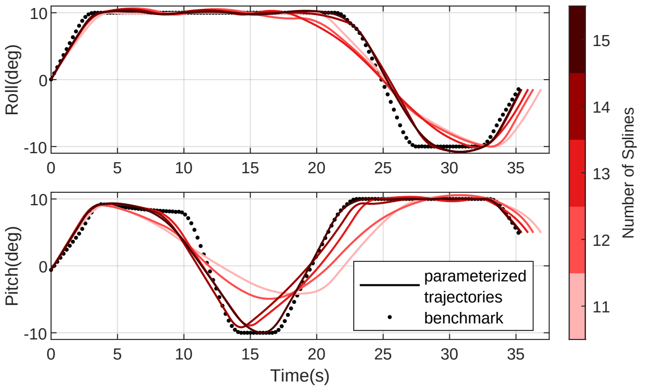

Fig. 11: Helicopter roll and pitch angles with different number of splines.

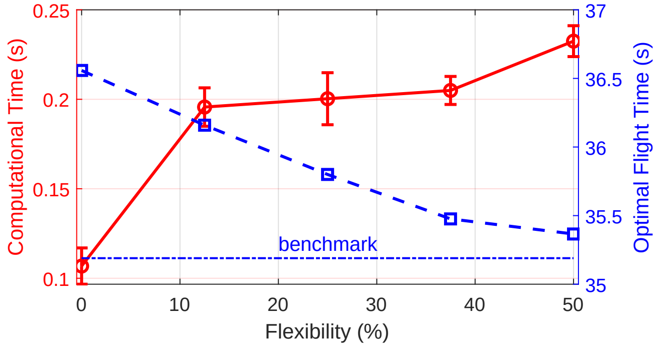

Fig. 12: Trade-off between computational time and optimal flight time accuracy with different levels of spline length flexibility.

Computational Cost: The left axis in Fig.12 shows the computational time spent on solving problem P2b† with different choices of spline length flexibility, while the right axis shows the optimal flight time of the corresponding resultant solution. The result is obtained by solving the trajectory for case 3 in Table 1 with number of splines $M = 16$, on a laptop with Intel Core i5-6200U 2.40 GHz.

Overall, compared to the average 1.25 s cost by solving the benchmark trajectory from problem P2a when $N = 151$, it takes less than 0.3 s to solve problem P2b† for all cases shown in Fig.12, which suggests that the proposed algorithm saves more than 75% of the computational time. On the other hand, by increasing the spline length flexibility from 0% to 50%, the optimal flight time solved from problem P2b† approaches the benchmark solution, with only an 0.5% growth in flight time.

Landing Phase

In this paper, all landing cases are assumed to initiate at 12 m above the sea level with a position hold state relative to the deck. Ship motion characterized by dominant heave movements with medium intensity from the SCONE database are used to fit an analytical expression for the heave motion from Eq.(4).

Similar to the work in (Ref. 13), two landing strategies are used to generate different trajectories: the aggressive strategy seeks for pure time-optimality by maximizing the vertical acceleration and deceleration; the gentle strategy relaxes the aggressiveness of the landing trajectory by constraining the deck heave rate at touch down. The trajectories adopting the above two strategies are solved from Problem P2b† respectively, with both $M = 2$ and $M = 9$ splines used. Additionally, the corresponding benchmark trajectories are also obtained by solving Problem P2a, with $N = 51$ selected.

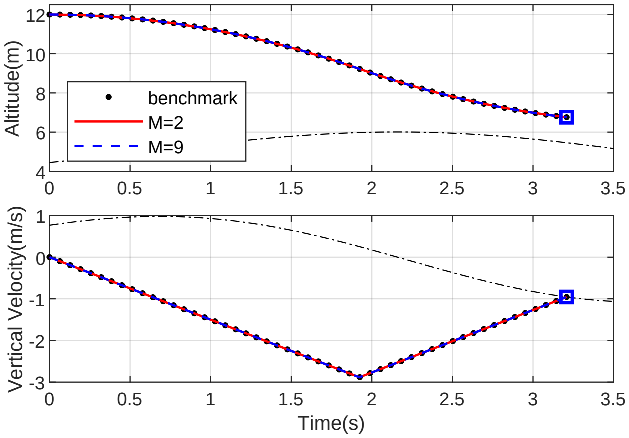

Fig. 13: Trajectories with aggressive landing strategy.

When applying the aggressive landing strategy, the optimal solution to problem P2 is essentially a bang-bang input of the vertical acceleration. Hence, when selecting $M > 2$, problem P2b† can always be solved for the exact optimal solution by optimizing the switch time of the vertical acceleration. This is shown in Fig.13.

On the other hand, the objective of the gentle landing strategy is to obtain a smoother trajectory by allowing the helicopter to wait for a more quiescent deck motion for touchdown. However, as can be seen in Fig.14, the trajectory still ends up with an aggressive bang-bang pattern in the vertical motion when $M = 2$, since the degree of freedom for the algorithm to shape the trajectory is too limited. Nonetheless, the trajectory solved from P2b† already approximates the benchmark well when $M = 9$, which meets the goal of relaxing the aggressiveness of landing.

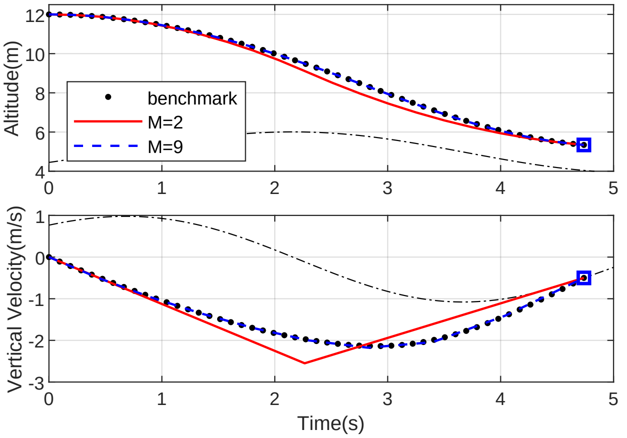

Fig. 14: Trajectories with gentle landing strategy.

Averagely, it takes 0.03 s and 0.05 s to solve problem P2b† for the landing trajectories when selecting $M = 2$ and $M = 9$ respectively. Therefore, compared to the average 0.25 s spent on solving for problem P2a when $N = 51$, the proposed algorithm using trajectory parameterization technique has a significantly higher computational efficiency.

CONCLUSIONS

In this paper, a real time path planning algorithm is proposed for time-optimal trajectory generation in the helicopter shipboard landing problem. The original infinite dimensional optimization problem is reformulated into a numerical tractable form by parameterizing the flat outputs with piece-wise spline functions. Compared to the discretization method applied in a previous work, the proposed method reduces the size of the optimization problem significantly. Additionally, an efficient way of enforcing the path constraint is developed to restrict key state and input variables throughout the flight. As a result, the trajectory generated by the proposed method is feasible and time-optimal, and can be solved in real-time.

Author contact information: Di Zhao, zhaod3@rpi.edu; Sandipan Mishra, mishrs2@rpi.edu; Farhan Gandhi, gandhf@rpi.edu

ACKNOWLEDGMENTS

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

This work was also supported in part by the Center for Mobility with Vertical Lift (MOVE) at Rensselaer Polytechnic Institute, Troy, NY.

REFERENCES

[1] Lange, S., Sunderhauf, N., and Protzel, P., “A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments,” 2009 International Conference on Advanced Robotics, Munich, Germany, June 2009.

[2] Truskin, B. L. and Langelaan, J. W., “Vision-based Deck State Estimation for Autonomous Ship-board Landing,” American Helicopter Society 69th Annual Forum, Phoenix, AR, May 2013.

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

[4] Horn, J. F., Yang, J., He, C., Lee, D., and Tritschler, J., “Autonomous Ship Approach and Landing Using Dynamic Inversion Control with Deck Motion Prediction,” 41st European Rotorcraft Forum, Munich, Germany, September 2015.

[5] Soneson, G. L., Horn, J. F., Yang, J., and Zheng, A., “Simulation Testing of Advanced Response Types for Ship-Based Rotorcraft,” *Journal of the American Helicopter Society*, Vol. 61, (3), July 2016, pp. 1–13.

[6] Ahmed, B. and Pota, H. R., “Backstepping-based Landing Control of a RUAV using Tether Incorporating Flapping Correction Dynamics,” 2008 American Control Conference (ACC), Seattle, WA, June 2008.

[7] Nonaka, K. and Sugizaki, H., “Integral Sliding Mode Altitude Control for a Small Model Helicopter with Ground Effect Compensation,” 2011 American Control Conference (ACC), San Francisco, CA, June 2011.

[8] Ngo, T. D. and Sultan, C., “Model Predictive Control for Helicopter Shipboard Operations in the Ship Airwakes,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, (3), 2016, pp. 574–583.

[9] Prasad, J. V. R., Comandur, V., Walters, R., and Guerrero, D., “Model Predictive Path Integral Approach for Trajectory Guidance of Rotorcraft Shipboard Landing,” American Helicopter Society 74th Annual Forum, Phoenix, AR, May 2018.

[10] Voskuijl, M., Padfield, G. D., Walker, D. J., Manimala, B. J., and Gubbels, A. W., “Simulation of Automatic Helicopter Deck Landings Using Nature Inspired Flight Control,” *The Aeronautical Journal*, Vol. 38, (12), August 2010, pp. 25–34.

[11] Halbe, O. and Hajek, M., “A Methodology towards Rotorcraft Piloting Autonomy for Approach on Moving Offshore Platforms,” American Helicopter Society 74th Annual Forum, Phoenix, AR, May 2018.

[12] Tritschler, J. and Horn, J. F., “Objective Function Development for Optimized Path Guidance for Rotorcraft Shipboard Recovery,” AIAA Atmospheric Flight Mechanics Conference, Dallas, TX, June 2015.

[13] 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, Phoenix, AR, May 2018.

[14] Schwartz, A., “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.