2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Motion Primitives and 3-D Path Planning for Fast Flight through a Forest

Aditya A. Paranjape, Kevin C. Meier, Xichen Shi, Soon-Jo Chung, and Seth Hutchinson

Abstract— **This paper addresses the problem of motion planning for fast, agile flight through a dense obstacle field. A key contribution is the design of two families of motion primitives for aerial robots flying in dense obstacle fields, along with rules to stitch them together. The primitives are obtained by solving for the flight dynamics of the aerial robot, and explicitly account for limited agility using time delays. The first family of primitives consists of turning maneuvers to link any two points in space. The locations of the terminal points are used to obtain closed-form expressions for the control inputs required to fly between them, while accounting for the finite time required to switch between consecutive sets of control inputs. The second family consists of aggressive turn-around maneuvers wherein the time delay between the angle of attack and roll angle commands is used to optimize the maneuver for the spatial constraints. A 3-D motion planning algorithm based on these primitives is presented for aircraft flying through a dense forest.**

I. Introduction

Birds flying through dense forests represent a combination of an agile airframe and an adroit motion planner capable of ensuring collision free flight at high speeds in obstacle-rich environments. Figure 1, taken from a video recorded from a camera attached to a goshawk flying through a forest, illustrates the problem addressed in this paper: broadly, ensuring that an aerial robot can fly rapidly through a dense obstacle field such as a forest.

A. Literature Review

Broadly speaking, the challenges involved in flying an aircraft through dense, unstructured, uncharacterised obstacle fields can be grouped into the two categories: (1) localization and navigation in the absence of positioning aids such as GPS [4], [24], and (2) motion planning for obstacle-free flight which is also optimum in some sense such as minimum time or area coverage. The former problem has been addressed widely in robotics at large using vision [3], [13] and lidar [2].

Two methodologies have been employed in the literature for computing the desired trajectory and the control inputs for aerial robots (primarily quadrotors and helicopters). The first methodology seeks to decompose the motion of the robot into *motion primitives* which are stitched together and combined with a high level guidance algorithm [2], [5], [6], [7], [9].

The second approach to motion planning combines the dynamics formally with collision avoidance constraints, and the resulting problem is solved in a receding horizon or model predictive control (MPC) framework [15], [22], [23], or using potential field methods [21]. While MPC can guarantee optimality with respect to the chosen cost function, it can be computationally intensive and solving the complete optimization problem may be difficult within the time constraints posed by high speed flight in dense obstacle fields.

To the best of our knowledge, notwithstanding some demonstrations of fixed-wing-aircraft flying through obstacles [2], the robotics literature does not present any instances of motion planning for aircraft wherein the impact of the aerodynamics may have been formally analysed, or a knowledge of the flight dynamics employed for efficient motion planning outside of incorporation into optimization methods as dynamical constraints. Incidentally, motion planning methods which incorporate the aerodynamics have been presented in the literature for large aircraft and missiles [1], [16], [20]. where they are occasionally referred to as pure pursuit guidance laws.

Fig. 1. View from a camera attached to a goshawk flying through a forest. Credits: BBC.

B. Contributions

The objective of this paper is to present two families of motion planning primitives for aerial robots flying through dense obstacle fields, together with novel time delay-based rules to stitch them together. The stitching rules, for the first time to the best of the authors' knowledge, make use of aircraft agility explicitly in motion planning. The primitives themselves are continuously parametrized in the space of *constant control inputs* (unlike primitives defined in terms of constant values of state variables, such as the flight speed or the turn rate). The first family of primitives guides the aircraft between waypoints, such that each primitive represents a mapping between the coordinates of the successive waypoints and *constant* control inputs required to fly between

The authors are with the University of Illinois at Urbana-Champaign. Aditya Paranjape, Xichen Shi, and Soon-Jo Chung are with the Department of Aerospace Engineering. Kevin Meier and Seth Hutchinson are with the Department of Electrical and Computer Engineering. Email: {paranja2, kcmeier2, shi12, sjchung, seth}@illinois.edu. This research was funded by ONR N00014-11-1-0088 and NSF IIS-1253758

them. The second set of primitives represent an aggressive turn around (ATA) maneuver a robot uses to back-track from localized dense pockets of obstacles which may be encountered in the course of high speed flight. The ATA manevuer could thus help increase the speeds at which aircraft can fly safely through dense obstacle fields [10].

The primary contributions of this paper are as follows.

1) Analytical formulae, in the form of algebraic relations, are derived for control inputs required to accomplish a turn between two points in 3-D space, subject to performance limitations of the aerial robot. In particular, these analytical formulae yield *constant control inputs* corresponding to a circular trajectory connecting the initial point and the waypoint instead of "active" guidance between a chosen pair of waypoints. The computation of the control inputs makes it possible to use their values as part of the cost function for optimization (see Sec. V). Moreover, since the mapping from the waypoints of the control inputs is in the form of algebraic equations, it is computationally light and frees up computational resources for tasks such as sensing, mapping, etc.
2) A formal, analytical approach is presented to account for limited aircraft agility, which manifests in the form of a non-zero value of the time required to switch between control inputs, and forms the basis of the switching logic between primitives.
3) An aggressive turn-around (ATA) maneuver is used to help the motion planner deal with localized impenetrable pockets of obstacles (see Fig. 2). The ATA used in this paper, and first presented by the authors in [19], is an instantaneous 3-D turn with constant control inputs with the time delay between them acting as an additional design parameter. Fig. 3 shows some sample ATA maneuvers, each of which is suited to different types of obstacle fields. Unlike some ATA maneuvers presented in the literature (e.g., [14]), the ATA maneuvers in this sequel use constant control inputs whose values depend on the shape of the obstacle field and do so without ignoring the dynamics of the aerial robot.
4) A 3-D motion planning algorithm is designed based on the aforementioned two primitives, and its capabilities are demonstrated by simulation. The motion planner presented in this paper consists of a *greedy* path planner augmented by an ATA maneuver. The path planner uses a combination of model predictive control (MPC) approach and gradient descent. During each step of the MPC, the control inputs are held constant. Closed-form expressions for the aircraft trajectory during each interval are used to compute the distance from obstacles and identify feasible trajectories.

The paper is organized as follows. The equations of motion are derived in Sec. II. Control laws for non-aggressive maneuvering (also called *routine* flight) are derived in Sec. III, together with an analytical approach for accommodating the agility of the aerial robot. Aggressive turns are modeled in

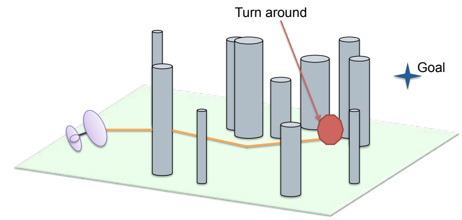


Fig. 2. Situation where an aggressive turn is mandated.

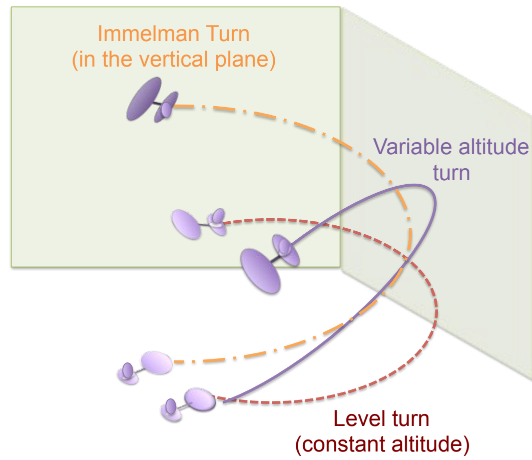


Fig. 3. Schematic of aggressive turn-around maneuvers

Sec. IV, and the motion planning algorithm is described in Sec. V. Simulation and experimental results are presented in Sec. VI, while the analysis performed in the aforementioned sections is used to assess high speed flight from a *first-principle* viewpoint in Sec. VII.

II. Equations of Motion

We will ignore the rotational dynamics of the MAV. This does not affect the motion planning principles put forth in this paper, and they can be controlled using a faster inner-loop controller [18], [19]. Commonly encountered symbols have been defined in Table II, following the notation used in flight mechanics. To simplify the notation, define

$$k = \frac{\rho S}{2m}, \quad T = \frac{\text{Thrust}}{m} \tag{1}$$

where ρ is the density of air, S is the area of the wing (a reference area), and m denotes the mass of the aircraft. The aircraft dynamics are then described by the following

TABLE I
Nomenclature

Symbol	Explanation
$C_L(\alpha)$, $C_D(\alpha)$	coefficients of lift and drag
T	thrust per unit mass
V	flight speed
α	angle of attack
γ	flight path angle
μ	wind axis roll angle
χ	aircraft heading

equations [11], [12]:

$$
\begin{aligned}
&\dot{x} = V\cos\gamma\cos\chi, \quad \dot{y} = V\cos\gamma\sin\chi, \quad \dot{h} = V\sin\gamma,\\
&\dot{V} = \left(T\cos\alpha - kV^2C_D(\alpha)\right) - g\sin\gamma,\\
&\dot{\gamma} = \left(\frac{T\sin\alpha}{V} + kVC_L(\alpha)\right)\cos\mu - \frac{g\cos\gamma}{V},\\
&\dot{\chi} = \left(\frac{T\sin\alpha}{V} + kVC_L(\alpha)\right)\frac{\sin\mu}{\cos\gamma}.
\end{aligned} \tag{2}
$$

where g is the gravitational constant and h denotes the altitude of the robot. We prescribe first-order dynamics for thrust T, angle of attack α and bank angle μ, with the understanding that a well-chosen first-order behavior can be achieved by using an inner-loop controller:

$$\dot{T} = a_T(T_c - T), \quad \dot{\alpha} = a_\alpha(\alpha_c - \alpha), \quad \dot{\mu} = a_\mu(\mu_c - \mu), \tag{3}$$

where $a_{\{\cdot\}}$'s denote the time constants, and the subscript c denotes values commanded by the motion planner. The motion planning problem involves choosing waypoints and mapping their choice to T_c, α_c, and μ_c. The design of inner-loop controllers for actuating the aforementioned control surfaces is relatively straight-forward [18], [19] and is not addressed in this paper.

Remark 1: The angle of attack α is defined as the angle made by the longitudinal axis of the aircraft with the projection of the velocity vector onto the plane of symmetry of the aircraft. The wind axis roll angle μ is the complement of the angle made by the lift vector with the *global* horizontal plane. □

III. Mapping End Points to Control Inputs: The Agility Connection

We derive an algebraic formula which maps the distance and the bearing of the desired waypoint to the control input required to reach it, such that the dynamics of the vehicle (2) are not ignored in the process. This is a unique feature of our algorithm. The waypoints are chosen inside a cone which is defined by placing the aircraft at its vertex, and by aligning the axis of symmetry of the cone with the instantaneous velocity vector (which makes an angle χ with the global heading reference and γ with the global horizontal plane). The "height" of the cone is bounded by the sensing radius. See Fig. 4 for an illustration.

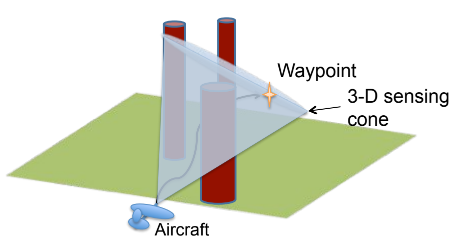


Fig. 4. A schematic showing a 3-D path between the initial point and a prescribed waypoint in the sensing cone.

We first make the notion of aircraft agility precise. We interpret agility as the ability to change accelerations rapidly, and therefore, define agility tentatively as the rate of change of acceleration for translational motion and rate of change of angular velocities for rotational motion [17]. For example, the turn rate (which is the rate of change of the velocity vector and hence an acceleration) is changed by rotating the lift vector about the longitudinal axis. Thus, the time required to rotate the lift vector through a prescribed angle is an important agility metric.

In this section, we will start with the assumption of unlimited agility (instantaneous rotation of the lift vector, Sec. III-A), and then use the results to analyse the case of finite agility (Sec. III-B).

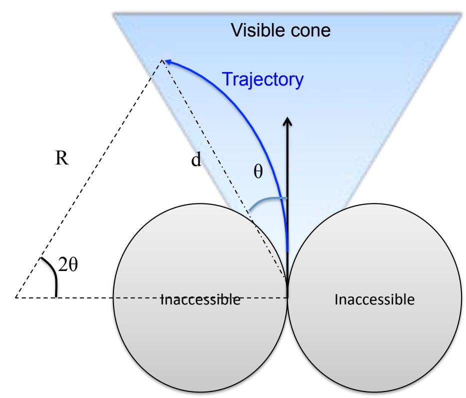


Fig. 5. Circular trajectory given an end point, and assuming infinite agility. This cone shown here is a 2-D projection of the 3-D cone in Fig. 4.

A. Unlimited Agility

Consider the dynamics of χ from (2), given by

$$\dot{\chi} = \left(\frac{T\sin\alpha}{V} + kVC_L\right)\frac{\sin\mu}{\cos\gamma}. \tag{4}$$

When the agility is infinite, it is possible to change $\dot{\chi}$ *instantaneously* between its limiting values, given by, $\max_\alpha$ (and $\min_\alpha$)$(T\sin\alpha/V + kVC_L(\alpha))/\cos\gamma$ (obtained from (2) by setting $\sin\mu = 1$), reflecting the ability to change μ and α instantaneously. We will assume that the range and the bearing to an object can be measured.

Consider Fig. 5 which shows the sensing cone at an arbitrary instant of time, with the aircraft at the vertex of the cone. Suppose that the aircraft needs to reach the point (d, θ) shown in Fig. 5, which is chosen as a candidate waypoint by the motion planning algorithm. We assume that the trajectory linking them is parametrized by a single set of constant control inputs (T, α, μ). From Fig. 5, we deduce that the turn radius is given by

$$R = \frac{d\cos\theta}{\sin 2\theta} = \frac{d}{2\sin\theta}, \tag{5}$$

Since the turn radius is also given by $R = V\cos\gamma/\dot{\chi}$, it follows from (4) and (5) that the commanded value of μ satisfies

$$\sin\mu_c = \frac{2\sin\theta\cos^2\gamma}{\left(kC_L + \frac{T\sin\alpha}{V^2}\right)d}. \tag{6}$$

If the value of μ_c is larger than the limiting value, it is possible to change γ to compensate for the deficiency in μ. In general, the flight path angle can be chosen to ensure that the aircraft reaches the waypoint at the required altitude. As an example, one may choose a proportional law

$$\gamma_c = k_\gamma(h_{\text{waypoint}} - h),$$

where $k_\gamma > 0$. The commanded values of the angle of attack and thrust can be chosen to ensure that

$$C_L(\alpha_c) = \frac{k_p V(\gamma_c - \gamma) + g\cos\gamma}{kV^2\cos\mu} - \frac{T\sin\alpha_c}{kV^2}, \quad (7)$$

$$T_c = kV_c^2 C_D(\alpha_c) + g\sin\gamma_c + k_T(V_c - V), \quad (8)$$

where $k_T > 0$ is a constant, and V_c is the commanded speed. Note that we have added proportional error terms to the equilibrium equations obtained from (2).

B. Finite Agility using Time Delay-Based Approach

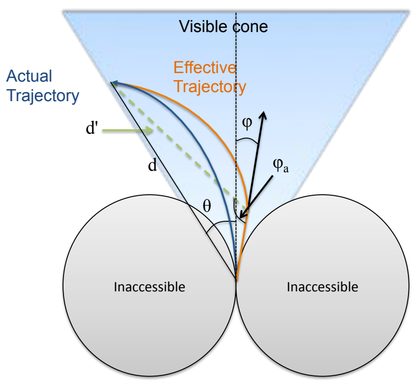


Fig. 6. Decomposing a trajectory into a delayed path and a circular trajectory to the desired end point when the agility is finite.

Finiteness in agility is a consequence of the fact that α and μ both require a finite amount of time to change values. A well-designed inner loop controller will ensure that the dynamics of α and μ behave like low-pass filters, as illustrated in (3).

A low pass filter of the form $\dfrac{1}{\tau s + 1}$ may be viewed as the first order Padé approximation of a time delay τ. Thus, we can model agility via a time delay in the system. This allows us to decompose the trajectory of the aircraft, as it switches from one control input to another and flies from the vertex of the cone in Fig. 6 to the waypoint located at a distance d and bearing θ from the vertex (labelled as the "actual trajectory" in Fig. 6), as the *sum* of two segments: (1) a drift with the initial control input μ_0 for time τ, and (2) a drift along the new roll angle μ_c for the remainder of the time. The two segments take the aircraft to (d, θ) in the same time as the actual trajectory, but do not coincide with the actual trajectory. They are labelled in Fig. 6 as the "effective trajectory." We seek to calculate μ_c given (d, θ).

We first note that the drift distance can be approximated by $V\tau$, and the aircraft may be assumed to turn through an angle $\dot{\chi}_0\tau$ during this time. As long as τ is small, the initial drift may be viewed as happening along a straight line.

After the initial drift is complete, the controls switch to the new configuration; in particular, $\mu_0 \mapsto \mu_c$. We can now use the formulation from Sec. III-A after replacing (d, θ) with the new distance and bearing angle.

The new distance, d', is given by

$$d'^2 = d^2 + (V\tau)^2 - 2dV\tau\cos(\theta + \phi), \quad \phi = \frac{\dot{\chi}_0\tau}{2}. \quad (9)$$

We calculate the angle ϕ_a (shown in Figure 6) using

$$\cos\phi_a = \frac{(d')^2 + (V\tau)^2 - d^2}{2d'V\tau} = \frac{V\tau - d\cos(\theta + \phi)}{d'}. \quad (10)$$

The new bearing is given by $\theta_{\text{n}} = 180 - \phi_a$, and we can use the formulation from the previous section with $(d, \theta) \leftarrow (d', \theta_n)$. For completeness, we note that the angle of attack and thrust commands are chosen as described in (7) and (8).

IV. Aggressive Turn Primitive

Aggressive turns are performed with the objective of reversing the aircraft heading, i.e., changing it by 180 deg, when collision-free forward flight is infeasible within the performance limitations of the aircraft. The word "aggressive" also suggests that these maneuvers take the aircraft to the boundary of its flight envelope, and they are unsustainable (and hence purely transient) in nature.

Mathematically, the design of an aggressive maneuver can be considered as an optimization problem:

$$\min_{T_c, \alpha_c, \mu_c} \int_{t_{\text{initial}}}^{t_{\text{final}}} (x^2 + y^2 + h^2)dx,$$
$$\text{subject to } |\chi(t_{\text{final}}) - \chi(t_{\text{initial}})| = \pi. \quad (11)$$

where the weights q_x, q_y, q_h depend on the shape of the volume available for turning. The optimization of an ATA maneuver for minimizing the turning volume in (11) has been addressed by the authors in [19]. We showed, in particular, that the optimal angle of attack command α_c is a constant and equal to the stall angle of attack α_{stall}. We also argued that the optimal control inputs can be approximated by constant commands for thrust and wind axis roll angle, together with the time delay between the commands of α_c (pull-up) and μ_c (roll), which is matched to the volume available for turning. The resulting ATA is described in Algorithm 1. Figure 7, from [19], shows the values of α and μ during an ATA maneuver, together with the three stages of the maneuver, as described in Algorithm 1. The maneuver starts with a pull-up to $\alpha_c = C_L^{-1}(C_{L,\max})$, followed by a roll to a prescribed value of μ. The parameter τ_d, which is the time delay between α_c and μ_c commands for the ATA, needs to be optimized.

Figure 8 depicts ATA trajectories for various values of τ_d, with $\mu_{\max} = 1.1$ rad. For example, $\tau_d \approx 1$ s can be used to turn around in a long but narrow corridor with a minimum turn radius of less than 0.5 m, while $\tau_d = 0$ can be used to turn when the turning volume has a sideways space of nearly 3 m but the permissible change in altitude is under 0.3 m. Interestingly, the latter case also minimizes the distance covered along the x axis (i.e., in the direction of the original flight path).

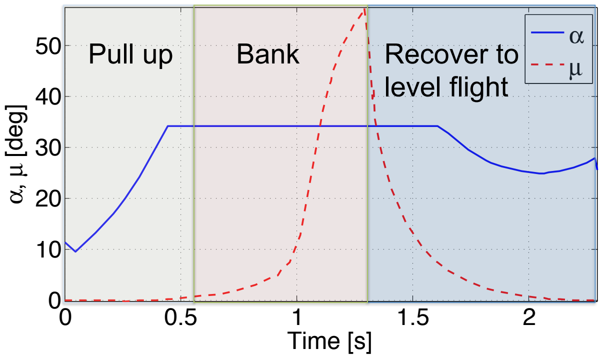


Fig. 7. The three stages of an optimum ATA maneuver [19].

Although the qualitative trends in Fig. 8 are independent of the initial conditions, a non-zero γ can improve the turning performance significantly. A lower initial speed reduces the forward distance covered during the turn, but it has virtually no bearing on the actual turn radius.

For the ATA maneuver in Algorithm 1, we need to estimate $\Delta\chi_{\text{crit}}$, the heading change after which the aircraft commences recovery to level flight. The value of $\Delta\chi_{\text{crit}}$ depends on the agility; for an aircraft with infinite agility, we would set $\Delta\chi_{\text{crit}} = 0$. For an aircraft with a finite agility, i.e, with $\dot{\mu} = a_\mu(\mu_c - \mu)$ and a_μ defined in (3), we calculate $\Delta\chi_{\text{crit}}$ by assuming that V, γ and α do not change significantly in the short time $1/a_\mu$. It was shown in [19] that $\Delta\chi_{\text{crit}}$ can be approximated by

$$\Delta\chi_{\text{crit}} \approx \frac{T\sin\alpha/V + kVC_L}{a_\mu}.$$

The thrust T_c required for the maneuver in Algorithm 1 can be approximated in terms of the speed at the start of the ATA maneuver, denoted by V_0, by assuming a zero change in altitude [19]:

$$T_c = \frac{kC_D(\alpha_{\max})V_0^2}{2}. \tag{12}$$

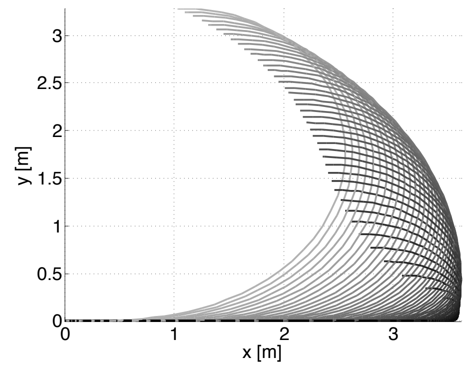
This value, however, needs to be used with caution because the derivation of (12) assumes that the aircraft recovers all of its kinetic energy at the end of the turn. This is usually not the case, and consequently, the thrust requirement can be significantly smaller [19].

Algorithm 1 Agile turn-around (ATA) maneuver

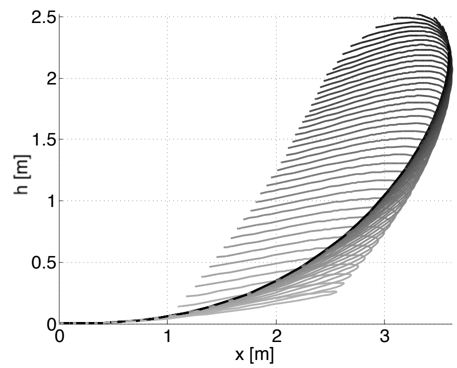
Result: $\chi \leftarrow \chi \pm \pi$
Initialize $t \leftarrow t_0$ and $\chi_f = \chi \pm \pi$
while $\chi \neq \chi_f$ **do**
  $\alpha_c = \alpha_{\text{stall}}$, T_c from (12)
  if $t > t_0 + \tau_d$ and $|\chi_f - \chi| < \Delta\chi_{\text{crit}}$ **then**
    $\mu_c \leftarrow \mu_{c,\max}$
  else
    $\mu_c = 0$
  end
end

V. Motion Planning Algorithm

The primary objective of this section is to show how the motion primitives derived in Sec. III and IV can be used in a motion planning algorithm for high speed flight in a densely crowded environment. The motion planning algorithm derived in this section combines the aforementioned motion primitives with a model predictive control (MPC) approach.



(a) $x - y$ projection



(b) $x - h$ projection

Fig. 8. Plots showing the aggressive turn trajectory for $\tau_d \in [0, 1]$ (dark curves denote a larger time delay) in the $x - y$ and $x - h$ planes.

Formally, let $\mathcal{V}$ denote the visible region. We choose points $\xi_i = (x_i, y_i, h_i) \in \mathcal{V}$ randomly, and the index i satisfies $1 \leq i \leq N$ for a suitably large sample size N. Let $\Sigma = \{\sigma_i(t)\}$ denote the set of trajectories $\sigma_i(t)$ which connect the starting point to the waypoint ξ_i. Note that the mapping $\xi_i \mapsto \sigma_i(t)$ is obtained from the analytical approximations described in Sec. III, which, in fact, yield the map $\xi_i \mapsto \mathbf{u}_{c,i}$, the vector of constant control inputs.

Let $d_{\text{goal}}(\xi_i)$ denote the distance to the goal from the waypoint ξ_i and let $\mathcal{T} = \{\mathcal{T}_j\}$ denote the set of obstacles (each of which carries a unique index j). Let us denote the distance of an obstacle from a trajectory by $d(\sigma_i(t), \mathcal{T}_j)$. One way to perform obstacle avoidance is to design navigation functions using this distance. Alternately, one may simply reject trajectories for which $\min_j \{d(\sigma_i(t), \mathcal{T}_j)\}$ is less than a certain threshold. In order to prevent overly conservative thresholds, one may allow the threshold to depend on the dynamics and a stochastic model of the disturbances [8].

The motion planner runs at the end of a pre-defined

Algorithm 2 Motion planner for agile flight

Result: Safe, fast flight through a forest
initialization: fly = 1
while $d_{\mathrm{goal}} > d_{\mathrm{nom}}$ **do**
  Draw n samples $\mathcal{V}_n := (x_i,\, y_i,\, h_i) \in \mathcal{V},\ 1 \le i \le n$
  Define $\mathrm{out} = \mathrm{zeros}(n, 5)$ (cost, feasibility, control)
  for every $\xi_i \in \mathcal{V}_n$ **do**
    Compute trajectory $\sigma_i(t)$ and control inputs $\mathbf{u}_{c,i}$
    Compute $d(\sigma_i(t),\ \mathcal{T}_j)\ \forall\, \mathcal{T}_j \in \mathcal{V}$
    if $\exists\ i$ s.t $\min(d(\sigma_i(t),\ \mathcal{T}_j)) > \mathrm{threshold}$ **then**
      $\mathrm{out}(i, :) = [J(\xi_i,\, \sigma_i(t)),\ 1,\ \mathbf{u}_{c,i}]$
    end
  end
  if $\max(\mathrm{out}(:, 2)) = 0$ **then**
    fly $= 0$
  else
    find $j = \arg\min_i \{\mathrm{out}(i, 1) | \mathrm{out}(i, 2) = 1\}$
    $\mathbf{u}_c \leftarrow \mathbf{u}_{c,j} = \mathrm{out}(j,\ 3{:}5)$
    set time of flight $t_{\mathrm{flight}} = \sigma_j^{-1}(\xi_j)$
  end
  if fly $= 1$ **then**
    Fly "routinely" with control $\mathbf{u}_c$ for t_{flight}
    update d_{goal}
  else
    Perform aggressive turn using Algorithm 1
    set fly $= 1$
    update d_{goal}
  end
end

interval, or after an aggressive turn from Sec. IV. It searches the visible region for possible locations where the aircraft could be directed. The choice of this point is obtained by minimizing a prescribed cost function, i.e., by computing

$$\arg\min_i \left\{ J_i \mid \min_j \{d(\sigma_i(t),\ \mathcal{T}_j)\} > \mathrm{threshold} \right\}. \qquad (13)$$

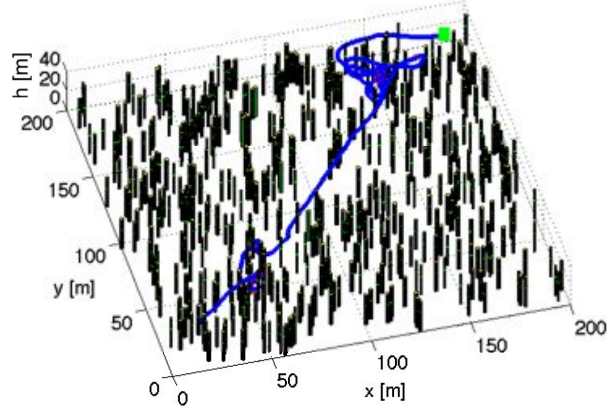
The cost function and the threshold can be designed on a case-by-case basis, as we illustrate in the following section.

VI. Simulation and Experimental Results

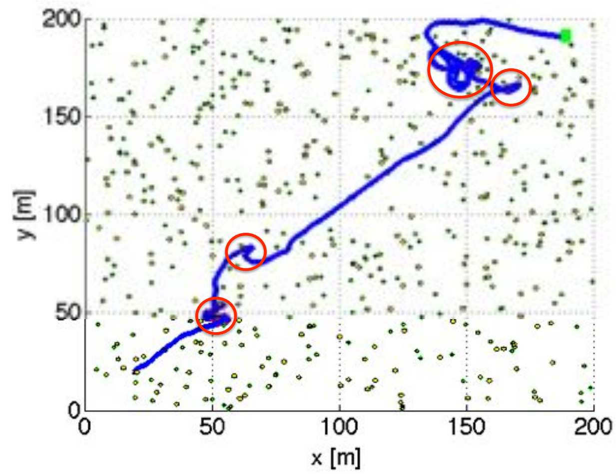
A. Setup

We demonstrate the capabilities of the motion planner described in Algorithm 2 and the ATA maneuver through simulations performed in Matlab. The aircraft model used for simulations has the following parameters: mass $m = 100$ g, wing area $S = 0.5\ \mathrm{m}^2$, coefficient of lift $C_L = 0.3 + 2.5\,\alpha$, and coefficient of drag $C_D = 0.03 + 0.3\,C_L^2$. The maximum values of thrust and wind axis roll angle are given by $T_{\max} = 1.2$ and $\mu_{\max} = \pm 1.1$ rad. The limiting angle of attack (used as a proxy for the stall angle of attack) is assumed to be $\alpha_{\max} = 35$ deg. A forest with a specified number of trees is generated such that the coordinates of the trees and their radii are chosen through (mutually independent) Poisson processes, and the tree radii are constrained between 0.5 and 1 m.

Since $d_{\mathrm{goal}}(\xi_i)$, as a cost function, offers a very poor resolution at large distances from the goal, we employ a cost function which penalises the commanded path relative to a straight line between the given position and the goal. Let ξ_0 denote the position of the aerial robot at which sampling is performed, and let $\theta_{\mathrm{goal}}(\xi_0,\, \xi_i)$ denote the angle between the segment $\xi_i - \xi_0$ and $\xi_0 - \mathrm{goal}$, i.e., $\theta_{\mathrm{goal}}(\xi_0,\, \xi_i)$ measures the bearing to the waypoint in relation to the bearing to the



(a) 3-D trajectory



(b) x-y projection of the 3-D trajectory

Fig. 9. Trajectory of the aerial robot in 3-D space as it navigates a forest, and its projection on the x-y plane. Red circles in the $x - y$ projection denote the locations of the ATA maneuvers.

goal. Then, the cost of ξ_i is defined by

$$J_i(\xi_i; \xi_0) = (1 - \cos(\theta_{\mathrm{goal}}(\xi_0,\, \xi_i)))\,.$$

The stopping condition is set to $d_{\mathrm{goal}}(\xi_i) < 20$ m, while the threshold distance in (13) is set to 2 m. The start point and the goal are at $(20,\, 20)$ and $(190,\, 190)$ respectively.

B. Simulations

Figure 9 shows the simulation of an aircraft flying through a $200 \times 200\ m^2$ forest with 500 trees distributed randomly with a uniform distribution. The commanded speed is set to 9 m/s. Figure 10 shows the zoomed in view of an area where a series of ATA maneuvers is employed to navigate a particularly dense patch. The results show that the motion planning algorithm (Algorithm 2) successfully guides the aircraft through the forest. Interestingly, the ATA maneuver was required even at a flight speed of 6 m/s (a different case

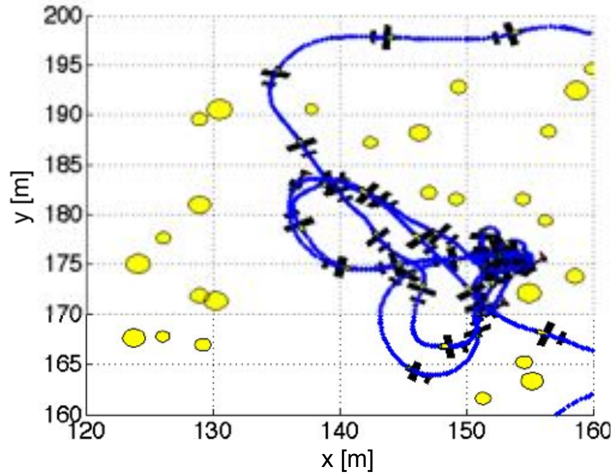


Fig. 10. Magnified view from Fig. 9 showing the trajectory of the aerial robot as it performs a series of ATA maneuvers in a dense patch of trees in the forest.

from the example shown in Fig. 9), which demonstrates its importance during flight in obstacle-rich environments.

C. Effect of Control Laws on ATA

Fig. 11. An off-the-shelf MAV called MiniVapor. It has a rudder for yaw control, but no direct roll control mechanism.

In order to examine the effect of including an inner loop controller on the performance of an ATA, we consider the case of an aircraft, such as the one shown in Fig. 11, which lacks a roll control device and has a rudder for yaw control. A two-step controller is designed, as explained in [19], which maps the wind axis roll angle onto a roll rate command which, in turn, commands a rudder deflection

$$\begin{aligned} p_c &= k_{p,\mu}(\mu_c - \mu) + k_{I,\mu}\int_0^t (\mu_c - \mu)\,dt, \\ \delta_r &= k_{p,p}(p_c - p) + k_{I,p}\int_0^t (p_c - p)\,dt. \qquad (14) \end{aligned}$$

This control law was tested experimentally on the aircraft in Fig. 11. The flight testing was conducted indoors, and the aircraft was tracked using the Vicon motion capture system.

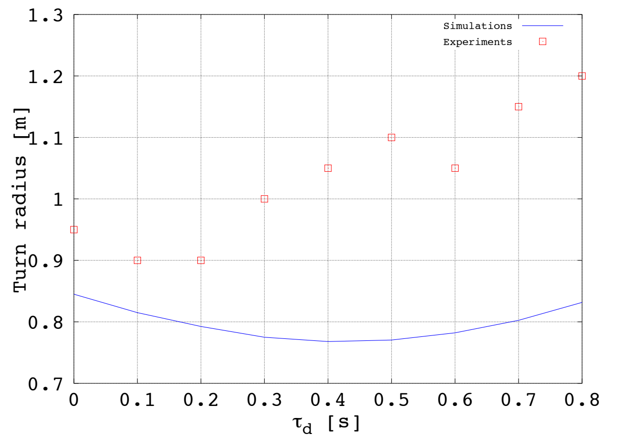


Fig. 12. Simulation results and experimental data showing the effect of a control law [19] on the turn diameter during an ATA.

Figure 12 shows the turn radius as a function of the time delay τ_d between the pull-up and roll. Simulation results are shown alongside the experimental data. The simulations show that, unlike the conclusions of Fig. 8, the turn radius is optimized for a time-delay of around 0.4 s. The variation in turn diameters is about 0.1 m (10 cm) for the complete range of τ_d considered here, which also contrasts sharply with Fig. 8. The experimental results yield an optimum time-delay of $0.1 - 0.2$ s, although the turn diameter does not increase uniformly with τ_d. Note that the discrepancy in the numbers between the simulation and experiments is largely due to different initial conditions in the two exercises. A more serious concern is the discrepancy in the variation of the turn diameter with τ_d, the reasons for which are not clear and represent an open problem.

VII. Discussion

Although flying an aircraft does present ample challenges when compared with a terrestrial robot, there are some critical differences between the two that actually make it *easier* to maneuver an aircraft than to drive a car at high speeds.

First, Eqs. (2) and (5) yield the following expression for the turning radius R after ignoring the contribution from T: $R = \frac{\cos^2\gamma}{kC_L \sin\mu}$. This is also the lower bound on the turn radius for given C_L and γ, because a non-zero value of thrust would only reduce the radius further. The expression for R is independent of V: therefore, the turn radius, which is a measure of how crowded an obstacle field can be flown through, is *independent of the flight speed*. The assertion that R is independent of V runs counter to intuition that the turn radius is proportion to the speed. Another way to appreciate this conclusion is to note that the centripetal acceleration, given by V^2/R equals the radial component of the lift per unit mass, the latter term also proportional to V^2. Thus, R is independent of V. Note, however, that the lower limit on the aircraft speed is dictated by k and α_{stall}, while and the structural strength of the airframe constrains the upper limit on the flight speed.

Second, a key metric which constrains the class of navigable obstacle fields is a_μ, which measures how rapidly an aircraft can change its turn rate. It turns out that $a_\mu \propto V^2$, i.e., the aircraft agility *increases* with its speed. These two observations indicate that high speed flight is, strictly speaking, a much better alternative to low speed flight from the point of view of performance.

Finally, we note that the turn radius is inversely proportional to $k = \frac{\rho S}{2m}$, which is clearly an important design parameter. A large value of k yields a tighter ATA maneuver, and the volume inside which an ATA maneuver can be performed increases rapidly with reducing k. However, a larger value of k is ideally suitable for *slow* flight. Therefore, k needs to be optimized for the class of obstacle fields that the aerial robot is designed to cross as well as the desired time of crossing.

VIII. Conclusions

This paper presented a novel approach for constructing and stitching together motion primitives for aircraft maneuvering in dense obstacle fields. A unique feature of this approach is the explicit accounting of the agility of the aerial robot, made possible by a combination of time delay-based modeling of agility and the derivation of algebraic formulae for the motion primitives. This aids the process of stitching together the motion primitives by ensuring compatibility of successive primitives at their boundary. An aggressive turn-around (ATA) maneuver was also designed in this paper to allow aircraft to back-track from impenetrable regions of the obstacle field, while being able to maintain a high flight speed otherwise.

Acknowledgement

The authors gratefully acknowledge the contributions of Sunil Patel, undergraduate student in the Department of Aerospace Engineering, to the experiments reported in this paper.

References

[1] K. Berg-Taylor, K. Seo, and S.-J. Chung, "Development of a car-like online navigation testbed," in *IEEE International Conference on Electro/Information Technology (EIT), Ames, IA*, 2008, pp. 337 – 342.

[2] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[3] K. Celik, S.-J. Chung, M. Clausman, and A. K. Somani, "Monocular vision SLAM for indoor aerial vehicles," in *International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1566 – 1573.

[4] A. Dani, G. Panahandeh, S.-J. Chung, and Seth, "Image moments for higher-level feature based navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan*, 2013.

[5] C. Dever, B. Mettler, E. Feron, J. Popovic, and M. McConley, "Nonlinear trajectory generation for autonomous vehicles via parametrized maneuver classes," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 289 – 302, 2006.

[6] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 1, pp. 116 – 129, 2002.

[7] ——, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077 – 1091, 2005.

[8] J. Hu, J. Lygeros, M. Prandini, and S. Sastry, "Aircraft conflict prediction and resolution using Brownian motion," in *Proc. of the 38th Conference on Decision and Control, Phoenix, AZ*, 1999.

[9] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient two-phase 3D motion planning for small fixed-wing UAVs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 1035 – 1041.

[10] S. Karaman and E. Frazzoli, "High-speed flight in an ergodic forest," in *IEEE Internation Conference on Robotics and Automation*, 2012.

[11] H. J. Kelley, "Reduced-order modeling in aircraft mission analysis," *AIAA Journal*, vol. 9, no. 2, pp. 349 – 350, 1971.

[12] H. J. Kelley and T. N. Edelbaum, "Energy climbs, energy turns and asymptotic expansions," *Journal of Aircraft*, vol. 7, no. 1, pp. 93 – 95, 1970.

[13] J. Langelaan and S. Rock, "Towards autonomous UAV flight in forests," in *AIAA Guidance, Navigation and Control Conference, San Francisco, CA*, 2005.

[14] T. Matsumoto, A. Konno, R. Suzuki, A. Oosedo, K. Go, and M. Uchiyama, "Agile turnaround using post-stall maneuvers for tail-sitter VTOL UAVs," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan*, 2010, pp. 1612 – 1617.

[15] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Decentralized model predictive control of swarms of spacecraft using sequential convex programming," in *Proc. 23rd AAS/AIAA Space Flight Mechanics Meeting, Kauai, Hawaii*, 2013.

[16] A. Ollero and G. Heredia, "Stability analysis of mobile robot path tracking," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Pittsburgh, PA*, 1995, pp. 461 – 466.

[17] A. A. Paranjape and N. Ananthkrishnan, "Combat aircraft agility metrics - a review," *Journal of Aerospace Sciences and Technologies*, vol. 58, no. 2, pp. 1 – 16, 2006.

[18] A. A. Paranjape, S.-J. Chung, and J. Kim, "Novel dihedral-based control of flapping-wing aircraft with application to perching," *IEEE Transactions on Robotics*, 2013, to appear.

[19] A. A. Paranjape, K. C. Meier, S.-J. Chung, and S. Hutchinson, "Optimum spatially constrained turns for agile micro aerial vehicles," in *AIAA Guidance, Navigation and Control Conference, Boston, MA*, 2013, to appear.

[20] S. Park, J. Deyst, and J. How, "Performance and lyapunov stability of a nonlinear path-following guidance method," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 6, pp. 1718 – 1728, 2007.

[21] S. Scherer, S. Singh, L. Chamberlain, and M. Elgersma, "Flying fast and low among obstacles: Methodology and experiments," *International Journal of Robotics Research*, vol. 27, no. 5, pp. 549 – 574, 2008.

[22] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proc. American Control Conference, Boston, MA*, 2004, pp. 5576 – 5581.

[23] D. H. Shim, H. J. Kim, and S. Sastry, "Decentralized nonlinear model predictive control of multiple flying robots," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, 2003, pp. 3621 – 3626.

[24] J. Yang, A. Dani, S.-J. Chung, and S. Hutchinson, "Inertial-aided vision-based localization and mapping in a riverine environment with reflection measurements," in *AIAA Guidance, Navigation, and Control Conference, Boston, MA*, 2013.