

# Maneuver-Based Autonomous Navigation of a Small Fixed-Wing UAV

Myung Hwangbo and Takeo Kanade

**Abstract**—An urban operation of unmanned aerial vehicles (UAVs) demands a high level of autonomy for tasks presented in a cluttered environment. While fixed-wing UAVs have been well suited for long-endurance missions at a high altitude, their navigation inside an urban area brings more challenges in motion planning and control. The inability to hover and low agility in motion cause more difficulties on planning a feasible path in a compact region, and a limited payload allows only low-grade sensors for state estimation and control.

We address the problem of achieving autonomous navigation of a small fixed-wing using low-cost sensors such as GPS, IMU and camera. Based on a set of primitive maneuvers integrated with flight controllers, we present a motion planning method that produces dynamically feasible plans in a low dimensional search space. To deal with motion uncertainty due to wind and imperfect control, Markov Decision Process is formulated to compute the most probable path to a target. A minimum target observation time is also imposed on the motion plan to consider a camera's limited field-of-view.

In experiments, we demonstrate that a 1m wing-span airplane achieves an "air slalom" task in which the UAV must search, localize and pass through multiple gates autonomously.

## I. INTRODUCTION

Recently unmanned aerial vehicles (UAVs) has been expanding their operations in both military and civilian applications. One of common maneuvers demanded for a UAV in an urban environment is to approach a target or region of interest in a specified direction. This kind of maneuver is necessary, for example, when the UAV is in the middle of entering a gate or passageway between buildings, following a target moving on a road network, or landing on a runway or driveway.

This fact leads us to design a benchmark mission in what we call an *air-slalom task* [1]. The slalom task illustrated in Fig.1 is focused on the evaluation and demonstration of vision-based autonomous navigation algorithms for unmanned aerial vehicles by presenting a challenging course associated with targets. A UAV needs to complete tight turns through a slalom course consisting of multiple gates. In order to precisely enter multiple gates, it is essential to integrate advanced realtime motion planning as well as reliable target and vehicle state estimation into the navigation system.

Small fixed wings have unique benefits such as readiness to fly and glide, low maintenance cost and a wide working area. However, its motion planning and control suffer from a non-stoppable and minimum forward-velocity motion and

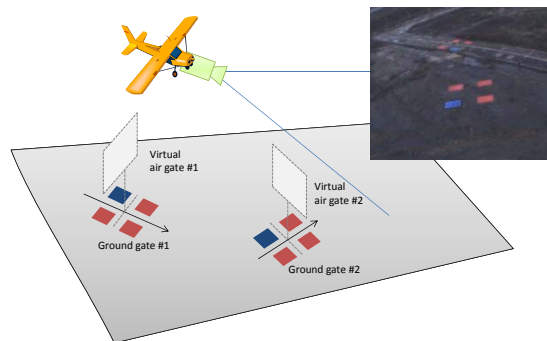


Fig. 1. An air-slalom task is designed as a benchmark test of an autonomous vision-based navigation system of a small fixed-wing UAV in an urban environment. The experimental setup consists of virtual gates in the air, each of which is conceivable from a set of colored square objects on the ground.

low-cost sensors. The change of an approaching direction requires not only space and time to make a turn but also a deliberate motion plan to meet the new direction. Even when a feasible pre-planned path is given, the precision and performance achievable in path tracking are limited by wind disturbances and low capabilities of sensing and control. Trajectory tracking, which demands a vehicle to be in a particular location at a specified time, can be significantly difficult since fixed-wings' dynamic characteristics are subject to a under-actuated and non-holonomic system.

In this paper, we will present an autonomous navigation scheme that combines motion planning and control using primitive maneuvers. A set of motion primitives are sampled from vehicle behaviors which are generated by simple roll and pitch controllers in an autopilot. Since these primitives are designed to start and end at a finite number of trim states, their interconnection inherently produces a feasible motion and the best path to a goal can be found in a low dimensional search space. Note that no particular path following or tracking is involved in our scheme. At the end of the execution of each maneuver, instead a new optimal plan to the goal will be obtained from a precomputed table.

The motion uncertainty, due to imperfect flight control and external disturbances such as wind, will be explicitly considered by a probabilistic motion model using a Markov Decision Process. We will describe how to find an optimal motion sequence that maximizes the probability that the vehicle reaches the target at a desired heading angle. Experiment results with a 1m wing-span airplane will show the effectiveness of our navigation system for the air-slalom task.

## II. RELATED WORK

The problem of UAV motion planning is especially challenging due to several complexities that have not been addressed by traditional planning strategies. Differential constraints become more important to find a feasible trajectory. Limited sensor capabilities increase uncertainty of not only vehicle and target states but also the knowledge about the environment. Previous research in robotics has explored a vision-based navigation [2], basic obstacle avoidance during flight [3] and 3D local trajectory planner with a predefined global path [4].

Many planning algorithms to produce 2D or 3D trajectories for autonomous aircraft guidance in known or unknown environments are based on a decomposition approach; first solve a dynamics-unconstrained planning problem, and then smooth the found path to a feasible trajectory using differential constraints. Traditional and state-of-the-art planning algorithms in textbooks [5], [6], [7], [8] and review papers [9], [10], [11], [12] may be used in this approach.

Our approach to planning with differential constraints is closely related with a finite-state motion model called a *maneuver automation* (MA) [13]. Trim states of an air vehicle has been effectively used for motion planning of a complicated nonlinear and high-dimensional dynamic system [1], [14], [15], [16], [13], [17], [18]. The concept of MA is based in part on the fact that human pilots can accomplish nimble control using a mixture of trim trajectories and maneuvers. The general idea of finite state models is to search or optimize a path connected in a discretized finite-dimensional space instead of an infinite one. This significantly reduces the computational complexity of a motion planning problem under differential constraints.

## III. PRIMITIVE MANEUVERS

Let a nonlinear differential system  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  be the dynamics of a fixed-wing airplane. The vehicle state  $\mathbf{x}$  is a high dimensional state vector and the control inputs  $\mathbf{u}$  are a thrust and surface deflections. Using a feedback controller  $\mathbf{u}^*$  during time  $k$  to  $k+1$ , this complex dynamic system can be reduced to a state machine  $\mathbf{f}_s$  that makes the transition from a trim state  $\mathbf{s}_k$  to another trim state  $\mathbf{s}_{k+1}$ .

$$\mathbf{s}_{k+1} = \mathbf{f}_s(\mathbf{s}_k, \mathbf{u}^*(k, k+1)) \quad (1)$$

Since the dimension of a trim state  $\mathbf{s}$  is much lower than that of the original state  $\mathbf{x}$ , the motion planning on a trim-state space is more advantageous to run in realtime when a set of motion primitives are built in such a way that they interconnect discretized trim states.

### A. Trim states

In aerodynamics literature [19], a fixed-wing's *trim* state is referred to as one of the following three motions: (1) level flight, (2) constant-altitude turn, and (3) turn with a constant climb rate, as illustrated in Fig. 2(a). Though an equilibrium state  $\dot{\mathbf{x}} = \mathbf{0}$  is generally called trim, trim states of a fixed

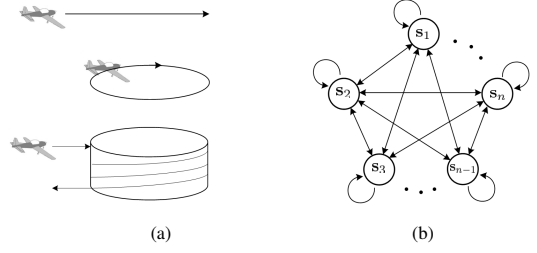


Fig. 2. Trim states of a fixed-wing airplane: (a) three motions in a trim state: level flight, constant altitude turning, and turning with a constant climb rate. (b) a finite state machine (FSM) defined by trim states  $\mathbf{s} = (\phi, \theta, V_a)$ .

wing also include cases where some vehicle states are not constant. In turning with a constant climb rate, the altitude and the yaw angle change at constant rates ( $\dot{z} = V_a \sin \gamma$  and  $\dot{\psi} = V_a/R$ ) when a climbing angle  $\gamma$ , turning radius  $R$  and airspeed  $V_a$  are given.

In the trim conditions above, the position  $(x, y, z)$  and the heading  $\psi$  in an inertial frame are irrelevant to the description of the motions and the remaining states are constrained by the dynamics  $\mathbf{f}$  and three parameters  $(\gamma, R, V_a)$ . Since  $\gamma$  is related to the pitch  $\theta$  by an attack angle  $\alpha$  and the roll  $\phi = V_a/R$  at the constant speed  $V_a$ , all of the trim conditions can be described by the following *trim state*  $\mathbf{s}$ :

$$\mathbf{s} = (\phi, \theta, V_a) \quad (2)$$

The control input  $\mathbf{u}^*$  in (1) is designed to drive the airplane to a given trim state. Suppose that there exist three controls (e.g., thrust, rudder and elevator) which are generally available in a small model airplane. Given a reference profile for desired vehicle states  $(\phi_d, \theta_d, V_{a,d})$ , the following PID (proportional-integral-derivative) controllers will minimize the error between measured and desired states. Each controller is independently dedicated to control the corresponding trim variable and implemented in autopilot:

$$u_{thr}(t) = K_{p,t}V_{a,err} + K_{i,t} \int V_{a,err} dt \quad (3)$$

$$u_{rud}(t) = K_{p,r}\phi_{err} + K_{d,r}\dot{\phi}_{err} + K_{i,r} \int \phi_{err} dt \quad (4)$$

$$u_{ele}(t) = K_{p,e}\theta_{err} + K_{d,e}\dot{\theta}_{err} + K_{i,e} \int \theta_{err} dt + \theta_{level} \quad (5)$$

where  $\theta_{level}$  is an angle of attack that holds an altitude at the air speed  $V_a$ . Vehicle behaviors are then governed by the closed-loop dynamic system  $\mathbf{f}_d$ , in which the past desired trim state  $\mathbf{s}_k$  advances to a new desired state  $\mathbf{s}_{k+1}$  by the feedback control  $\mathbf{u}^*$  during time  $T_s$ .

### B. Motion primitive selection

There exist infinitely many motion primitives, which of each makes a transition between two points in a continuous space of trim states. To select a limited number of transitions in building motion primitives, we confine the trim states to a one-dimensional space  $\mathbf{s}_i^* = \mathbf{s}(\phi_i, \theta^*, V_a^*)$  where  $\theta^*$  and  $V_a^*$  are predefined constants. The roll angle  $\phi_i$  is discretized and its change to another trim state creates a *lateral* motion

primitive. While the pitch angle  $\theta$  is always  $\theta_{level}$  at both ends of a motion (*i.e.*,  $\theta^* = \theta_{level}$  at trim), it is not necessarily constant during a transition so that it can create a *longitudinal* motion primitive.

The desired roll and pitch angles ( $\phi_d$  and  $\theta_d$ ) in (5) are given by a ramp and trapezoidal motion profile, respectively, during a transition between the discretized trim states  $s^*$ . For the pitch, the peak of  $\theta_d(t)$  determines the amount of altitude change. Each profile provides a sufficient settling time so that the angle error and the angular velocity becomes stationary under small ripples.

The lateral motion primitives in Fig.3 is generated using seven discrete roll angles. A constant cruise speed  $V_a$  is maintained in flight and each motion is driven by a change in the roll angle, which is discretized between -30 to 30 degrees in a 10-degree resolution. The motion length is proportional to the amount of the corresponding roll angle change. Relationships between the motion primitives can be described by a deterministic finite state machine (FSM) between the trim states in Fig.2(b). Each motion primitive is equivalent to a transition function from one state  $s_i^*$  to another  $s_k^*$  in the FSM.

Note that in this paper longitudinal motion primitives are not explicitly considered for the 2D motion planning in the following section (*i.e.*,  $\theta_d(t) = 0$ ) but separately used when altitude recovery is needed in experiments.

#### IV. MDP-BASED MOTION PLANNING UNDER MOTION UNCERTAINTY

When a motion primitive is executed in practice, a real trajectory resulted from the autopilot controllers is not always the same as the trajectory in Fig.3. Due to imperfect flight control and external disturbances such as wind, each motion primitive varies and is subject to some uncertainty as shown in Fig.4. The goal in this section is to find an optimal motion sequence that maximizes the probability that the vehicle will reach the target at a desired heading angle.

For a goal region located at the origin of a workspace, we formulate the planning problem as a Markov Decision Process (MDP) based on the following three factors; an uncertainty motion model using probability distributions, efficient discretization of the state space, and infinite horizon Dynamic Programming (DP) that computes a sequence of motion transitions that maximizes the probability of success. Since the DP returns an optimal action for every vehicle state in a discretized workspace, the best motion requested at any moments during navigation will be examined in a DP look-up table.

##### A. Probabilistic model of a motion primitive

Let  $\mathbf{q} = (x, y, \psi, \phi)$  be the vehicle state on a 2D horizontal plane. Since only lateral vehicle motions are considered here, the trim state of a fixed-wing UAV is fully characterized by the position  $(x, y)$ , heading angle  $\psi$  and roll angle  $\phi$ . A control input  $u$  that causes a transition between states corresponds to the change of a desired roll angle.

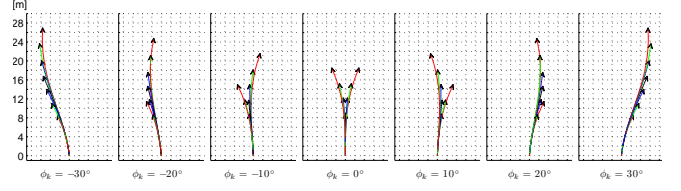


Fig. 3. A set of lateral motion primitives used in the MDP-based motion planning: each motion primitive makes a state transition from  $\mathbf{q}_k = (x, y, \psi, \phi)$  to  $\mathbf{q}_{k+1}$  using the control input  $u$  in the feedback control dynamic system  $\mathbf{q}_{k+1} = \mathbf{f}_d(\mathbf{q}_k, u)$ . At each column, the motions start with the same  $\phi_k$  and reach different final  $\phi_{k+1}$  ( $\phi_k$  and  $\phi_{k+1}$  are one of -30, -20, -10, 0, 10, 20 and 30 degrees). The motion length is proportional to the amount of roll angle change, *i.e.*,  $t = 0.3|\phi_{k+1} - \phi_k| + 0.6$  sec at  $V_a = 10.5$  m/sec.

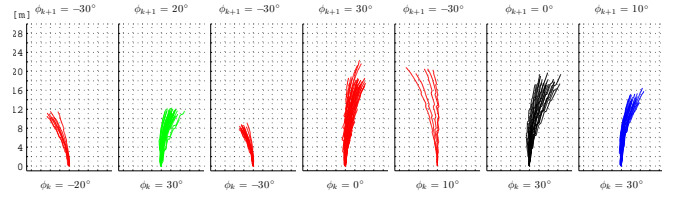


Fig. 4. Uncertainty of a vehicle motion in the execution of lateral motion primitives in the real world. At each column, the trajectories of a motion primitive are collected from real flight experiments when the same control input  $u$  is applied to change from  $\phi_k$  to  $\phi_{k+1}$ . The state transition from  $\mathbf{q}_k$  to  $\mathbf{q}_{k+1}$  corresponding to both ends of each trajectory is not deterministic due to control error and disturbance.

Due to an unpredictable error in the feedback control and imprecise vehicle state estimation, the next state  $\mathbf{q}_{k+1}$  driven by a control input  $u$  from a current state  $\mathbf{q}_k$  is *not* deterministic. Additional uncertainty in the state transitions may be caused by external disturbances such as wind gusts and atmospheric turbulence. One can see that actual paths would not always exactly trace the simulated trajectories in Fig.3. Lateral motion primitives in Fig.4, executed and collected in real flight experiments using the same control input, reveals non-deterministic responses of state transitions. Since transient and residual errors in the roll controller (4) make the most significant impact on the state transition, we model the uncertainty of a vehicle motion using a probability distribution of  $u$  as follows:

$$\mathbf{q}_{k+1} = \mathbf{f}_d(\mathbf{q}_k, u), \quad u \sim \mathcal{N}(\phi_{k+1} - \phi_k, \sigma_\phi) \quad (6)$$

where  $\mathbf{f}_d$  is a discretized dynamic system of the fixed-wing airplane and  $\sigma_\phi = \rho|\phi_{k+1} - \phi_k|$  reflects that the greater motion uncertainty occurs as the larger control input is given. From the feedback control dynamic system  $\mathbf{f}_d$  in (1), the control input  $u$  is modeled as normally distributed with the mean of a roll angle change,  $\Delta\phi = \phi_{k+1} - \phi_k$ , and the standard deviation  $\sigma_\phi$ . Fig.5 illustrates a probabilistic distribution of  $\mathbf{q}_{k+1}$  when finite discrete samples of a normal distribution  $u$  are applied.

##### B. State space discretization

The discretization of a state space is based on a grid of points with an equal spacing  $D$  in  $x$  and  $y$  and angular bins

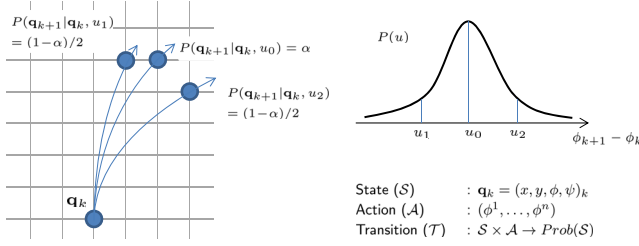


Fig. 5. Probabilistic model of a state transition in a Markov Decision Process by the corresponding lateral motion primitive in a discretized state space.

in  $\psi$  and  $\phi$ . For a rectangular workspace bounded by  $x_{max}$  and  $y_{max}$ ,  $N_s = x_{max}y_{max}/D^2$  position states are aligned at the origin. The heading angle  $\psi$  is discretized by dividing  $[-\pi, \pi]$  with  $N_\psi$  equally spaced bins where  $N_\psi$  is a multiple of 4 for purposes of symmetry. The roll angle  $\phi$  is discretized in the same way as done in the lateral motion primitives in Fig. 3, i.e., one of -30, -20, -10, 0, 10, 20 and 30 degrees and  $N_\phi = 7$ .

Using this discretization, a vehicle state  $\mathbf{q}$  can be approximated as a discrete state where the position and heading angle are rounded to a point closest to  $(x, y, \phi)$  in the corresponding grid space and the index of a roll angle changes by the control input. The total number of discrete states is  $N = N_s N_\psi N_\phi$ .

Since the value iteration for an optimization process in the MDP requires  $\mathcal{O}(bN)$  time and memory for each iteration where  $b$  is the number of discrete samples of the probabilistic state transition, the discretization resolution  $D$  and  $N_\psi$  should be taken into consideration for a reasonable computation time and memory. In our implementation, we use  $D = 2\text{m}$  and  $N_\psi = 120$  for a 3-degree resolution in  $\psi$ .

### C. Markov Decision Process

The goal of our MDP-based motion planning is to compute an optimal control  $u^*$  for every state  $\mathbf{q}$  in a workspace to maximize the probability of success  $P_s$  where  $P_s(\mathbf{q}) = 1$  when  $\mathbf{q}$  is inside a goal region  $\mathcal{G}$ . Given a control  $u$  for some state  $\mathbf{q}$ ,  $P_s$  depends on the response of the vehicle to the control and the probability of success in the subsequent state. The expected probability of success is described as  $P_s(\mathbf{q}) = E[P_s(v)|\mathbf{q}, u]$  over a random variable  $v$  for the next state. For  $N$  discrete states, the motion planning problem is equal to determining an optimal control  $u_i$  for each state  $i = 1, \dots, N$  using the discrete approximation and the expansion of the expected value to a summation as follows:

$$\begin{aligned} P_s(\mathbf{q}) &= \max_u E[P_s(v)|\mathbf{q}, u] \\ &= \max_{u_i} \sum_{j=1}^N P_{ij}(\mathbf{q}_j|\mathbf{q}_i, u_i) P_s(\mathbf{q}_j) \end{aligned} \quad (7)$$

where  $P_{ij}(\cdot)$  is the probability of entering state  $\mathbf{q}_j$  given a control  $u_i$  at current state  $\mathbf{q}_i$ .

A Markov Decision Process (MDP) is a stochastic process on the random variables of state  $\mathbf{q}$ , control  $u$  and reward function  $R$ . The control at each state is purely a function of the state without explicit dependence on time and past controls. Since the process is stationary, the expected probability (7) becomes a form of the following Bellman equation in the MDP [20]:

$$V^*(\mathbf{q}_i) = \max_{u_i} [R(\mathbf{q}_i, u_i, \mathbf{q}_j) + \sum_{j=1}^N P_{ij}(\mathbf{q}_j|\mathbf{q}_i, u_i) V^*(\mathbf{q}_j)] \quad (8)$$

where  $R(\mathbf{q}_i, u_i, \mathbf{q}_j)$  is a reward for the state transition from  $\mathbf{q}_i$  to  $\mathbf{q}_j$  using  $u_i$ . We set the reward function  $R$  as follows:

$$R(\mathbf{q}_i, u_i, \mathbf{q}_j) = \begin{cases} 1 & : \mathbf{q}_i \text{ in goal} \\ 0 & : \mathbf{q}_i \text{ out of workspace} \\ -0.001 - \alpha_1|\phi_j - \phi_i| - \alpha_2|\phi_i| \end{cases} \quad (9)$$

where  $\alpha_1$  and  $\alpha_2$  are user-defined constants. The reward function is designed to return the higher value as the smaller changes in the roll angle or the motions closer to a level flight ( $\phi = 0$ ) are made. The optimal policy  $\pi^*$  at the state  $\mathbf{q}_i$  is given as

$$\pi^*(\mathbf{q}_i) = \operatorname{argmax}_{u_i} \sum_{j=1}^N [R(\mathbf{q}_i, u_i, \mathbf{q}_j) + P_{ij}(\mathbf{q}_j|\mathbf{q}_i, u_i) V^*(\mathbf{q}_j)] \quad (10)$$

We solve this dynamic programming problem defined by the Bellman equation (8) by the value iteration, which iteratively updates  $P_s$  for each state by evaluating (7). This generates a DP look-up table containing the optimal control  $u_i$  and the probability of success  $P_s$  for  $i = 1, \dots, N$ . To improve iteration speed, the sparsity of the matrices  $P_{ij}(u) \in \mathcal{R}^{N \times N}$  is exploited since each row of  $P_{ij}$  has only  $b$  nonzero entries where  $b \ll N$  thanks to the spatial vicinity of a state transition. Hence, only accessing nonzero entries of  $P_{ij}(u)$  during computation requires only  $\mathcal{O}(bN)$  rather than  $\mathcal{O}(N^2)$  time and memory at each iteration [21]. The value iteration terminates when the maximum value change over all states is less than a threshold (we use  $10^{-4}$ ).

### D. MDP path examples

Fig. 6 shows several examples of an MDP optimal path using the lateral motions in Fig. 3 and the probabilistic model in Fig. 5. Each plot shows three paths that start at the same position and heading angle but different roll angles (0, 30, -30 degrees, respectively). Due to the vehicle dynamics, distinctive routes to the goal are generated depending on the starting roll angle.

We set a goal region  $\mathcal{G}$  to a set of states satisfying  $-10 \leq x \leq 0\text{m}$ ,  $|y| \leq 3\text{m}$ ,  $|\psi| \leq 8^\circ$ , and  $|\phi| \leq 10^\circ$  for a gate located at the workspace center and oriented toward the positive  $x$ -axis. The number of discrete state transitions for each motion primitive is set to  $N_p = 3$  such that three control inputs  $u_0$ ,  $u_1$ , and  $u_2$  are sampled at  $0$ ,  $-1.0\sigma_\phi$  and  $1.0\sigma_\phi$ , respectively when  $\rho = 0.1$  in  $\sigma_\phi = \rho|\phi_{k+1} - \phi_k|$ . Their probabilities are

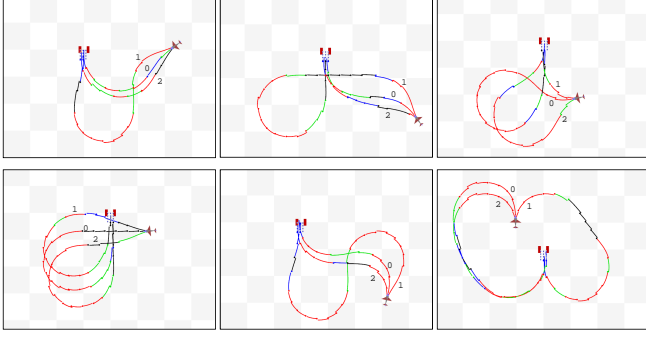


Fig. 6. MDP-based optimal motion plans in an obstacle-free 2D space: each plot shows MDP paths starting at three different roll angles ( $0$  labeled on the path indicates  $\phi_0 = 0^\circ$  and so does for  $\phi_1 = 30^\circ$  and  $\phi_2 = -30^\circ$ ). The color of a path segment in the plan represents the magnitude of a final roll angle of the corresponding lateral motion primitive (the red is  $|\phi_{k+1}| = 30^\circ$ , green is  $|\phi_{k+1}| = 20^\circ$ , blue is  $|\phi_{k+1}| = 10^\circ$ , and black is  $|\phi_{k+1}| = 0^\circ$ ).

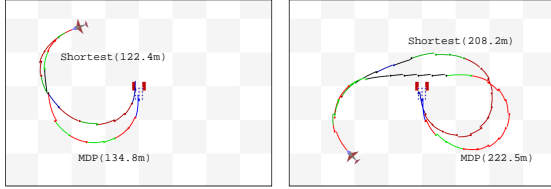


Fig. 7. Comparison between the stochastic MDP optimal path maximizing the probability of success  $P_s$  and the deterministic shortest path ignoring motion uncertainty. The MDP path is slightly longer than the shortest path but more probable to enter the gate.

computed by integrating the corresponding area under the normal curve. We set the workspace by  $x_{max} = y_{max} = 100\text{m}$  and used discretization parameters  $D = 2\text{m}$  and  $N_\psi = 120$ . The total number of resulting discrete states is  $N = 2,100,000$ . The DP problem was implemented in C++ and tested on a 2.17GHz Intel PC. The computation time for each value iteration was 55.3 sec and 50 iterations were needed until the maximum value change was less than  $10^{-4}$ . Overall, the total time to compute DP lookup table was 46.1 min.

Fig. 7 compares an optimal path maximizing the probability of success  $P_s$  with the shortest path ignoring motion uncertainty. A deterministic shortest path was computed by replacing the reward function  $R$  in (9) with the motion length and using a single state transition at the mean of the normal distribution. The MDP path is slightly longer than the corresponding shortest path but is more probable to enter the gate because less aggressive turns are involved near the gate.

In the real experiments presented later in Section VI the MDP paths in the look-up table will be described in a local frame aligned with a current target gate. Their coordinates are transformed with respect to an identified and estimated goal state.

## V. TARGET VISIBILITY CONSTRAINT

Suppose that a camera in a fixed-wing UAV has a limited field-of-view lens and is aligned downward to detect a target

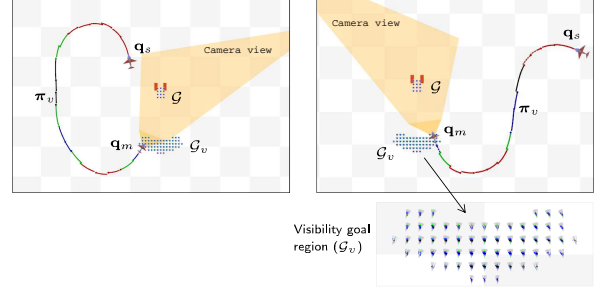


Fig. 8. MDP motion plans  $\pi_v$  to the visibility goal region  $\mathcal{G}_v$  that is used as an intermediate goal to ensure a sufficient target observation time  $t_v$ . Every state in  $\mathcal{G}_v$  satisfies  $2.5 \leq t_v(\pi_g(\mathbf{q}_i, \mathcal{G})) \leq 3.2$  sec for the original goal region  $\mathcal{G}$  when  $V_a = 10.5\text{m/sec}$  and  $\mathcal{G}$  is located around 40m ahead of the target gate.

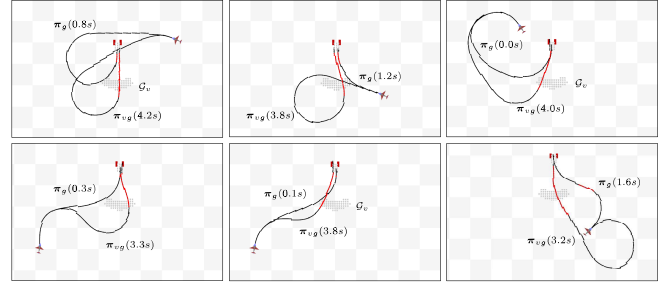


Fig. 9. Comparison between MDP motion plans when the target visibility constraint is ignored ( $\pi_g$ ) or considered ( $\pi_{vg}$ ). The red segment on each path indicates the states that the target is visible and the number is the target observation time  $t_v$ . The plan  $\pi_{vg}$  that arrives at the goal by way of  $\mathcal{G}_v$  is longer than  $\pi_g$  but guarantees the minimum  $t_v$ . The target visibility is computed when the vehicle's altitude is 18m and the camera's field of view and downward angle are 70 and 40 degrees, respectively.

on the ground. The camera's viewing area for a surrounding environment is bounded and subject to the airplane's pose. If a motion plan consists of tight turns in high roll angles near the target, the target is less likely to be exposed to the camera before the vehicle passes through the target. It is not possible to achieve seamless target observation unless specialized units such as a pan-and-tilt gimbal system or omnidirectional lens are additionally attached to the camera.

If this limited target visibility is ignored in motion planning when the confidence on a target state is low, the vehicle is more likely to miss a target because it does not have a sufficient time to observe the target and respond to a refined target pose produced from additional target observation.

### A. Minimum target observation time

To guarantee a continuous target-in-view approach with an increased confidence on the target, we define a visibility goal region  $\mathcal{G}_v$  that satisfies  $t_{v,min} \leq t_v \leq t_{v,max}$  for a non-discontinuous target observation time  $t_v$  until goal completion. Given a look-up table of the MDP plan  $\pi_g$  in Section IV-C which is computed for an original goal region  $\mathcal{G}$  around the target object  $g$ , we can find the following visibility



goal region  $\mathcal{G}_v$  by investigating  $\pi_g$  for every state:

$$\mathcal{G}_v = \{\mathbf{q}_i | t_{v,min} \leq t_v(\pi_g(\mathbf{q}_i, \mathcal{G})) \leq t_{v,max}, \\ \mathcal{V}(\mathbf{q}_k, g) = true \text{ for all } \mathbf{q}_k \text{ in } \pi_g(\mathbf{q}_i, \mathcal{G})\} \quad (11)$$

where  $\mathcal{V}(\mathbf{q}_k, g)$  denotes the visibility of the target  $g$  at the vehicle state  $\mathbf{q}_k$ . The path  $\pi_g$  starting at any state in  $\mathcal{G}_v$  always sees the target for the time longer than  $t_{v,min}$ .

Providing  $\mathcal{G}_v$  as a set of intermediate goals, we design a two-step motion planning scheme in order to impose a target observation time  $t_v$  on a resulting motion plan. Firstly, a new MDP optimal plan  $\pi_v$  is computed for  $\mathcal{G}_v$  using the same MDP framework but the goal region. Then, two individual paths ( $\pi_v$  and  $\pi_g$ ) are combined into a final motion plan  $\pi_{vg}$  via an intermediate goal  $\mathbf{q}_m$  in  $\mathcal{G}_v$ . In other words,  $\pi_{vg}(\mathbf{q}_s, \mathcal{G}) = \pi_v(\mathbf{q}_s, \mathbf{q}_m \in \mathcal{G}_v) + \pi_g(\mathbf{q}_m, \mathcal{G})$  where  $\mathbf{q}_s$  is a starting state and  $\mathbf{q}_m$  is a final state of  $\pi_v$ .

Figure 8 shows the visibility goal region  $\mathcal{G}_v$  which guarantees the minimum target observation time  $t_{v,min} = 2.5$  sec at the cruise speed  $V_a = 10.5$  m/sec, and is located around 40m ahead of the goal. In Figure 9, we compare the path produced when the visibility constraint is either ignored ( $\pi_g$ ) or considered ( $\pi_{vg}$ ). The path length of  $\pi_{vg}$  is typically longer than that of  $\pi_g$  in order to provide a minimum target observation time in  $\pi_{vg}$ .

## VI. EXPERIMENTS

Our experimental setup for the air slalom task is illustrated in Fig.1. Since it is difficult to install real gates in the air maintained at a certain altitude, we instead construct virtual gates that can be easily conceivable from real ground targets. We placed two gates are placed on the ground and their orientations are unchanged and fixed by the target color order as shown in Fig 11. The task is to autonomously search and localize gates present in a workspace and navigate through a slalom course consisting of the identified gates.

Each virtual gate was constructed using four identical squares on the ground and consist of one blue and three red squares. These squares are arranged as a rectangle in 6m wide and 6m long and its zero orientation corresponds to when the blue square is at a bottom-left corner.

### A. UAV system

Our UAV system development in Fig.10 is targeted for a small-size and low-cost platform that is deployable and expendable with a high budget-performance ratio. All of the components in Table I are commodity-level products and available in market. The fixed-wing airplane is based on a conventional RC model that has a 1.2m wingspan, three control inputs (thrust, rudder and elevation) and its maximum payload is 250g.

The UAV system consists of the following two main parts linked by wireless serial communication: (1) an airplane system which is a fixed-wing model airplane equipped with onboard sensors and processors for flight control and vision processing, and (2) a ground station which not only monitors flight status but also provides sufficient computation

TABLE I

SPECIFICATIONS OF LOW-COST COMPONENTS IN OUR UAV SYSTEM

Component	Model	Specifications
Airplane	SuperCub	3-ch, 1.2m wingspan
Battery	Cellpro	Li-Po, 900mAh
Barometer	BMP085	0.03 hPa accuracy
IMU	Analog Devices	$\pm 2g$ , $\pm 200^\circ/\text{sec}$
GPS	U-Blox LEA-5H	4Hz position update
Camera	FireFly MV	$320 \times 240$ at 15FPS
Autopilot	Ardupilot Mega	16MHz, 3 PWMs
Onboard comp.	Gumstix Overo	720MHz, Summit
Comm.	XBee PRO 900	38.4kbps, 900 MHz



Fig. 10. UAV testbed platform and onboard components

capabilities for state estimation and motion planning. The execution of navigation algorithms is distributed over on-body and off-body computation. A vision processing to detect ground targets runs on the onboard computer and their image locations are sent to the ground station for target state estimation.

### B. Vehicle and target state estimation

The estimation of vehicle and target states for the air-slalom task is based on the sensor fusion of low-grade and lightweight onboard sensors. We used an Unscented Kalman filter (UKF) [22] to combine a predicted vehicle motion and the sensor measurements in nonlinear equations. The prediction step assumes a non-accelerating and non-acrobatic maneuver that follows a current heading direction. Onboard sensor measurements from GPS, IMU, barometer and camera are combined in the update phase. Particularly, the update for the attitude ( $\phi$  and  $\theta$ ) used in the autopilot controllers is similar to the nonlinear complementary filter [23] and direction cosine matrix [24] or provided by our visual-inertial attitude estimation method [25] when line features are available.

### C. Multiple-gate air slalom

In this experiment, we used separate motion planning schemes for lateral and longitudinal maneuvers, respectively. For lateral maneuvers moving toward a gate, the MDP planner was used with taking the target visibility constraint into account. When the vehicle entered the visibility goal region, the straight line following proposed by Nelson *et al* [26] began to guide the vehicle to the gate. For longitudinal maneuvers, the altitude  $z_g$  for each virtual gate was not strictly specified but the flight altitude was maintained between the upper and lower bounds to prevent a crash ( $15m \leq z \leq 25m$ ). At the end of each lateral motion primitive, the

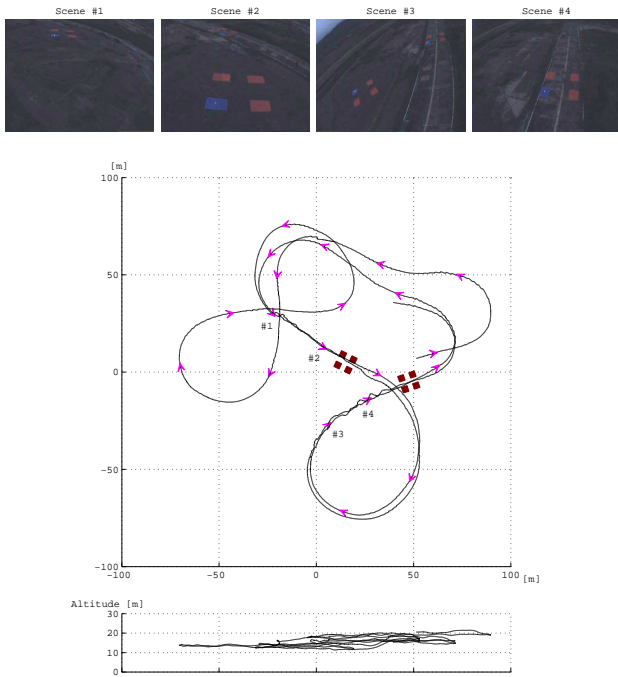


Fig. 11. Navigation result on the multiple-gate air slalom task: two gates are placed in the workspace and their orientations are unchanged and fixed by the blue square target. The whole route to enter both gates in a row is completed twice. The images at the top show the gate observations at the corresponding moments indicated in the plot.

vehicle altitude was checked for its desired bound. In the case where the altitude was out of this bound, an appropriate longitudinal motion primitive was immediately executed to push the vehicle back to a safe altitude.

The UAV first started to navigate around the workspace in a predefined pattern to search and identify all the gates. The first reliable reading of the GPS and barometer set the initial vehicle position  $\mathbf{x}_0$  in the UKF. Once four square-shaped ground targets were detected for the  $k$ -th gate, the gate state  $\mathbf{x}_g^k$  was initialized by four points in the world coordinate, each of which is an intersection point of the ground plane and the camera ray computed from a current vehicle state  $\mathbf{x}$  and target measurement.

In Fig 11, an entire navigation trajectory to enter the two gates sequentially was completed twice. The images at the top show the gate observations at the corresponding moments indicated in the plot by the number. The minimum turning radius was about 20m when a lateral motion primitive maintained the 30-degree roll angle. The relative posture between the gates was estimated to be (27.5m, -20.1m) and 45.2 degrees. This slalom setting led our optimal motion planning to eventually produce a figure-eight route shape.

The MDP motion planner achieved a successful autonomous guidance to enter the gates multiple times. Fig 12 shows the progress of the motion plan during the first slalom completion. Entering the first gate was repeated three times because position and heading deviations from the planned path were larger than the motion uncertainty modeled in the MDP due to an unexpected wind gust. The entrance to the

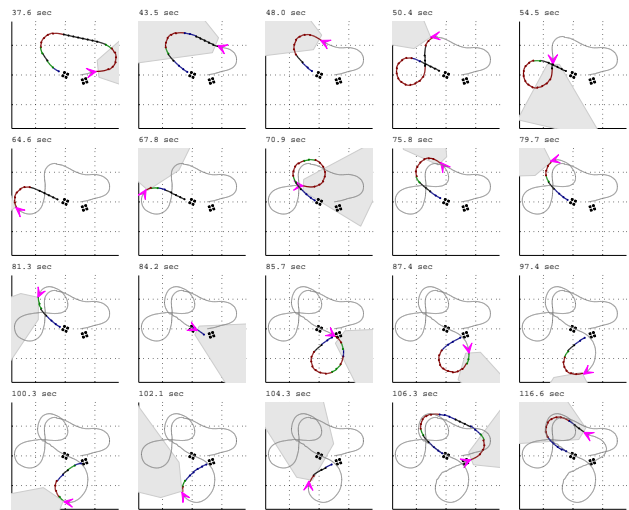


Fig. 12. MDP motion planning results in the multiple-gate air slalom task. At the end of execution of each motion primitive, a new optimal plan to the gate is found from the precomputed MDP look-up table and sent to the autopilot. The first two trials to approach the first gate fail due to insufficient target observation time, but the entrance is successful at the third approach. The gray area indicates the camera's field-of-view. The color of each motion primitive in the planned path represents its final roll angle (30 degrees in red, 20 in green, 10 in blue, and 0 in black).

second gate was smoothly accomplished.

Fig 13 shows the responses of the autopilot feedback controllers during the multiple-gate air slalom. Since the figure-eight route commanded a full swing of the roll angle, both right and left-turn lateral motion primitives were used almost equally. The maximum overshoot in the regulation of  $\phi$  was about 10 degrees when  $\phi_d$  changed quickly. The altitude gradually decreased since the execution of a lateral motion lost the altitude by up to 5m. When the altitude was below 15m, a longitudinal motion was executed to recover the altitude.

The jittering of the vehicle position that occurs each time the UAV approaches the gate is caused by a large innovation in the target measurement update in the UKF. When the camera starts to see the ground targets again after turning back, the expected target measurements are less likely to be close to the real ones due to attitude estimation error, which could be large except during a level flight. This error is thus distributed over the vehicle position and heading in the UKF until it is eliminated by a sufficient number of target observations.

## VII. CONCLUSION AND FUTURE WORK

This paper presented an integrated motion planning and control method that is a useful and efficient building block for the autonomous navigation system for an unmanned fixed-wing airplane operating in an urban environment. One challenge in motion planning for a small fixed-wing is to provide a feasible motion plan that can be not only directly executed by flight controllers but also deal with inevitable motion uncertainty due to wind and control error.

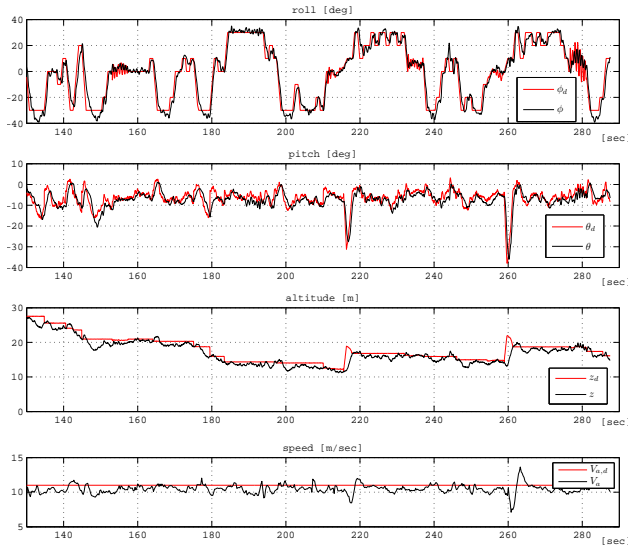


Fig. 13. Autopilot control outputs to generate lateral and longitudinal motion primitives during the multiple-gate air slalom task in Fig. 11. The MDP planner provides the desired roll angle  $\phi_d$  which is switched between -30 to 30 degrees in a 10-degree resolution. The crooked  $\phi_d$  in a high frequency indicates the line following mode near the gate entrance. Altitude recovery by the peaks in the desired pitch angle occurs only when the entrance is completed and the altitude is less than 15m.

We addressed this problem through a discrete planning based on the finite trim states of a fixed wing. A set of discrete primitive motions was constructed as transitions between the trim states; the interconnections of these motion primitives were explored to find a feasible motion plan for reaching a goal. Rather than seeking for the shortest path, we maximized the probability of success to reach a goal in the MDP using a probabilistic model of motion uncertainty. For a target to be sufficiently exposed to a camera, a minimum target observation time was also guaranteed by setting the visibility region as an intermediate goal. The airslalom experiment demonstrated that the MDP-based motion planner successfully guides the vehicle to enter a gate.

Future work includes the extension of our MDP-based planning to 3D. Since the MDP demands significant computation time and memory space when the number of states is large, we limited it to lateral maneuvers on a horizontal plane and the altitude was controlled separately. A deliberate design of state-space discretization (e.g., multi-resolution approach) or an appropriate choice of a workspace range will reduce the number of states and enable the MDP to include longitudinal motions for a complete 3D motion plan.

## REFERENCES

- [1] M. Hwangbo, J. Kuffner, and T. Kanade, "Efficient two-phase 3d motion planning for small fixed-wing UAVs," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, Apr 2007, pp. 1035–1041.
- [2] B. Sinopoli, M. Micheli, G. Donata, and T. Koo, "Vision based navigation for an unmanned aerial vehicle," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, 2001.
- [3] R. Zapata and P. Lepinay, "Flying among obstacles," in *European Workshop on Advanced Mobile Robots*, 1999.
- [4] J. Sasiadek and I. Duleba, "3d local trajectory planner for uav," *Journal of Intelligent and Robotic Systems*, vol. 29, no. 2, pp. 191–210, Oct. 2000.
- [5] J. F. Canny, *The Complexity of Robot Motion Planning*. MIT, Cambridge, 1988.
- [6] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publication, 1991.
- [7] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms and Implementations*. MIT Press, 2005.
- [8] S. M. LaValle, *Planning Algorithms*. Cambridge University Press (also available at <http://msl.cs.uiuc.edu/planning/>), 2006.
- [9] A. G. Barto, S. J. Bradtke1, and S. P. Singh2, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995.
- [10] K. Tarabani, P. Allen, and R. Tsai, "A survey of sensor planning in computer vision," *IEEE Trans. on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.
- [11] P. Svestka and M. Overmars, "Coordinated motion planning for multiple car-like robots using probabilistic roadmaps," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, May 1995, pp. 1631–1636.
- [12] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic-based path planning," in *Proceedings of ICAPS Workshop on Planning under Uncertainty for Autonomous Systems*, AAAI, June 2005.
- [13] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *AIAA Journal of Guidance, Control and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.
- [14] —, "A hybrid control architecture for aggressive maneuvering of autonomous helicopters," in *IEEE Conf. on Decision and Control*, 1999.
- [15] O. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *AIAA Journal of Guidance, Control and Dynamics*, vol. 23, no. 5, pp. 865–875, 2000.
- [16] V. Gavrilits, E. Frazzoli, and B. Mettler, "Aggressive maneuvering of small autonomous helicopters: A human-centered approach," *International Journal of Robotics Research*, vol. 20, no. 10, pp. 795–807, 2001.
- [17] C. Dever, B. Mettler, E. Feron, J. Popovic, and M. McConley, "Trajectory interpolation for parametrized maneuvering and flexible motion planning of autonomous vehicles," *AIAA Journal of Guidance Control and Dynamics*, vol. 29, no. 2, pp. 1–28, 2006.
- [18] B. Mettler and Z. Kong, "Receding horizon trajectory optimization with a finite-state value function approximation," in *American Control Conference (ACC)*, 2008, pp. 3810–3816.
- [19] R. W. Beard, *An Introduction to Autonomous Miniature Air Vehicles*. Unpublished, 2006.
- [20] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, Belmont, MA, 2000.
- [21] R. Alterovitz, M. Branicky, and K. Goldberg, "Constant-curvature motion planning under uncertainty with applications in image-guided medical needle steering," in *In Workshop on the Algorithmic Foundations of Robotics*, 2006.
- [22] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.
- [23] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel, "A complementary filter for attitude estimation of a fixed-wing uav," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, Sept 2008, pp. 340–345.
- [24] W. Premerlani and P. Bizard, *Direction Cosine Matrix IMU: Theory*, Mar. 2009. [Online]. Available: <http://gentlenav.googlecode.com/files/DCMDraft2.pdf>
- [25] M. Hwangbo and T. Kanade, "Visual-inertial UAV attitude estimation using urban scene regularities," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, Sept 2011, pp. 340–345.
- [26] D. Nelson, D. Barber, T. McLain, and R. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. on Robotics*, vol. 23, no. 3, pp. 519–529, Jun 2007.