An Online Motion Planning and Control Strategy for UAVs in Wind using Reduced Order Forward Models

Ashray A. Doshi, Surya P. N. Singh and Adam J. Postula
The University of Queensland, Australia
{a.doshi, spns}@uq.edu.au, adam@itee.uq.edu.au

Abstract

This paper presents an online motion planning strategy for Unmanned Aerial Vehicles (UAVs) in an obstacle-strewn environment that is complicated by uncertainties stemming from unmodeled effects and environmental disturbances, particularly cross-winds. During execution underlying wind currents are estimated and applied to the forward model used in online trajectory planning. To facilitate on-board computations, a reduced order forward model is used. We consider a 2D slice of the operating environment and assume that the UAV has independent bounded control over airspeed and altitude. Results are demonstrated from simulation using a medium fidelity non-linear model of a small UAV (Sig Rascal) under wind disturbances of varying magnitude and direction. The approach is tested in base winds in excess of 22 knots (50% forward speed). Results suggest that the method provides an efficient strategy for navigating to the desired destination by re-routing the trajectory as opposed to pure-pursuit methods under significant disturbance.

1 Introduction

Small fixed-wing UAVs are increasingly being used to replace manned aircraft in applications such as law enforcement, photogrammetry, disaster management and surveillance. Given the weight and operating speeds, manned aircraft are inherently stable to higher magnitudes of wind and gust, compared to their light UAV counterparts. UAV motion planning is especially difficult due to several complexities not considered by earlier path planning efforts including (as a whole), vehicle kinodynamic constraints, environmental disturbance, and geometric constraints such as obstacles.

In the case of fixed-wing aircraft, a forward speed greater than its stall speed (Vs) is required to maintain laminar flow over its wings and generate lift. Turns are executed by banking and diverting the horizontal component of lift in the direction of the turn. This limits the rates of translation of the aircraft. In order to address kinodynamic limitations of fixed-wing aircraft, UAV path planners [Ren and Beard, 2004] often impose fixed input dynamic constraints, and assume full control authority during execution. However, under significant and time-varying disturbance (such as wind currents), the plan may not always succeed due to controller saturation. The context is illustrated in Figure 1 where the UAV collides with an obstacle while attempting to stabilize a trajectory without considering wind effects in the planning stage.

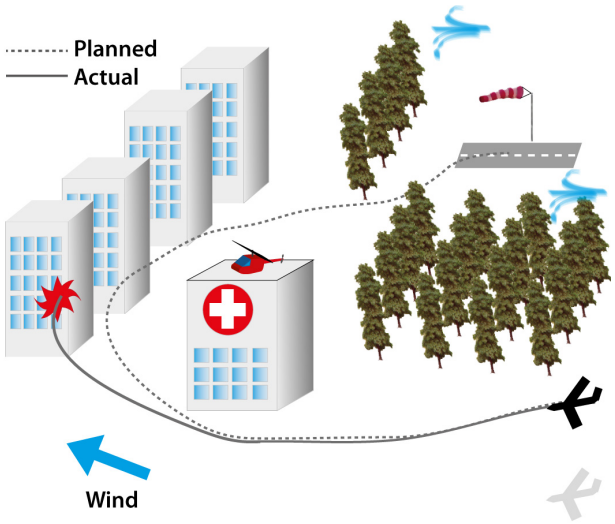

Figure 1: UAV deviating from planned trajectory under significant cross-wind and gust while executing a banked turn.

Due to inherent fast dynamics of UAVs, real-time replanning and associated computational complexity are

detrimental to the outcome of a planning strategy. If previously unknown obstacles are detected, a replan has to be carried out in real-time. This limits the application of deterministic methods in the case of urban UAV planning, which require at exponential time in the dimension of the state space of the dynamical system [Frazzoli *et al.*, 2002].

Motion planning algorithms have gained significant amount of interest in the past decades. Recently, sampling-based algorithms such as RRTs [LaValle, 1998] have demonstrated success for solving planning problems involving high dimensional search space. However, the basic algorithm relies on an accurate forward model which is difficult to determine for a real UAV application.

In this paper, we use an extension of RRT where the forward model is adapted to wind estimates during execution, followed by online replanning in order to generate new policies that are easier to execute by a local feedback controller. An approximate model of the aircraft dynamics is learnt offline based on collocation methods using high-fidelity simulation data under various environmental conditions. This approach enables online replanning performance, which is otherwise infeasible using a full flight dynamics model. The advantage is that new policies are created that drive trajectory generation based on wind effects rather than treating it purely as a disturbance.

The paper is organized as follows. In Section 2 we review related work in the field of motion planning, in Section 3 we outline the model adaptive motion planning approach. Experimental and simulation results are presented in Section 4. Conclusions and future work are presented in Section 5.

2 Related Work

Determining the motion of a UAV in wind can be treated several ways. The most direct approach is to design the vehicle with sufficiently large actuation and control surfaces so as to provide control authority in most wind conditions.

A less direct approach is to treat this as a feed-forward control problem in which the route is adapted. Model Predictive Control (MPC) uses a model to predict a sequence of process outputs at future time instants (limited horizon) by minimizing an objective function. A receding strategy is applied so that at each time instant the horizon is displaced towards the future which involves application of the first control output of the calculated sequence [Camacho *et al.*, 2004]. MPC naturally introduces feed-forward actions which can greatly improve controller performance. The downside of MPC methods is the fact the controller performance is only as good is its model approximation. In order to address modeling errors and changes to the model in flight, MPC is often assisted by online estimation of model parameters as described in [Mendel, 1973] and [Ljung, 1983]. MPC with parameter estimation has been used in UAVs for the purpose of fault tolerance (e.g., actuator faults) in [Ward and Barron, 1995], however, it fails to address overall motion planning with geometric constraints as well as time varying disturbances such as wind.

If the look-ahead is sufficiently large, this can be treated by rerouting the UAV. The overall path maybe viewed as a kinodynamic motion planning problem, which may be solved using a sample-based approach, such as the RRT algorithm [LaValle, 1998]. As with MPC, a forward model is needed, as the RRT uses this to efficiently forward simulate candidate solutions in a manner that explores the state space towards the goal.

RRT is a search algorithm well suited for high dimensional search spaces. Majority of work involved in employing RRTs for motion planning applications is focused on the EXTEND function including NEAERST_NEIGHBOUR, which in its original form is based on Euclidian distance. Reachability-guided RRT (RG-RRT) [Shkolnik *et al.*, 2009] limits the set of nodes under consideration for NEAREST_NEIGHBOUR to a set that is actually able to expand towards a given sample, which can lead to improvements in trajectory computation time. Linear Quadratic Regulator RRT [Glassman and Tedrake, 2010] demonstrated the use of LQR based pseudo-distance metric for NEAREST_NEIGHBOUR on a simulated torque-limited pendulum and acrobat. LQR-Trees [Tedrake, 2009] used LQR based cost-to-go for NEAREST_NEIGHBOR, followed by a generating a library of local reachability regions around the local trajectory based on feedback controller limitations. LQR-trees have been demonstrated in simulation using a simple pendulum setup. However, these methods do not address changing environment or online replanning performance that is required for UAVs.

Improvements in tracking error and actuator utilization were illustrated for motion planning of a 3-DOF manipulator in [Maeda *et al.*, 2011], by applying viscous friction updates to the RRT forward model. An autonomous quadruped robot for outdoor application [Raibert *et al.*,] demonstrated success in walking through rough terrain using a model based control system and trajectory primitives (contingencies). However some of the unique characteristics that distinguish UAVs from ground robots include minimum forward speed limits, high-priority attitude stability requirements and difficulty in directly sensing disturbance (wind). Small UAVs face additional challenges due to limited payload capacity which limit the on-board processing resources that can be allocated to motion planning.

Probabilistic Road Map and D* light were used for online motion planning of small helicopter UAVs with

dynamically changing geometric constraints in [Hrabar, 2008] . In another approach, RRTs were used with post-smoothing in [Yang and Sukkarieh, 2008] to address online performance and replanning. However, these do not address changing environmental conditions and associated controller saturation.

3 Model Adaptive RRT Algorithm for UAVs

The basic RRT algorithm provides a computationally efficient method of searching for a trajectory for linear time-invariant systems with geometric constraints [Lavalle and Kuffner Jr, 2000]. However, in the case of UAVs, wind disturbance is not easily observed before flight. Also, as a trajectory is executed, the UAV may find vortices near obstacles. This calls for a motion planning strategy that adapts trajectories while addressing the limited computational capacity on small UAVs (due to size and weight restrictions).

3.1 Basic RRT Algorithm

In the basic RRT algorithm [LaValle, 1998] a forward model is essential so as to extend candidate trajectories. If the environment (including the wind magnitude and direction) and vehicle are known a priori, one approach for computing trajectory candidates is to simulate this using a flight simulator, such as FlightGear [Sorton and Hammaker, 2005]. This is illustrated for a Sig Rascal UAV in Fig. 3 (of Sec 4.1).

3.2 Forward Model Adaptation

To address the disturbance uncertainty in the simulation steps of the RRT, an adaptive approach is taken. In particular, the approach uses the disturbance to drive parameter identification. This leads to an adaptive variation of the kinodynamic RRT algorithm for cross wind.

The adaption algorithm (as shown in Algorithms 1-3) works as follows: During execution of the trajectory, the process views underlying bias as coming from steady wind. Tail-wind and cross-wind components are estimated based on tracking errors similar to [Kumon *et al.*, 2005]. The UPDATE_MODEL function takes in the wind vectors and the forward model is adapted by applying the skew functions to the nominal forward model (Ref. Sec. 3.3). Upon detection of changes in wind conditions, an online replanner is triggered using the updated forward model. This way, new policies (trajectory re-routing) are generated rather than treating wind purely as a disturbance. However, it is to be noted that the feedback controller (autopilot) would have the same tracking limitations even with an adapted trajectory (with unchanged gains). A distinction of this approach is that the method returns a series of compensated waypoints that are given to the autopilot in order to track the original trajectory.

Algorithm 1: UPDATE˙MODEL (*wind*)
1: $p_{model} \leftarrow$ INITIAL˙MODEL;
2: $p_{new} \leftarrow$ ADAPT˙FORWARD˙MODEL(*wind*);
3: DISCARD (T);
4: BUILD˙RRT (x_{init}, p_{new})

Return p_{new};

Algorithm 2: BUILD˙RRT (x_{init}, p_{model})
1: T.init (x_{init});
2: **for** $k = 1$ **to** K **do**
3: (x_{rand}) $\leftarrow$ RANDOM˙STATE();
4: EXTEND (T, x_{init}, p_{model});

Return T

Algorithm 3: EXTEND (T, x_{init})
1: $x_{near} \leftarrow$ NEAREST˙NEIGHBOUR (x, T);
2: **if** NEW˙STATE ($x, x_{near}, x_{new}, u_{new}, p_{model}$) **then**
3: T.add˙vertex (x_{new});
4: T.add˙edge ($x_{near}, x_{new}, u_{new}$);
5: **if** $x_{new} = x$ **then**
6: Return *Reached*;
7: **else**
8: Return *Advanced*;

Return *Trapped*;

3.3 Forward Model Approximation for Online Operations

Computational efficiency is an additional operating constraint of (small) UAVs. Instead of running a complete 6-DOF Flight Dynamic Model (as is the basis of JBSim [Berndt and De Marco, 2009]), an approximate forward model is proposed for the forward extension steps of the aforementioned adaptive RRT.

Using a strategy similar to direct collocation [Wei *et al.*, 2008], an approximate forward model (see Fig. 3) of the form

$$\dot{x} = \hat{f}(x, u) \tag{1}$$

is estimated. Here u is the desired waypoint (e.g., WPT1) and x is the initial position, tracking x_{des} (Fig. 2). ψ_{des}, ψ_{act} are the angles subtended by aircrafts forward velocity vector to commanded waypoint and final (actual) position in time t respectively. $\hat{f}$ represents the UAV model including the controller effects of the waypoint autopilot. Similarly, a skew function is approximated based on effects of steady wind vector on the nominal (no wind) function $\hat{f}$ using third-order order polynomial structure. The parameters are estimated using linear least squares optimization for various wind conditions and aforementioned states offline using time history of simulation data. The inverse of the skew function is computed in order to determine the series of compensated waypoints.

This can also simplify the adaptation approach by assuming that the cross wind is the primary source of

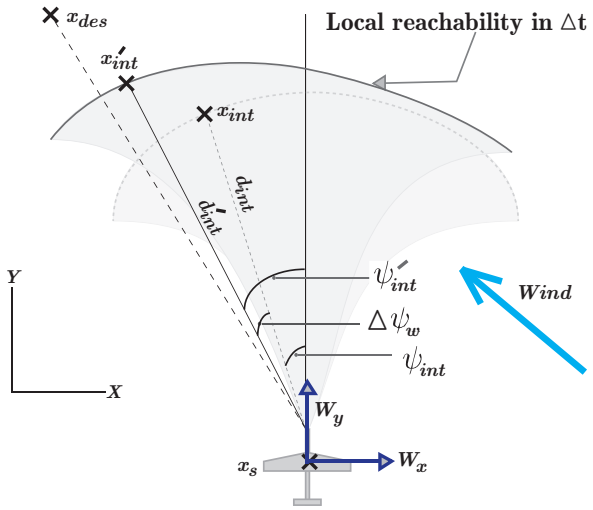

Figure 2: UAV model approximation for 2D search showing effects of waypoint following autopilot. The UAV is initially at x_s given a desired waypoint x_{des} tracks to a final position of x_{int} in zero wind conditions. When the UAV is subject to wind, the cross and tail-wind components drive the final estimate to point x'_{int}.

steady-state disturbance. While this results in an ensemble disturbance estimate, it is acceptable in this case as the system does not have direct observation of cross-winds from other sources. This is given by:

$$b_1 + b_2 \times W_x + b_3 \times W_y = \Delta\psi_w \quad (2)$$

$$c_1 + c_2 \times W_x + c_3 \times W_y = \Delta d_w \quad (3)$$

Where W_x and W_y are the magnitudes of wind in the x and y directions. $\Delta\psi_w$ and Δd_w are the offsets due to wind on d_{int} and $\Delta\psi$ respectively. b_i and c_i can be determined by using linear least squares regression on the training data.

A detailed explanation into deriving Reduced Order Forward Models (RFMs) can be found in [Doshi *et al.*, 2012]. Together the online adaptation and approximation methods provide an online solution for continuous trajectory planning and control. Thus, the adaptive planning method can handle the bias, leaving the autopilot to regulate for high-frequency disturbances. By focusing on flight trajectories in wind (as compared to fuel-burn and other associated metrics), this approximation can provide significant computational savings. For example, 1000 iterations of 10-second way-point simulation (per requirements of the forward model of RRTs) takes 20sec (Intel i7 2.8GHz PC), where as via this approach it takes 0.1 seconds.

4 Simulation Results: UAV Navigation under Disturbance

We demonstrate this approach on a small fixed-wing UAV model (Sig Rascal) in two dimensions flying through obstacles in varying wind conditions. Non-linear small fixed-wing UAV (Sig Rascal) dynamics are simulated by using FlightGear [Sorton and Hammaker, 2005] full Flight Dynamics engine.

The simulation pipeline begins with the generation of an initial trajectory in the form of waypoints. Waypoints are tracked by a local feedback controller (Such as a waypoint autopilot similar to [Beard *et al.*, 2005]). The autopilot is also tasked with maintaining constant airspeed and altitude in order to facilitate the use of a full flight dynamics model. During runtime, cross-track error is analyzed and co-related to wind components. This is followed by online replanning using the updated forward model.

4.1 Simulation under Normal (No-Wind) Conditions

To test the basic operation of approach flights were constructed to a destination on the other side of two obstacles.

The success criterion (final destination) of the motion planner is defined by a circle of radius ε_{rrt} (see Fig. 3). Similarly, in order to prevent the autopilot from hunting for waypoints, the success criteria of sub-goals (waypoints) is set to ε_{wpt}. As expected, in flight operations under little disturbance the actual flight path approximates that of the expected (forward planned) trajectory. The variation seen is a consequence of the forward model approximation (see also Sec. 3.2) and can be seen in Fig. 3. That is, the Adaptive RRT uses an approximation, where as the simulator is computing the full dynamics.

4.2 Simulation under Wind Disturbance Conditions

The approach is tested for the case of a small fixed-wing UAV navigating in various wind conditions. For example, even under moderate wind conditions (15 knots) the way-point controller will saturate and no-longer be able to track waypoint. A consequence of this is that the UAV risks deviating to an extent that it flies into obstacles. This is illustrated in Fig. 4.

A simulation in moderate winds (18 knots or 40% airspeed) was conducted with the direction of wind from 270 degrees (left to right) as shown in Fig. 5. The UAV initially tracks a trajectory generated offline without any wind information. Upon noticing large deviations at the first-waypoint, the algorithm re-computes a series of waypoints that are compensated for the wind and actuator constraints of the UAV. This is illustrated in Fig. 5. Here the Compensated WPT for the same

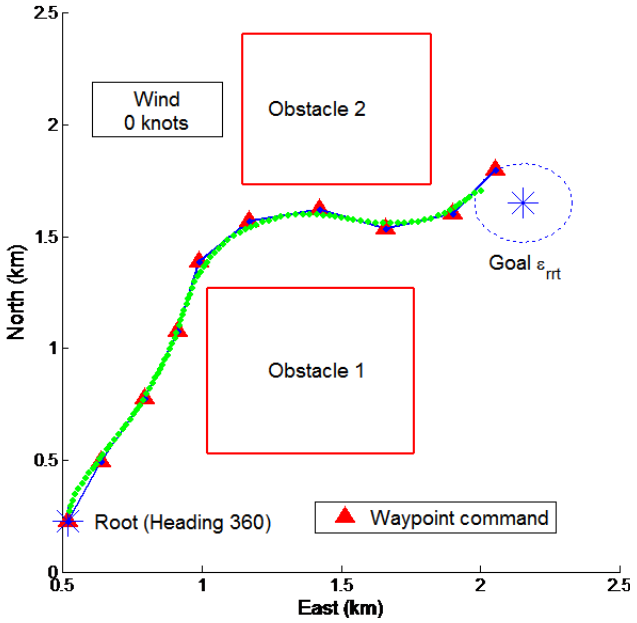

Figure 3: Trajectory generated offline using RRT without model updates is successfully tracked by an autopilot (green) in no wind conditions.

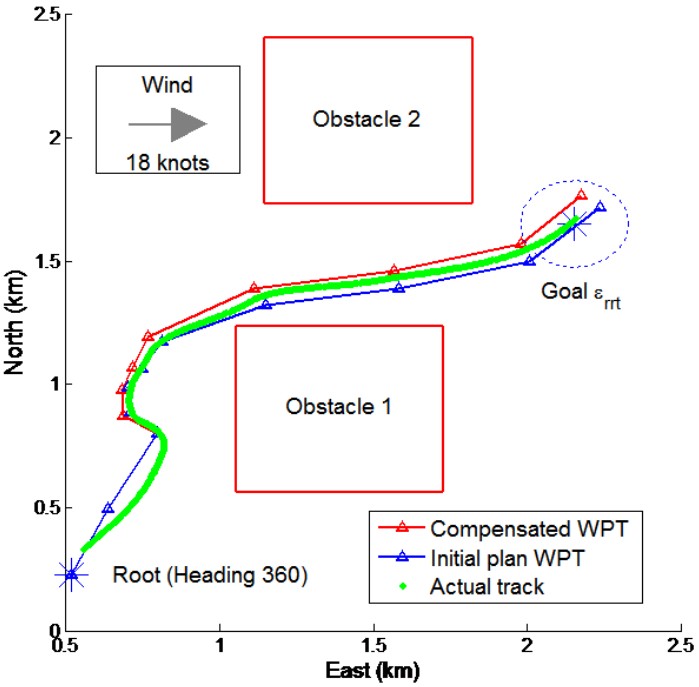

Figure 5: Simulation in moderate winds at 18knots from left to right (40% air-speed). RRT forward model is updated (online replanning) and wind-compensated waypoints are tracked by the autopilot using the same feedback gains

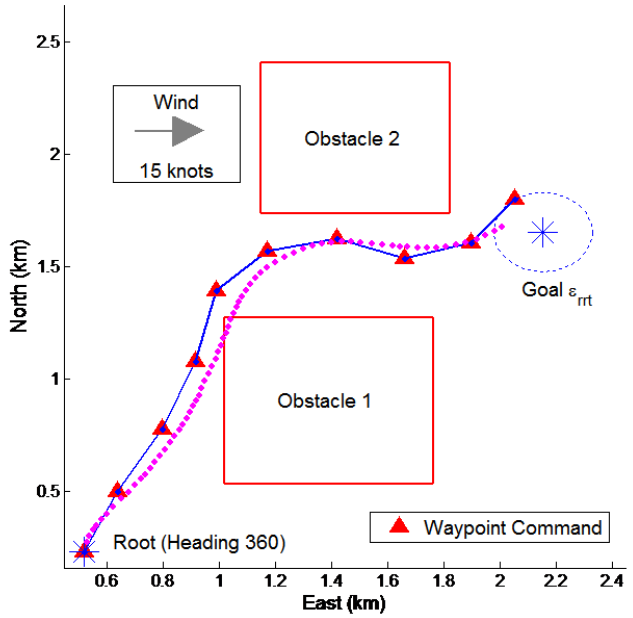

Figure 4: Trajectory generated offline using RRT under significant steady wind without model updates is infeasible to track by a waypoint following autopilot (magenta).

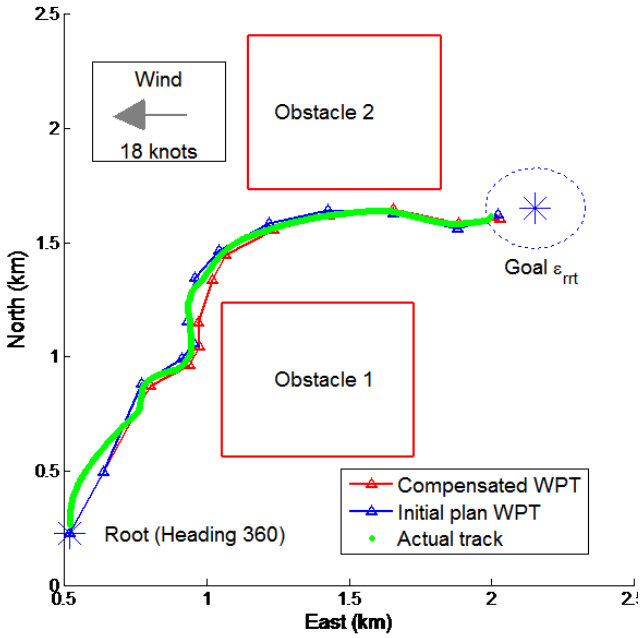

Figure 6: Simulation in moderate winds at 18knots from right to left (40% air-speed).

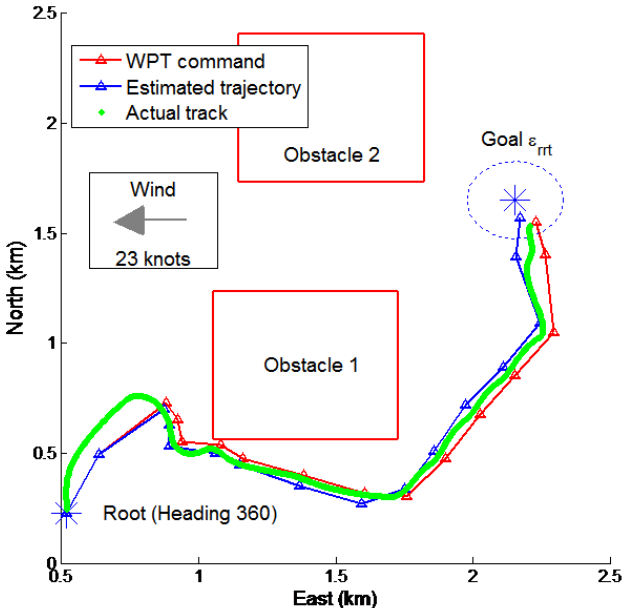

Figure 7: Simulation in high winds at 23 knots from left to right (50% air-speed). Policy is updated to re-reoute the trajectory

initial destination routing as in Fig. 4 (the Initial plan WPT). Waypoints have to be compensated because the feedback controller is tuned under normal conditions (no wind). By having the waypoint control of the UAV target an alternative set of waypoints the overall resulting UAV flight path avoids obstacles and more closely tracks the initial plan (depicted in green).

One of the effects of forward model adaptation is apparent Fig. 5 where the waypoints are spaced closely around areas where the aircraft is expected to face a head-wing or a strong cross-wind. In contrast, the waypoints are spaced away from each other in areas that the UAV is expected to get assistance from a tail-wind.

Similar to the simulation in Fig. 5, a simulation was conducted by changing the wind direction (right to left) while keeping the same magnitude (as shown in Fig. 6). It is observed that the Adaptive RRT alters the path to handle the change in wind direction, as opposed to traditional methods that utilize a high-gain feedback controller to stabilize the same trajectory.

Similarly, a simulation was conducted in high winds of 23knots or 50% of the UAVs forward speed blowing from right to left (while keeping the same wind direction as shown in Fig. 6). Fig. 7 shows that the Adaptive RRT is able to re-route around obstacle 1 (different to Fig. 6) to generate a feasible trajectory.

The later cases show the effect of the adaptive approach to compensate for the wind by updating the estimated wind state. This guides the UAVs way-point

Figure 8: Small fixed-wing UAV (1.0m wing-span, 950g MOTW) platform used for flight testing.

controller to avoid the obstacle in cases where if left to its own devices it would not be able to. This effect is particularly noticeable in the case of a strong headwind disturbance (Fig. 7) in which the approach shifts the routing below and to the right of the obstacle when it observes such large deviations at the first waypoint. This also results in a routing in which the UAV moves around the Obstacle 1 even though this new routing is initially opposite to the direction provided by the initial plan.

Run-time Performance

The computation time on an Intel i7 2.8GHz PC for the RRT search described in section 4, utilising a reduced-order forward model (RFM) is significantly less at an average of 0.9 seconds to generate a trajectory (Fixed 3600 forward iterations), when compared to that of JSBSim which took 12 minutes.

5 Conclusions and Future Work

We presented the use of Adaptive RRTs that consider the effects of wind to form a policy for navigating in wind rather that treating it purely as a disturbance. To facilitate on-board and online operations, the use of an approximate model of reduced complexity was presented that significantly improves computational requirements compared to the use of a full (high-fidelity) model. The method is well suited for applications that require online replanning based on changes in environmental conditions.

Work is under way to extend the method will to a 3D case with varying altitude and airspeeds. The method will be tested in real flight on small fixed-wing UAV platforms such as the UQ-TS01 (shown in fig. 8).

References

[Beard *et al.*, 2005] Randal W Beard, Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen, Walt Johnson, Timothy McLain, and Michael Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *Journal of Aerospace Computing, Information, and Communication*, 2(1):92–108, 2005.

[Berndt and De Marco, 2009] Jon S Berndt and Agostino De Marco. Progress on and usage of the open source flight dynamics model software library, jsbsim. In *Proceedings of the 2009 AIAA Modeling and Simulation Technologies Conference, Chicago, AIAA Paper*, volume 5699, 2009.

[Camacho *et al.*, 2004] Eduardo F Camacho, Carlos Bordons, Eduardo F Camacho, and Carlos Bordons. *Model predictive control*, volume 2. Springer London, 2004.

[Doshi *et al.*, 2012] Ashray A Doshi, Surya PN Singh, and Adam J Postula. Towards reduced-order models for online motion planning and control of uavs in the presence of wind. In *2012 Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association (ARAA), 2012.

[Frazzoli *et al.*, 2002] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.

[Glassman and Tedrake, 2010] Elena Glassman and Russ Tedrake. Lqr-based heuristics for rapidly exploring state space. In *Submitted to IEEE International Conference on Robotics and Automation. ICRA*, 2010.

[Hrabar, 2008] Stefan Hrabar. 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 807–814. IEEE, 2008.

[Kumon *et al.*, 2005] Makoto Kumon, Ikuro Mizumoto, Zenta Iwai, and Masanobu Nagata. Wind estimation by unmanned air vehicle with delta wing. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1896–1901. IEEE, 2005.

[Lavalle and Kuffner Jr, 2000] Steven M Lavalle and James J Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*. Citeseer, 2000.

[LaValle, 1998] Steven M LaValle. Rapidly-exploring random trees a new tool for path planning. 1998.

[Ljung, 1983] Lennart et al. Ljung. Theory and practice of recursive identification. 1983.

[Maeda *et al.*, 2011] Guilherme J Maeda, Surya PN Singh, and Hugh Durrant-Whyte. A tuned approach to feedback motion planning with rrts under model uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2288–2294. IEEE, 2011.

[Mendel, 1973] Jerry M Mendel. *Discrete techniques of parameter estimation: the equation error formulation*. M. Dekker, 1973.

[Raibert *et al.*,] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, and Rob Playter. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress of the International Federation of Automatic Control*.

[Ren and Beard, 2004] Wei Ren and Randal W Beard. Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints. *Control Systems Technology, IEEE Transactions on*, 12(5):706–716, 2004.

[Shkolnik *et al.*, 2009] Alexander Shkolnik, Matthew Walter, and Russ Tedrake. Reachability-guided sampling for planning under differential constraints. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2859–2865. IEEE, 2009.

[Sorton and Hammaker, 2005] Eric F Sorton and Sonny Hammaker. Simulated flight testing of an autonomous unmanned aerial vehicle using flightgear. 2005.

[Tedrake, 2009] Russ Tedrake. Lqr-trees: Feedback motion planning on sparse randomized trees. *Robotics: Science and Systems conference*, 2009.

[Ward and Barron, 1995] DG Ward and RL Barron. A self-designing receding horizon optimal flight controller. In *American Control Conference, 1995. Proceedings of the*, volume 5, pages 3490–3494. IEEE, 1995.

[Wei *et al.*, 2008] Shangming Wei, Milos Zefran, and Raymond A DeCarlo. Optimal control of robotic systems with logical constraints: Application to uav path planning. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 176–181. IEEE, 2008.

[Yang and Sukkarieh, 2008] Kwangjin Yang and Salah Sukkarieh. 3d smooth path planning for a uav in cluttered natural environments. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 794–800. IEEE, 2008.