Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Tracking Controllers for Small UAVs with Wind Disturbances: Theory and Flight Results.

Stephen Jackson, John Tisdale,
Maryam Kamgarpour, Brandon Basso and J. Karl Hedrick

Abstract— **This work outlines two approaches for small unmanned aerial vehicles (UAVs) performing surveillance with fixed cameras. Small UAVs present significant control challenges, due to relatively low-bandwidth actuation and significant disturbances due to wind. This work features implementations of a spatial sliding mode controller and a receding-horizon kinodynamic controller. The spatial sliding mode controller is designed to follow a desired aircraft path which places the camera-footprint on the desired locations, while the kinodynamic controller is designed to directly track a camera-footprint path. Since our objective is surveillance, we aim to compare the effectiveness of each controller in tracking a desired sensor path. Discussion of each controller is followed by simulation and flight test results.**

I. INTRODUCTION

Small UAVs with fixed cameras have become prevalent in many applications. While gimbaled camera systems are becoming increasingly popular, the vast majority of UAVs in use today feature fixed cameras. In this work, we examine two methods for low-level aircraft control for target sensing using a fixed, downward-pointing camera. These controllers function in a control architecture in which desired vehicle paths are developed from a receding horizon path planning scheme. Receding horizon, or model predictive, path planning has been featured recently in many works, such as [1], [2], and [3]. In many receding horizon schemes, a high-level trajectory optimization routine is coupled with a low-level path following controller. The high level path planner uses a simplified model, and develops paths at a relatively low rate. A low-level controller is necessary to track the path developed by the planner.

In this work, a high-level planner develops paths to maximize information collection. Fundamentally, the planner is optimizing an objective function over the space of expected sensor placements. The sensor placement is determined from the vehicle location and attitude. A four dimensional path is developed, assuming constant altitude and pitch, that specifies the desired vehicle state. Additionally, the projection of the camera footprint on the ground plane is computed and given to the controller.

Two different low-level control approaches are used and their performances are compared. The first is a spatial sliding mode controller that attempts to minimize the difference between the desired and actual aircraft positions. Sliding mode control is known for its robustness with respect to model uncertainties and disturbances and has been applied ubiquitously in the field of trajectory tracking for autonomous vehicles. An overview of sliding mode control may be found in [4]. Spatial sliding mode control for UAVs was first introduced in [5]. It is used for the cases in which the desired path is described as a function of a spatial coordinate rather than time. A spatial sliding surface is defined as a stable differential operator, with respect to a defined spatial variable as opposed to time in the temporal sliding control, that acts on the error. The performance of the spatial sliding mode controller can be analyzed similarly to the temporal one.

The second controller studied is a probabilistic receding-horizon kinodynamic controller. This controller attempts to minimize the error between the camera projection on the ground and the desired camera projection. The kinodynamic controller is based on the work by [6]. It has many advantages over other existing control methods with respect to finding feasible paths through obstacles. Probabilistic arguments have been used to guarantee its optimality. In this experiment, it is used primarily for path tracking. However, we anticipate flights in cluttered or dynamic environments in future experiments.

The two control algorithms aim to minimize different errors: the sliding mode controller is minimizing aircraft path error and the receding-horizon controller is minimizing sensor footprint path error. However, at the higher path-planner level, the control objectives are both based on the same desired sensor path. The respective control actions try to achieve the same result: placing the sensor on a desired trajectory. Results from the two controllers are then comparable in terms of sensor path average error and variance. Controlling sensor's line-of-sight has gained increasing interest due to increasing use of sensors mounted on moving vehicles to track paths or dynamic targets [7]. In the results section, we also discuss the effect of explicitly modeling the sensor footprint path in the control strategy–a topic of interest for mobile robotics applications where the end goal is to aim the sensor as opposed to the vehicle.

II. CONTROLLER FORMULATION

The controllers featured in this work exist in a larger system designed for multi-UAV cooperative sensing. A receding-horizon path planner, designed to maximize an information-based objective function, is used to develop aircraft paths over some short planning horizon. The path planner minimizes a cost $J(x, y, \psi, \phi)$ subject to the system

This work was supported by the Office of Naval Research and the National Science Foundation

The authors are from the Department of Mechanical Engineering, University of California, Berkeley. Berkeley, CA, 94720, USA.

dynamics. Here, position of the UAV in the plane is represented by (x, y) as shown in Fig. 1. The vehicle velocity is denoted by V, and ψ, ϕ denote the yaw and roll angles respectively. We represent the components of wind velocity by W_x and W_y. The equations of motion of the unicycle model are given by:

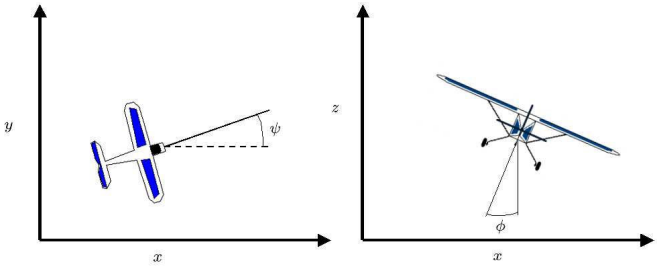

Fig. 1. Unicycle Model

$$\begin{aligned} \dot{x} &= V cos(\psi) + W_x \\ \dot{y} &= V sin(\psi) + W_y \\ \dot{\psi} &= u \\ |u| &\leq \dot{\psi}_{max}. \end{aligned} \tag{1}$$

In addition, the algebraic equation relating yaw rate and roll angle are calculated from the coordinated turn assumption as follows:

$$\phi = -\arctan\left(\frac{V\dot{\psi}}{g}\right) \tag{2}$$

Equation (1) is discretized, and then used to calculate a path, using the control input u. The output of the path planner is the optimal input u_d along with the optimal path characterized by x_d, y_d, ψ_d, ϕ_d. Here, subscript d is used to denote desired path. Additionally, the desired sensor path (x_d^s, y_d^s), defined by projecting a vector perpendicular to the camera image plane onto the ground plane, as shown in Fig. 1, is calculated and is available for the controller. The sensor path (x_s, y_s) is related to the UAV path (x, y) through:

$$\begin{aligned} x_s &= x - h\tan(\phi)\sin(\psi) \\ y_s &= y + h\tan(\phi)\cos(\psi), \end{aligned} \tag{3}$$

where h is the altitude of the UAV. Using relation (2) between yaw rate and roll angle in the sensor path (3) results in:

$$\begin{aligned} x_s &= x - c\dot{\psi}\sin(\psi) \\ y_s &= y + c\dot{\psi}\cos(\psi), \end{aligned} \tag{4}$$

where $c = -\frac{hV}{g}$. Since there is no bound on the turn acceleration in the unicycle model, the desired camera path may be discontinuous. A thorough discussion of the path planner may be found in [2].

Due to disturbances, modeling, and measurement uncertainties, open loop control of the aircraft, using the output of the path planner, is undesirable. Two low-level controllers that run at a much faster rate than the path planner steer the aircraft on the desired path. The control architecture is shown in Fig. 2.

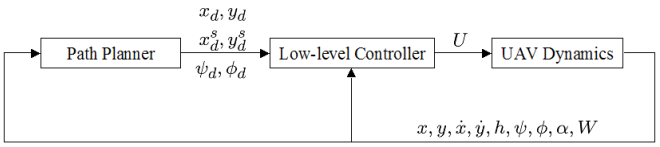

Fig. 2. Control Architecture Diagram.

The spatial sliding mode controller uses the unicycle model described above. The input to the controller is the desired UAV path (x_d, y_d). This controller attempts to steer the UAV path (x, y) to the desired path (x_d, y_d). We anticipate that this controller will result in good tracking of (x_d, y_d). However, it is possible that this approach may result in poor sensor footprint tracking, as the controller does not directly account for the sensor. For this reason, we have also designed the kinodynamic controller, which explicitly accounts for control of the sensor footprint. In the next section we describe the design of the sliding mode controller in more detail.

A. Spatial Sliding Mode Controller

Let $y_d = f(x)$ be a continuous path to be followed by the UAV with continuous first and second derivatives. Assuming that the curvature of the path is lower than the achievable curvature determined from the UAV input bound, one can design the control so that perfect path tracking is attained given perfect initial condition and no disturbances. In order to have robustness with respect to the wind disturbance and the deviations from the path we consider a spatial sliding mode controller as proposed in [5]. The spatial sliding mode controller is derived similarly to the well-known temporal sliding mode controller. It can be applied in cases where the vehicle needs to track a path rather than a trajectory. The spatial term refers to the fact that the derivatives of the error are taken with respect to one of the spatial coordinates rather than time.

Since the desired path is given as a function of the x-coordinate, we choose x as the spatial variable. We define the error as $e = y - f(x)$. We use $'$ to denote derivatives with respect to the x-coordinate. For this derivative to exist, x needs to be monotonically increasing or decreasing. The path planner defines the positive direction of x as the desired direction of travel and generates paths such that x is monotone along each path segment [5]. We define the sliding surface as $s = e' + \lambda e$ which is a curve in the (x, y) plane. On the sliding surface, $s = 0$, the error dynamics are governed by a stable first-order ordinary differential equation. Hence, on this surface, the error converges to zero exponentially. In order to ensure the UAV reaches the sliding surface in finite time from anywhere in the (x, y) plane, we choose the control such that the sliding condition:

$$ss' < -\eta|s| \tag{5}$$

is satisfied everywhere in the plane. Here, η is a positive constant whose magnitude governs the speed of convergence to the sliding surface.

We express the UAV dynamics given in (1) with respect to the spatial variable

$$\begin{aligned} x' &= 1 \\ y' &= \frac{V\sin(\psi)+W_y}{V\cos(\psi)+W_x} \\ \psi' &= \frac{u}{V\cos(\psi)+W_x}. \end{aligned} \tag{6}$$

We make the following assumptions on the wind disturbance: $W_x = \bar{W_x}+\delta W_x$, $W_y = \bar{W_y}+\delta W_y$, $\dot{W_x} = \bar{\dot{W}_x}+\delta\dot{W}_x$ and $\dot{W}_y = \bar{\dot{W}_y} + \delta\dot{W}_y$. Here, $\bar{W_x}, \bar{W_y}, \bar{\dot{W}_x}, \bar{\dot{W}_y}$ are our estimates of the wind speed and acceleration components. We assume that the unknown terms have known bounds: $|\delta W_x| \le \alpha_x$, $|\delta W_y| \le \alpha_y$, $|\delta\dot{W}_x| \le \dot{\alpha_x}$, $|\delta\dot{W}_y| \le \dot{\alpha_y}$. From (6) we find the spatial derivative of the sliding surface as:

$$s' = au + b + c, \tag{7}$$

where we define a, b, and c for notational convenience as follows:

$$\begin{aligned} a &= \frac{(V_g^2+V^2-W^2)}{2\dot{x}^3} \\ b &= \frac{\dot{x}\dot{W}_y - \dot{y}\dot{W}_x}{\dot{x}^3} \\ c &= -f''(x)+\lambda(y'-f'(x)). \end{aligned} \tag{8}$$

In the above, $V_g = (\dot{x}^2+\dot{y}^2)^{\frac{1}{2}}$ and $W = ({W_x}^2+{W_y}^2)^{\frac{1}{2}}$ are the ground and wind speed respectively. Given that the controller has access to $\dot{x}, \dot{y}, f'(x), f''(x)$, the value of c is known while a and b are not exactly known as they contain uncertain wind velocity and acceleration terms. Define $\bar{a}$ and $\bar{b}$ as our mean estimates of a and b:

$$\begin{aligned} \bar{a} &= \frac{(V_g^2+V^2-\bar{W}^2)}{2\dot{x}^3} \\ \bar{b} &= \frac{\dot{x}\bar{\dot{W}}_y - \dot{y}\bar{\dot{W}}_x}{\dot{x}^3}. \end{aligned} \tag{9}$$

We then choose

$$u = \frac{-\bar{b}-c-K\text{sgn}(s)}{\bar{a}}, \tag{10}$$

where K is a positive constant. The resulting sliding surface derivative is given by

$$\begin{aligned} s' &= (1-\frac{a}{\bar{a}})(\bar{b}+c) - \frac{a}{\bar{a}}K\text{sgn}(s) + \delta b, \\ \delta b &= \frac{\dot{x}\delta\dot{W}_x - \dot{y}\delta\dot{W}_y}{\dot{x}^3}. \end{aligned} \tag{11}$$

To achieve sliding condition stated in (5) we choose

$$K = \max|(\frac{\bar{a}}{a}-1)(\bar{b}+c) + \frac{\bar{a}}{a}\delta b| + \eta, \tag{12}$$

where η is a positive constant that guarantees sliding in the case of worst disturbances. The maximization in the above equation is taken over δW_x, δW_y, $\delta W_x'$, and $\delta W_y'$. It can be calculated analytically. The resulting K is time-varying and is a function of $\dot{x}, \dot{y}, f'(x), f''(x), \lambda$.

As with the temporal sliding mode controller, the above controller results in chattering near the sliding surface. To prevent chattering, as we get close enough to the switching surface, $|s| < \phi$ for some $\phi > 0$, we approximate the discontinuous term $\text{sgn}(s)$ by s/ϕ. Hence, using this controller, we approach the boundary layer, $|s| < \phi$, in finite time. Once inside this boundary layer, we remain inside it.

In practice, we observed that K can be chosen as a constant whose value needs to be tuned. Both K and λ should be chosen small to avoid oscillations in the input and to achieve the threshold bound on the input.

Next, we consider applying the spatial sliding mode technique to design a controller that ensures tracking of sensor path. Recall that the sensor path is related to the UAV path as stated in equation (4). In the unicycle model, it is assumed that the yaw rate is set directly by the control input. Hence, the control input appears directly in the sensor path equations (4). In order to control the sensor path, we need to consider a more accurate model of the UAV dynamics which accounts for time delay in control of the yaw rate. For this reason, the aircraft model in this controller has been extended beyond unicycle model to include the kinematic constraints of the vehicle.

The detailed model assumes that the lateral component of the lift force, f_r, is due to the centripetal acceleration of the UAV performing a coordinated turn:

$$f_r = \frac{mV^2}{r} = mV\dot{\psi}. \tag{13}$$

The derivative of the lateral force on the aircraft is governed by the lower level roll dynamics of the UAV. These dynamics are given by a first order approximation of the roll rate of the UAV, based on experimental results as:

$$\dot{f_r} = k\left(u - \dot{\psi}\right). \tag{14}$$

By taking the derivative of f_r in (13) and setting it equal to (14), we obtain the roll dynamics:

$$\ddot{\psi} = \frac{k}{mV}\left(u-\dot{\psi}\right) \tag{15}$$

Hence, the fourth-order model of the UAV dynamics is given by:

$$\begin{aligned} \dot{x} &= V\cos\psi + W_x \\ \dot{y} &= V\sin\psi + W_y \\ \ddot{\psi} &= \frac{k}{mV}\left(u-\dot{\psi}\right) \end{aligned} \tag{16}$$

In order to apply the spatial sliding mode to control the sensor path, assume that the desired sensor path is given by $y_d^s = f(x_d^s)$. Consider choosing x_s as the spatial coordinate. Differentiating x_s with respect to time we find:

$$\dot{x}_s = b_1(\psi, \dot{\psi}) + b_2(\psi)u, \tag{17}$$

where $b_1 = V\cos(\psi) + W_x + \frac{hV}{g}(\dot{\psi}\cos(\psi) - \frac{k}{mV}\sin(\psi))$, and $b_2 = \frac{hk}{gm}\sin(\psi)$. From above, we see that $\dot{x}_s$ depends on the input u and in particular, we cannot guarantee that x_s would be monotone. Consequently, x_s cannot be taken as a spatial variable. The same result holds for considering y_s as the independent variable. This motivates applying other control techniques for keeping the sensor path error small. In the next section we consider a receding horizon controller to address this problem.

B. Kinodynamic Receding Horizon Controller

The second control algorithm is based on a randomized kinodynamic motion planner described in [6] and [8]. The motion planner allows for the possibility of restricted airspace, moving obstacles and moving targets. It is intended for environments where it may be difficult to find a feasible solution, much less an optimal solution. Though this paper only focuses on the controller's tracking performance, we chose to use this controller because it provides the basis for future work that will include more dynamic environments.

Hsu's trajectory planning algorithm randomly and uniformly samples the control space of the vehicle in order to generate many potential vehicle trajectories. It is shown that this planner is probabilistically complete, i.e. as the number of random samples increases, the probability of finding a feasible solution exponentially increases to one. In our case, a feasible solution is defined as a trajectory that ends with the aircraft's sensor footprint on the desired sensor path. By generating many sample trajectories, many feasible solutions are likely to occur. The controller chooses the solution with the lowest associated cost, where the definition of the cost is followed. The validity of the potential trajectories is entirely dependent on the accuracy of the aircraft model. Hence, we use the fourth-order model developed in previous section (16).

The controller presented here differs from a more traditional rapidly-expanding random tree motion planner in two ways. The first is that it uses a receding horizon approach. The unmodeled dynamics of both the aircraft and the environment make it extremely difficult to accurately calculate the cost of trajectories that extend too far into the future. Rather than trying to calculate the exact cost of every trajectory that ends with the aircraft's sensor footprint on the desired sensor path, the controller uses a receding horizon approach that calculates the exact cost of the first n steps of each feasible solution. It then uses a heuristic to estimate the remaining cost of each trajectory. The number of steps the receding horizon calculates into the future can be chosen to match the extent of the aircrafts knowledge of the future.

The second difference, is how the controller calculates its search tree. Rather than uniformly sampling the configuration space, it creates a vector of n uniformly sampled control actions, which defines a complete trajectory from the initial configuration to final configuration of the aircraft. Though this process reduces the efficiency of the RRT algorithm, experimental results show that increasing the speed of our controller beyond 10hz does not significantly increase the controller's performance.

The control algorithm can be broken into two main parts: the random trajectory generator and the cost function that defines the optimal solution.

The control space of the aircraft is defined as the set of turn rates ranging from $\dot{\psi}_{min}$ to $\dot{\psi}_{max}$ radians per second. The trajectory generator uniformly samples the control space to create a vector of n turn rates, which represent n seconds of potential flight.

$$U = [u_1, u_2, \ldots, u_n] \tag{18}$$

Several trajectories are generated that sample the entire configuration space of the aircraft. By sampling the entire configuration space, rather than performing a traditional gradient-descent optimization, the controller is able to avoid local minima in the configuration space. Several other trajectories are generated by adding small random variations to the controller's last optimal solution. By using the previous output of the controller as a starting point, the controller is more likely to improve the solution with every run. In addition to these random vectors, several canonical paths are added; for example, a hard left turn, a hard right turn, and a straight path. Random sampling is unlikely to generate these canonical trajectories, but in many circumstances they will have the lowest associated costs.

Let the sensor error be defined as the distance between desired sensor path and the actual sensor path:

$$e_t = \left\| \begin{bmatrix} x_t^s \\ y_t^s \end{bmatrix} - \begin{bmatrix} x_d^s \\ y_d^s \end{bmatrix} \right\|. \tag{19}$$

The cost associated with each trajectory is equal to the sum of the sensor errors at every second along the potential trajectory. It is given by:

$$\begin{aligned} C &= \sum_{t=1,2}^{n} \left[e_t^2(1 + (u_t - u_{t-1})^2)\right] + C_{inf} \quad (20) \\ C_{inf} &= k e_n \left(\frac{e_n}{V} + 1\right) \end{aligned}$$

The controller chooses the trajectory with the lowest associated cost and sends the first turn rate of that trajectory, u_1, to the lower level turn rate controller.

III. RESULTS

A. Hardware-in-the-Loop Results

The system was tested hardware-in-the-loop (HIL) to verify controller performance prior to flight. In the tests, both controllers attempt to fly a specified sinusoidal path. The results are presented below.

For the sliding mode controller, the average aircraft path error was 1.3 meters, while the average sensor path error was 4.0 meters.

For the kinodynamic controller, the average aircraft path error was 3.7 meters, while the average sensor path error was 6.1 meters. Both controllers demonstrate the ability to track the specified path. It is difficult to truly assess the relative performance of the two controllers with simulation. Comparative data is presented in the flight test results.

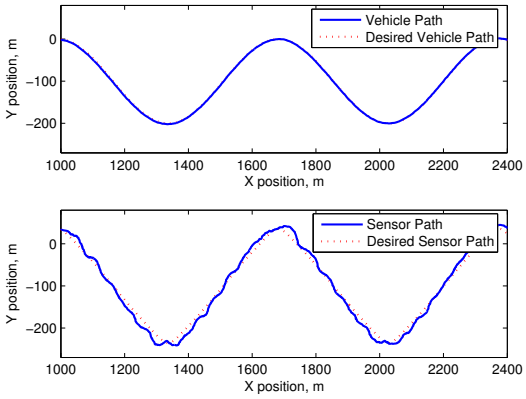

Fig. 3. Spatial sliding mode controller aircraft and sensor paths.

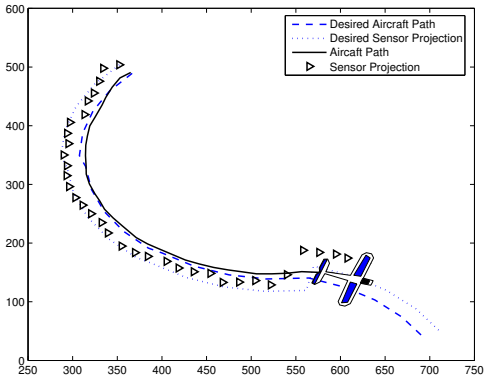

Fig. 5. Sample path and aircraft trajectory.

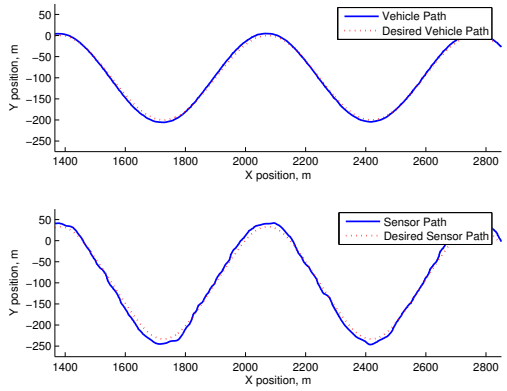

Fig. 4. Kinodynamic controller aircraft and sensor paths.

B. Flight Results

Flight tests were performed at Camp Roberts, CA, in February of 2008. All experiments were conducted on our Sig Rascal 110 platform [9]. We examine the results of the spatial sliding mode controller followed by the receding horizon controller. Then, we compare the performance of the two controllers. The high-level planning algorithm develops paths at 2 second intervals, with a 10 second planning horizon. These paths are piecewise continuous, with piecewise continuous first and second derivatives. The paths generated by the planning algorithm are feasible with respect to the vehicle kinetic constraints and current mean wind values; the path generated at time t_0 starts at $(x(t_0), y(t_0))$ and is parallel to the aircraft velocity vector.

Fig. 5 shows desired and actual sensor and vehicle paths for a sample trajectory, using the spatial sliding mode controller. While ten second paths are computed by the planner, only two seconds of each desired path are shown.

The kinodynamic controller runs in two modes: path generation, and optimal path selection. In the path generation stage, the first iteration generates 300 paths randomly and uniformly sampled from the configuration space. The optimal path in this set is perturbed slightly in the second iteration of the controller, generating 200 new paths. The kinodynamic controller was also run at 10 Hz with a receding horizon of 9 seconds.

For the sliding mode controller, average sensor path error was 9.2 meters with a standard deviation of 7.2 meters. For the kinodynamic controller, the average sensor path error was 7.5 meters, with a standard deviation of 7.2 meters. From the current experiments, we observed that the tracking performance of the sliding mode controller is comparable to that of the receding horizon controller. One may expect a controller that attempts to place a sensor footprint without explicitly accounting for roll dynamics not to perform well. However, at the path planner level and in the context of the unicycle model, the position coordinates are deterministically related to roll and yaw. Therefore, by minimizing spatial error, the sliding mode controller indirectly attempts to place the sensor footprint on the desired trajectory, while the receding horizon controller achieves this directly.

In the case of both controllers, the standard deviation of the tracking error is relatively large. This is directly related to the effects of wind on both controllers. Plots show that upwind tracking performance is significantly better than downwind performance, simply because higher true airspeed results in greater control authority. Additionally, continuously updated estimates of the average wind speed are taken into account in both controllers, but large wind gusts that existed during both flight tests resulted in poor tracking.

IV. CONCLUSIONS AND FUTURE WORK

The goal of this experiment was to design control algorithms for tracking a desired sensor footprint and to compare the effectiveness of the proposed algorithms in real-world framework. A sliding mode controller and a kinodynamic receding horizon controller were developed and flight tested. The sliding mode controller has strong advantages with respect to robustness and ease of computation. The kinodynamic controller presents advantages with respect to flight constraints, such as obstacle avoidance, at the expense of heavy computation. More significantly, for this experiment,

the kinodynamic controller can explicitly account for tracking a desired sensor footprint. In the flight tests, the current forms of both controllers display the ability to accurately place the sensor on a desired trajectory in the presence of model uncertainty and wind uncertainty. Advances in onboard computation make the kinodynamic controller a more appealing choice. While the advantages of the kinodynamic controller are not on display in this experiment, a metric of basic tracking performance is necessary for future work.

In future flight experiments, we plan to better compare the performance of the two controllers by tracking the same path and in similar wind conditions. We will also fly the same flight path in HIL simulations to better quantify the deviation from ideal performance for each controller. We hope to gain better performance by incorporating a more accurate aircraft and wind model in both the path planner and sliding mode controller. We also would like to develop model predictive control for sensor path tracking and compare the results to the other control algorithms developed.

REFERENCES

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "A hierarchical model predictive control framework for autonomous vehicles," in *IEEE American Controls Conference*, 2008.

[2] J. Tisdale, Z. Kim, and J. K. Hedrick, "An autonomous system for cooperative search and localization using unmanned vehicles," in *Submitted to the AIAA Guidance, Navigation and Control Conference*, 2008.

[3] Y. Kuwata and J. How, "Receding horizon implementation of MILP for vehicle guidance," in *IEEE American Controls Conference*, 2005.

[4] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Prentice Hall, 1991.

[5] T. G. McGee and J. K. Hedrick, "Path planning and control for multiple point surveillance by an unmanned aircraft in wind," in *Proceedings of the American Controls Conference*, 2006.

[6] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. Journal of Robotics Research*, vol. 18, pp. 233–255, 2002.

[7] M. K. Masten, "Inertially stabilized platforms for optimal imaging systems tracking dynamic targets with mobile sensors," *IEEE Control Systems Magazine*, vol. 28, pp. 47–64, 2008.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Department, Iowa State University, Research Report TR 98-11, 1998.

[9] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, S. Spry, R. Sengupta, and J. Hedrick, "A distributed control system for unmanned aerial vehicles executing collaborative sensing missions using multi-step plans," in *American Controls Conference*, 2007.