Hierarchical Adaptive Planning in Environments with Uncertain, Spatially-Varying Disturbance Forces

Vishnu R. Desaraju and Nathan Michael

Abstract— **This paper presents a hierarchical planning architecture that generates vehicle trajectories that adapt to uncertain, spatially-varying disturbance forces toward enhanced tracking performance. The disturbance force is modeled as a discrete conditional probability distribution that is updated online by local measurements as the vehicle navigates. A global planner identifies the optimal route to the goal and adapts this route according to a cost metric derived from the belief distribution on the disturbance force. A local planner embeds the belief distribution in the trajectory generation process to compute dynamically feasible trajectories along the global plan that evolve with the belief. Simulation studies analyze and demonstrate the increased trajectory tracking accuracy via the proposed methodology with a single vehicle and the impact of the approach to multiple agents performing collaborative inference toward enhanced collective performance.**

I. Introduction

The success of an autonomous micro air vehicle (MAV) executing a complex mission hinges on its ability to plan trajectories that accomplish the mission objectives and track them accurately. For example, a persistent surveillance mission requires a MAV or team of MAVs to make repeated trips through an environment to maintain situational awareness. The trajectories are planned around sensor constraints, so poor tracking may lead to constraint violation and degrade the information collected [1].

However, MAVs must often operate in environments with significant disturbances to their motion stemming from bulk fluid flow due to pressure differentials, parasitic drag, ground effect, and other unmodeled external forces acting on the vehicle [2]. This is especially challenging for vehicles with limited capabilities, such as MAVs, that prevent them from simply rejecting the effects of disturbances through feedback control. Using robust or adaptive control to follow a trajectory is insufficient as the effects of the disturbance force may warrant selecting an alternate trajectory that can be tracked more accurately or require less energy usage (e.g., by avoiding turbulent regions or exploiting a tailwind [3]). Therefore, a MAV must be able to adapt its trajectories to the effects of disturbances to operate reliably in these environments.

Disturbances can vary in indoor and outdoor environments due to uncertain air flow sources (e.g., HVAC systems, wind flow). While many planning algorithms consider the effects of external disturbances [1, 4, 5], they typically are not amenable to fast replanning. Several online planners consider the vehicle's dynamics when generating plans to guarantee path feasibility [6, 7]. However, a spatially-varying disturbance force results in a time-varying vehicle model, making it impractical to use methods that rely on precomputed libraries or lookup-tables.

Robust planning algorithms select paths based on the disturbances in the environment. Deterministic approaches consider disturbance-invariant sets that effectively yield safety tubes about a nominal path [8, 9]. Other techniques plan for stochastic disturbances using metrics based on uncertainty to inform their path cost [10, 11] or the exploration process [12, 13]. However, these approaches are primarily focused on generating paths that will take the vehicle to the goal, rather than accurately tracking the path.

This highlights the need for an online, predictive path planning algorithm that can account for vehicle dynamics and uncertain disturbance forces in order to accurately follow the computed trajectory. Therefore, we propose a hierarchical adaptive planning architecture that estimates online the disturbance force and uses this estimate to drive two planners: a global planner that guides the vehicle to the goal and avoids routes that will incur a high cost due to disturbances, and a local planner that uses the disturbance estimates to generate dynamically feasible trajectories about the global plan that the vehicle can follow accurately.

The Closed-loop Rapidly-exploring Random Tree (CL-RRT) algorithm is particularly well-suited to generating dynamically feasible local trajectories, as it incorporates a model of how the closed-loop system evolves over time and uses forward-simulation to provide guarantees on path feasibility [14]. Its ability to handle model uncertainty has also been studied [10, 15]. However, we take an alternate approach based on the observation that CL-RRT explores the reachable space for a dynamic system [16]. By introducing disturbance estimates in the forward-simulation process, we generate a mapping between the reference space of the closed-loop system and its reachable space in the presence of external disturbances. The algorithm verifies that any trajectory generated will satisfy all constraints given the current estimate of the external disturbances in the environment.

II. Approach

We consider the problem of planning trajectories through an environment in the presence of a spatially-varying disturbance force acting on a MAV or team of MAVs. To satisfy mission-level requirements, we wish to select trajectories that the MAV can execute accurately. Consequently, the planning

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {rajeswar, nmichael}@cmu.edu
We gratefully acknowledge the support of ARL Grant W911NF-08-2-0004.

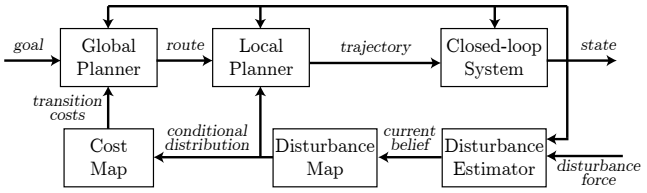


Fig. 1. System architecture showing the hierarchical planner driven by the disturbance force belief.

objective is not only to find trajectories to the goal, but also to select trajectories that maximize tracking accuracy. These two objectives motivate the use of a hierarchical adaptive planning architecture. This architecture, outlined in Fig. 1, enables the MAV to adapt its trajectory in response to an online updated conditional probability distribution on the disturbance force.

As the MAV moves through the environment, it can estimate the disturbance force it experiences based on its motion [17] or using measurements from onboard sensors, such as artificial hair cells [18]. These measurements drive an estimator (e.g., a Kalman filter) to maintain a belief distribution for the current disturbance force, $\tilde{D} \sim N(\mu_D, \Sigma_D)$. This Gaussian model is motivated by its use in [17], although this merits further study, as discussed in Sect. IV. The current belief is used to update a discrete conditional probability distribution representing the uncertainty in the disturbance force, as described in Sect. II-B. This continuously updated disturbance belief map enables the MAV to plan trajectories more intelligently by considering the effects of disturbance forces on the performance of system.

A global planner computes a nominal trajectory through the environment to guide the MAV to a specified goal location. It adapts to the disturbance force by augmenting its normal cost metric with a cost map generated from the disturbance belief map, as described in Sect. II-C.1. The local planner in Sect. II-C.2 is a modified CL-RRT planner. It evaluates the effects of the disturbance force (based on the current belief) on the motion of the MAV to compute dynamically feasible trajectories about the global trajectory. As the conditional distribution is updated over time, both planners will adapt to the new information and generate different trajectories that can be tracked more accurately.

A. Vehicle Model

To simplify the presentation of this planning architecture, we consider a quadrotor MAV with mass m and inertia J constrained to the world y-z plane and subject to an additive disturbance force, as shown in Fig. 2. The equations of motion in terms of position $p = [y, z]^T$, pitch angle θ, and the disturbance force $D = [d_y, d_z]^T$ are

$$\begin{aligned} m\ddot{y} &= -F\sin\theta + d_y \\ m\ddot{z} &= F\cos\theta - mg + d_z \\ J\ddot{\theta} &= M \end{aligned} \tag{1}$$

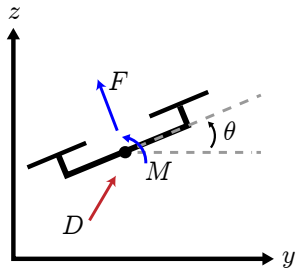


Fig. 2. The state space configuration for a 2-D quadrotor in the y-z plane with pitch angle θ, inputs F and M, and external disturbance force D.

The total thrust F and pitch moment M are inputs that can be mapped back to the speeds of the motors ω_i [19],

$$\begin{bmatrix} F \\ M \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ l & -l \end{bmatrix} \begin{bmatrix} k_F\omega_1^2 \\ k_F\omega_2^2 \end{bmatrix} \tag{2}$$

We consider the vehicle motor dynamics via a saturated first-order model of the motor RPM for a commanded ω_{d_i},

$$\dot{\omega}_i = -k_m(\omega_i - \omega_{d_i}), \qquad \omega_{min} \le \omega_i \le \omega_{max} \tag{3}$$

The parameters l, k_F, and k_m are determined via experimental characterization [19] and are platform specific.

A nonlinear backstepping controller [20] enables a quadrotor to track a smooth reference trajectory $p_d(t)$ with the thrust computed via:

$$T_d = -k_p(p - p_d) - k_v(\dot{p} - \dot{p}_d) - mge_2 + m\ddot{p}_d \tag{4}$$

where e_2 is a unit vector along the world z-axis. The desired thrust defines a reference pitch angle θ_d for the inner loop controller to track, and the inputs F and M are given by

$$\begin{aligned} F &= [-\sin\theta \;\; \cos\theta]T_d \\ M &= -k_R(\theta - \theta_d) - k_\Omega(\dot{\theta} - \dot{\theta}_d). \end{aligned} \tag{5}$$

B. Spatial Disturbance Model

The key component that allows the hierarchical planner to adapt online is the model of the disturbance force. We model the true disturbance force in a given environment as an uncertain, spatially-varying process, and the net disturbance forces observed by the vehicle at a set of locations in the environment are samples drawn from this process. Adapting the vehicle's trajectory to the true disturbance field requires reconstructing this process from these samples.

To do so, we represent the process as a discrete conditional probability distribution, which can be viewed as a grid map of local distributions. Although this is an approximation of the continuous process, it provides a means of identifying the true distribution in each grid cell, and the resolution of the grid map can be selected to control the fidelity of the discrete model to the continuous process. The conditional distribution is updated in a distributed fashion with each grid cell maintaining a local estimate of the disturbance force using a Kalman filter. The evolution of the disturbance force in a cell is modeled as

$$D_{k+1} = D_k + w_k, \qquad w_k \sim N(0, \Sigma_{grid}),$$

where the covariance Σ_{grid} defines the rate at which the uncertainty about old estimates increases.

As the vehicle moves, the current estimate of D from the disturbance estimator serves as a measurement for the Kalman filter corresponding to its current position in the grid. A simple measurement model

$$z_k = \mu_{D_k} + v_k, \qquad v_k \sim N(0, \Sigma_{D_k}).$$

uses both the mean and the covariance of $\tilde{D}$ to update the local estimate.

A prior for the conditional probability distribution can often be constructed from earlier missions, environment geometry, or other scenario-specific information [21]. This will guide the MAV's initial trajectory, while new disturbance estimates update and refine the distribution.

C. Online Adaptive Planning

1) Global Planner: The main purpose of the global planner is to guide the local planner toward the goal by taking into account obstacles or other aspects of the environment that are beyond the local planner's horizon. In this work, we employ A* search to identify a path to the goal and fit a polynomial spline to it using an unconstrained quadratic program [22]. This produces a nominal trajectory with continuous derivatives to better guide the local planner.

The global planner's interaction with the disturbance force is through a cost map derived from the discrete conditional probability distribution. In order to avoid routes with strong disturbances, we define the cost map to be the magnitude of the mean of the disturbance estimate at each grid cell. We then smooth the cost map via a discrete 2-D convolution with a Gaussian kernel [23] to avoid sharp changes in cost between cells.

2) Local Planner: We wish to ensure dynamic feasibility of the local trajectory for accurate tracking, making the Closed-loop RRT (CL-RRT) algorithm a prime candidate [14]. CL-RRT quickly generates trajectories by randomly sampling reference points and simulating the closed-loop dynamics of the system toward them to grow a tree of trajectories. This simulation guarantees that all trajectories will be feasible for the dynamic model and satisfy all constraints by construction [15].

We include the effects of the disturbance force in the dynamic model so that each branch represents the expected trajectory of the vehicle subject to the current disturbance force belief. The disturbance force at each simulation step is sampled from the current grid cell's distribution, $\tilde{D} \sim N(\mu_D, \Sigma_D)$. As a result, a branch grown through a cell with high covariance will be subject to a greater variety of disturbances, reflecting the uncertainty about the disturbance force at that location.

In each planning iteration, we grow a new tree to explore locally around the global trajectory since adding this spatially-varying disturbance produces a time-varying system model. The trajectory returned by CL-RRT is only selected if it has a lower cost than the current trajectory re-simulated with the updated disturbance belief. This prevents the planner from selecting higher-cost trajectories that may be found due to random sampling and re-growing the tree.

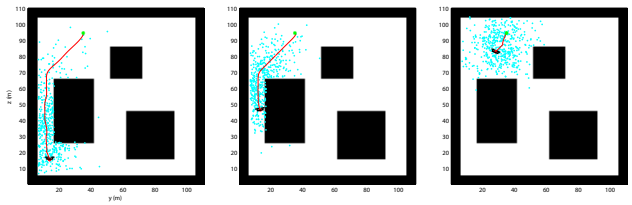


Fig. 3. CL-RRT sampling distribution biased about the global A* trajectory

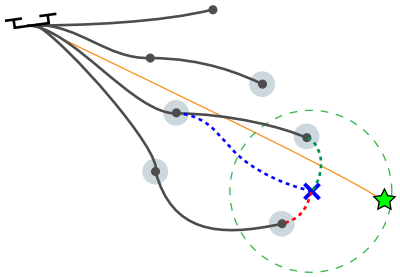


Fig. 4. Standard CL-RRT connects to the nearest-neighbor (red) while RRT* connects to the min-cost node within the radius drawn (green). We connect to the min-cost node of the k-nn (blue with neighbors highlighted) to generate smooth branches that follow the global trajectory (orange).

3) Global and Local Interaction: The global planner generates a proposal trajectory for the vehicle to reach the goal from its current position. However, to ensure trajectory feasibility, the local planner must generate the actual reference that is sent to the controller. To explore feasible deviations from the global trajectory, we bias the sampling distribution for the CL-RRT as shown in Fig. 3. Samples are drawn from a Gaussian distribution whose mean follows a half-normal distribution along the global trajectory. The Gaussian focuses the search for feasible trajectories around this trajectory [24], while the half-normal distribution prevents CL-RRT from planning too far ahead with the current belief.

To reduce the dimension of the search space, we sample from the space of differentially flat outputs, $[y, z]$, rather than the full reference space of the controller, $[y, z, \dot{y}, \dot{z}, \ddot{y}, \ddot{z}]$. The sample is then augmented with derivatives taken from the projection of the sample onto the global trajectory [25]. This produces branches that try to follow the global trajectory in position, velocity, and acceleration.

RRT-based planners are also known to generate trajectories with extraneous excursions simply due to the nature of exploration through random sampling. However, the local planner only needs to explore the region around the global trajectory and should plan trajectories that follow it. Therefore, we grow the branch from the lowest cost node of the k-nearest neighbors (k-nn) to the sample point, as shown in Fig. 4. This modified node selection approach produces smooth, lower-cost trajectories that follow the global trajectory by avoiding expensive excursions.

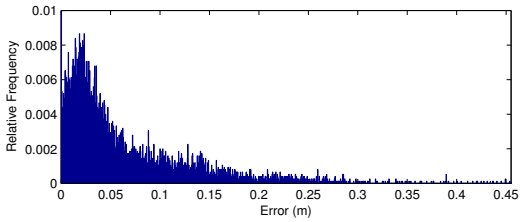
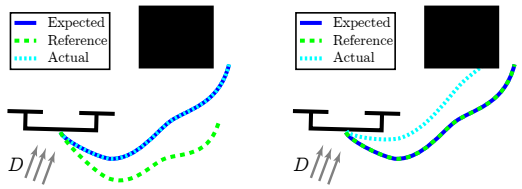
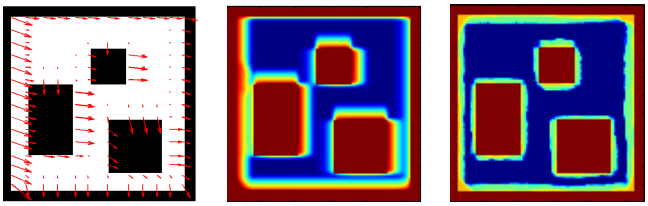


Fig. 5. Distribution of the error between a CL-RRT trajectory and its corresponding spline showing the accuracy of the spline fit.



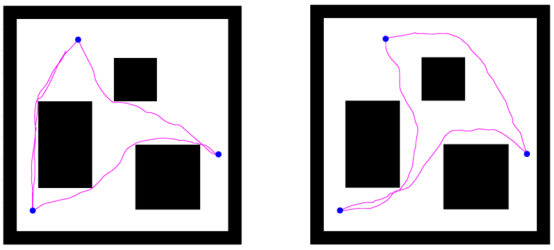
(a) (b)

Fig. 6. (a) The proposed reference trajectory accounts for expected disturbances. (b) Ignoring expected disturbances in reference trajectory generation can result in infeasible trajectories.



(a) True disturbance (b) True cost map (c) Cost map prior

Fig. 7. True disturbance force and corresponding cost maps for the true and prior belief distributions. The color gradient from red to blue indicates regions of high to low cost.



(a) No information case (b) Perfect information case

Fig. 8. Example trajectories showing the routes taken to the three targets (blue dots) in the two limiting cases.

4) Polynomial Trajectory Representation: CL-RRT produces a set of waypoints for the vehicle to follow [14, 26] and not a smooth trajectory to track, e.g., using (5). Therefore, we generate a smooth trajectory by fitting a spline to the CL-RRT waypoints [22]. We preserve feasibility by constraining the derivatives at each waypoint using state information from CL-RRT branches. This produces a smooth trajectory that accurately tracks the CL-RRT trajectory, as shown in Fig. 5. Feasibility is verified by re-simulating the spline.

Including disturbances in CL-RRT may lead to a significant difference between the reference and the expected trajectories. Therefore, we re-simulate the reference without disturbances and use the resulting trajectory for the spline fit. Figure 6 illustrates the difference between tracking this new reference and tracking the expected trajectory. CL-RRT can also embed mission-level constraints in its plans, making accurate tracking essential for feasibility. Therefore, this approach minimizes tracking error, defined as the distance between the actual trajectory and the expected trajectory from CL-RRT.

III. Simulation Results

To demonstrate the utility of this hierarchical adaptive planning architecture, we consider a scenario in which the 2-D quadrotor must continuously fly between three known targets (e.g., for persistent surveillance). The targets are spread across a large environment with obstacles and an uncertain, spatially-varying disturbance force (Fig. 7(a)). The true disturbance force yields the corresponding cost map in Fig. 7(b). The planner is provided with a prior on the disturbance force belief, and the corresponding cost map is shown in Fig. 7(c).

A. Single Vehicle Performance

We first consider a single quadrotor operating in this environment. The effects of this adaptive planning architecture are most evident when comparing the limiting cases of planning with no information and with perfect information about the disturbance force. In the first case, the quadrotor plans using A* and CL-RRT without any adaptation to the disturbances and selects the shortest paths between targets, as shown in Fig. 8(a). With perfect information, A* avoids high-disturbance regions, as seen from a comparison of Fig. 7(b) and Fig. 8(b), while CL-RRT uses the disturbance force belief to generate reference trajectories that the quadrotor can track well.

Figure 9 shows snapshots of the 2-D quadrotor flying between the three targets using the prior in Fig. 7(c). The top image in each snapshot shows the current global trajectory (red), the CL-RRT (yellow), the biased sampling distribution (cyan), the local spline trajectory (green), and quadrotor's actual trajectory (magenta). The quadrotor initially follows the same route as in the no-information case, since the prior underestimates the disturbance force. However, as the belief is updated, the planner adapts and alters its trajectories to follow routes that are believed to have weaker disturbance forces. For example, in Fig. 9(b), the quadrotor deviates from the shortest path and flies under the lower-right obstacle, which is believed to be a low-cost region. It avoids the known regions of high cost, leading to it exploring different routes suggested by the current belief. It eventually explores all the viable routes and settles on trajectories through low-disturbance regions as determined by the updated belief.

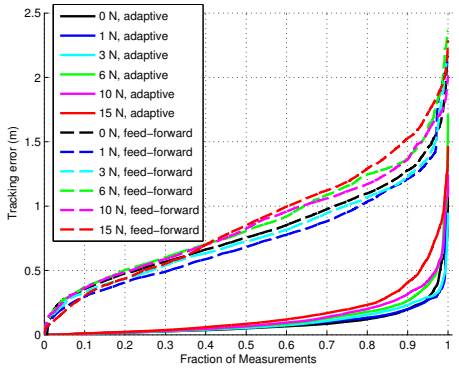
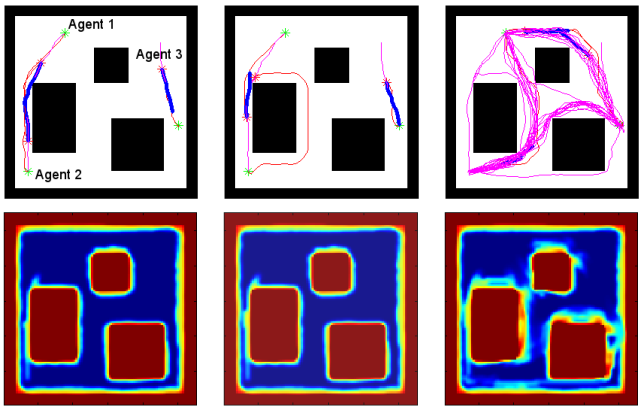


Fig. 10. Tracking error for the adaptive planner remains well below the feed-forward approach, even as the disturbance force magnitudes vary.



(a) (b) (c)

Fig. 11. Snapshots of the three quadrotor scenario. (a) The agents on the left both select shortest paths. (b) The updates to the disturbance map from Agent 2 prompt Agent 1 to replan to avoid the high-disturbance region. (c) The team converges to similar routes as in the single agent case.

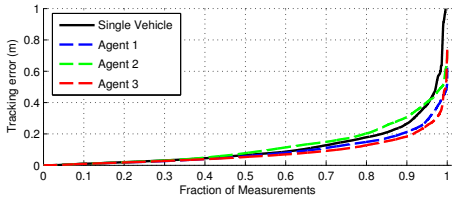


Fig. 12. Tracking error is reduced for two quadrotors in the multi-agent case. The third is comparable to the single agent case.

Figure 9(e) shows that once it has explored the map and estimated the disturbance force along each route, the trajectories follow the routes shown in Fig. 8(b).

As described earlier, the key metric for this adaptive planning architecture is tracking error, due to the importance of following the expected trajectory. To illustrate its performance, we compare it to an approach that tracks the global A* trajectory with a feed-forward term in the controller to compensate for the current disturbance force belief. Since the quadrotor is expected to follow the global trajectory, tracking error is defined as the deviation from the smoothed A* trajectory. Figure 10 shows the percentage of tracking error measurements below a given threshold for the scenario in Fig. 9. The shallow slope for the adaptive planner illustrates its ability to track the expected trajectory much more accurately due to its use of CL-RRT to predict the effects of the disturbance. Accuracy is also more consistent than in the feed-forward case as the maximum magnitude of the disturbance force is increased. The non-adaptive planner from Fig. 8(a) is unable to avoid obstacles for a maximum disturbance force greater than 1 N.

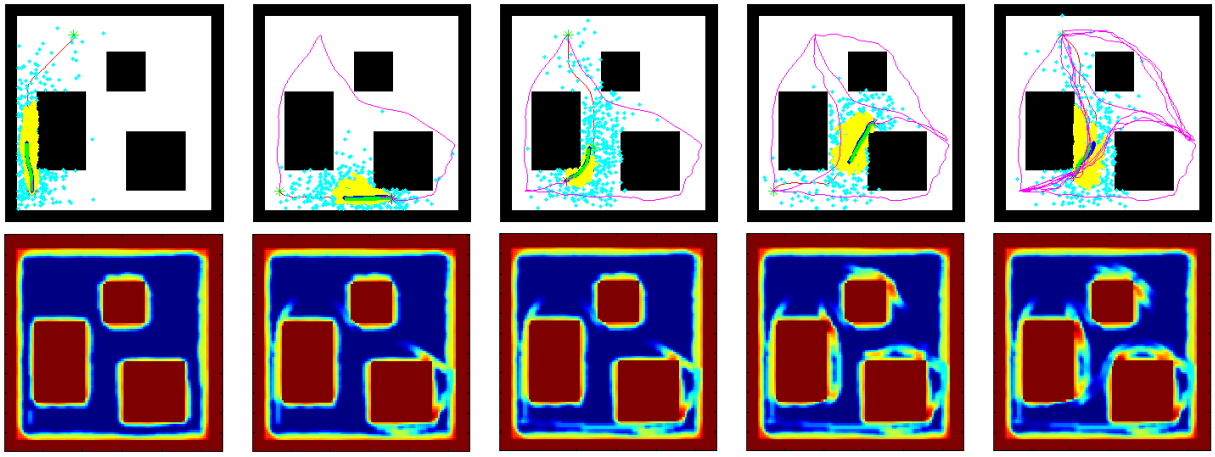
B. Multi-Agent Performance

This adaptive planning architecture is particularly effective when applied to a multi-agent team. For simplicity, we assume the agents form an ideal network and that the disturbance map is centrally managed. Inter-agent interactions can be accounted for via a coordination strategy [27]. Figure 11 highlights a three-quadrotor mission scenario in which Agent 1 and Agent 2 initially must switch positions and select the shortest paths to do so. However, Agent 2's updates to the disturbance map inform Agent 1's global planner to reroute and avoid a high-disturbance region without having to traverse it. This enables the team to quickly identify the best routes without multiple agents exploring each one and improves tracking accuracy, as shown in Fig. 12.

IV. Conclusions and Future Work

In this work, we have presented a hierarchical adaptive planning architecture that allows one or more micro air vehicles to plan trajectories that can be tracked accurately in the presence of an uncertain, spatially-varying disturbance force. The discrete conditional probability distribution for the disturbance force is updated online with local observations as the vehicle traverses the environment. This enables the global planner to select low-disturbance routes, while the CL-RRT local planner ensures dynamic feasibility according to the accuracy of the conditional distribution. The simulation studies demonstrate the advantages of adapting at the planner level, rather than compensating for the disturbance force at the controller level.

We are presently working toward an experimental hardware evaluation of the methodology which necessitates a 3-D real-time implementation on a mobile class processor. Current results empirically suggest a two- or three-fold increase in run-time complexity for an analogous high-fidelity 3-D model [2], thereby restricting the rate of online re-planning to 1 Hz or less. While these informal remarks suggest that real-time evaluation is viable, we are interested in pursuing a more precise characterization of the computational complexity of the proposed approach. We are also interested in investigating alternate representations of the conditional distribution that more accurately represent the spatially-varying disturbance force and can capture spatial correlation and variation over time, permitting the planner to propagate its current belief to enhance its motion prediction and reduce the dependence on the prior.



(a) (b) (c) (d) (e)

Fig. 9. Snapshots of the 2-D quadrotor flying between three targets (first row) while updating its disturbance force belief (second row). (a) The quadrotor initially takes the shortest path to the first target. (b) The belief is updated with the disturbances observed, biasing the trajectories toward routes believed to be low-disturbances. (c) To re-visit the first target, it takes a different route to avoid the known high-disturbance region. (d) Similarly, it explores different routes to the other targets while continuing to update the distribution. (e) Subsequent trajectories follow the routes found to have weaker disturbance forces.

REFERENCES

[1] N. Ceccarelli, J. J. Enright, E. Frazzoli, S. J. Rasmussen, and C. J. Schumacher, "Micro uav path planning for reconnaissance in wind," in *Proc. of the Amer. Control Conf.*, New York City, NY, July 2007.

[2] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor," in *Proc. of the Australasian Conf. on Robot. and Autom.*, Wellington, New Zealand, Dec. 2012.

[3] R. L. McNeely, R. V. Iyer, and P. R. Chandler, "Tour planning for an unmanned air vehicle under wind conditions," *AIAA J. Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1299–1306, Sept.

[4] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Barcelona, Spain, Apr. 2005, pp. 194–198.

[5] A. L. Jennings, R. Ordonez, and N. Ceccarelli, "Dynamic programming applied to UAV way point path planning in wind," in *Proc. of the IEEE Intl. Conf. on Comp.-Aid. Control Syst.*, San Antonio, TX, Sept. 2008, pp. 215–220.

[6] P. Reist and R. Tedrake, "Simulation-based LQR-trees with input and state constraints," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Anchorage, AK, May 2010, pp. 5504–5510.

[7] M. Pivtoraiko and A. Kelly, "Kinodynamic motion planning with state lattice motion primitives," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, San Francisco, CA, Sept. 2011, pp. 2172–2179.

[8] D. Q. Mayne, M. M. Seron, and S. V. Rakovi, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, Feb. 2005.

[9] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Proc. of the Intl. Workshop on the Algorithmic Foundations of Robot.*, Cambridge, MA, June 2012.

[10] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Proc. of the AIAA Guidance, Navigation, and Control Conf.*, Boston, MA, Aug. 2013.

[11] T. Schouwenaars, B. Mettler, E. Feron, and J. P. How, "Robust motion planning using a maneuver automation with built-in uncertainties," in *Proc. of the Amer. Control Conf.*, Denver, CO, June 2003.

[12] N. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Rome, Italy, Apr. 2007, pp. 1617–1624.

[13] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 723–730.

[14] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sept. 2009.

[15] B. D. Luders, S. Karaman, E. Frazzoli, and J. P. How, "Bounds on tracking error using closed-loop rapidly-exploring random trees," in *Proc. of the Amer. Control Conf.*, Baltimore, MD, June 2010.

[16] G. S. Aoude, J. P. How, B. D. Luders, and T. E. Pilutti, "Sampling-based threat assessment algorithms for intersection collisions involving errant drivers," in *Proc. of the IFAC Sym. on Intell. Auton. Vehicles*, Lecce, Italy, Sept. 2010.

[17] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011.

[18] M. M. Sadeghi, R. L. Peterson, and K. Najafi, "A 2-D directional air flow sensor array made using stereolithography and MEMS micro-hydraulic structures," in *Intl. Conf. on Transducers & Eurosensors*, Barcelona, Spain, June 2013, pp. 722–725.

[19] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "Experimental evaluation of multirobot aerial control algorithms," *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, Sept. 2010.

[20] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. of the IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 5420–5425.

[21] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Proc. of the Intl. Sym. on Exp. Robot.*, Quebec City, Canada, June 2012, pp. 289–302.

[22] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Proc. of the Intl. Sym. of Robot. Research*, Singapore, Dec. 2013.

[23] L. G. Shapiro and G. Stockman, *Computer Vision*. Prentice Hall, 2001.

[24] L. Krammer, W. Granzer, and W. Kastner, "A new approach for robot motion planning using rapidly-exploring randomized trees," in *IEEE Intl. Conf. on Industrial Informatics*, Caparica, Portugal, July 2011.

[25] M. J. V. Nieuwstadt and R. M. Murray, "Real time trajectory generation for differentially flat systems," *Intl. J. Robust & Nonlinear Control*, vol. 8, no. 11, pp. 992–1020, Sept. 1998.

[26] D. Levine, B. Luders, and J. P. How, "Information-theoretic motion planning for constrained sensor networks," *J. Aero. Inf. Syst.*, vol. 10, no. 10, pp. 1–26, Oct. 2012.

[27] V. R. Desaraju and J. P. How, "Decentralized path planning for multi-agent teams with complex constraints," *Auton. Robots*, vol. 32, no. 4, pp. 385–403, Feb. 2012.