Adaptive Receding Horizon Control for Vision-Based Navigation of Small Unmanned Aircraft

Eric W. Frew, *Member, IEEE*, Jack Langelaan, *Student Member, IEEE*, Sungmoon Joo, *Student Member, IEEE*

Abstract— **This paper presents an integrated vision-based navigation system for small autonomous aircraft. Previously developed sensor fusion algorithms based on the Unscented Kalman Filter (UKF) are combined with an adaptive receding horizon controller (ARHC) for guidance. Control and planning horizons are computed based on the sensor range and the effective speed of the UAV, which is computed as a weighted sum of estimated vehicle speed and time rate of change of the uncertainty of the obstacle position estimates. Simulation results demonstrate the integrated system and illustrate the value of the adaptive approach.**

I. Introduction

Successful operation of small unmanned aircraft (wingspan<5 feet) in obstacle strewn environments (such as mountainous terrain, forests, or urban areas) requires integrated sensing and control strategies for obstacle detection, obstacle avoidance and navigation that consider the complexities of the entire UAV system. Small UAVs undergo 6 degree of freedom motion, are subject to significant external disturbances, require high bandwidth control, have limited on-board sensing due to small payload capacities, and operate in a highly dynamic environment.

Fusing computer vision data with measurements from inexpensive inertial measurement units (IMUs) provides a navigation solution that satisfies the limited payload and energy capacities of small unmanned aircraft, provides an information-rich stream of data which is useful to a human operator, can be used for obstacle detection, and can be used in cases where GPS is unavailable due to obstructions or jamming [1, 2]. While stereo camera configurations provide both range and bearing to features, vehicle dimensions impose limits on the baseline between cameras and thus range accuracy. By using a monocular camera one can obtain bearings to fixed landmarks. By fusing bearings to these fixed landmarks (whose positions may be initially unknown) with data from the low-cost IMU we can obtain a localization solution both for the UAV and for the landmarks. This is known as Simultaneous Localization and Mapping (SLAM), sometimes referred to as Concurrent Localization and Mapping.

In contrast to many SLAM applications that use both range and bearing measurements (e.g. [3, 4]) the vision system in the application considered here only provides a bearing measurement to each landmark. This provides unique challenges to the sensor fusion system, including: limited observability of states, which results from having only inertial and bearing measurements available; increased effect of non-linearities in the measurement model, which results from the close proximity of the vehicle to the objects; and finally the significant uncertainty in predicted vehicle state, which results both from the measurement noise and drift error induced by the low cost IMU and the likelihood of significant external disturbances. In previous work we described an implementation based on an Unscented Kalman Filter (UKF), demonstrating successful mapping and navigation through an obstacle-strewn environment [2].

The limited range and field of view of the vision sensor combined with significant uncertainties in the vehicle and landmark states suggest the use of receding horizon control (RHC) strategies for vehicle guidance. Receding horizon control frameworks have become popular for control of unmanned aerial vehicles, especially in the context of obstacle avoidance [5-7] because they offer built-in mechanisms for responding to a dynamic environment. However, these approaches have been employed for UAV trajectory generation and navigation by assuming obstacles are sensed immediately, with no error, upon entering a local detection region or shell.

Receding horizon control can be augmented to include the output of vision-based estimation [6, 7]. This can consist of a learning algorithm that combines feature points into a terrain map of the environment [6] or inclusion of the estimate covariance matrices as extended obstacle boundaries [7]. In previous work we developed an RHC controller for passive, non-cooperative see-and-avoid that was independent of the specific implementation of the vision-based estimation [7].

In this work we combine previously developed vision-

Eric W. Frew is an Assistant Professor in the Aerospace Engineering Sciences Department at the University of Colorado, Boulder. (e-mail: eric.frew@colorado.edu).

Jack Langelaan is a graduate student at Stanford University. (e-mail: jackl@stanford.edu).

Sungmoon Joo is a graduate student at Stanford University. (e-mail: joosm@stanford.edu).

based estimation and RHC control algorithms to produce a complete navigation system. The main contribution in this work is the introduction of an adaptive planning horizon. The planning horizon is computed based on an effective velocity which incorporates vehicle speed and the rate of change of uncertainty of obstacle positions, recognizing that lengthy plans may quickly become invalid when significant information is being obtained about the environment or when vehicle speed is large compared with sensor range.

The remainder of this paper is organized as follows: Section II describes the vision-based navigation problem, Sections III summarizes the adaptive RHC controller respectively, and Section IV provides simulation results.

II. Vision-Based Navigation

A. Navigation Scenario

The navigation scenario consists of navigating a small UAV through an unsurveyed environment such as a forest. An on-board camera provides bearings to obstacles and an inertial measurement unit provides accelerations and angular rates in the body frame.

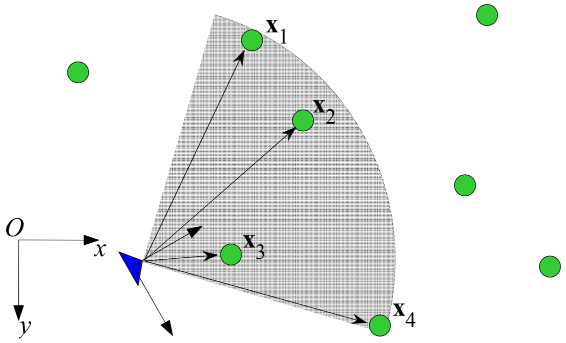

Fig. 1. The navigation scenario. The grey region shows the field of view of the vision system, obstacles are show by green dots.

Trees are located at $\mathbf{x}_i$ in frame O, an inertial frame. The position and orientation of the aircraft is defined by $[x \quad y \quad \psi]^T$, expressed in frame O and $[u \quad v]^T$ represents the velocity of the aircraft in the body frame. The problem is to navigate from a known start position to a known goal position without the aid of absolute position sensors such as GPS and without an a priori map of the forest.

B. System Description

The block diagram in Fig. 2 shows a system that uses the given sensors to perform obstacle avoidance. The vector of states which is estimated includes the vehicle state $\mathbf{x}_v$ (which includes position in the inertial frame; orientation with respect to the inertial frame; velocity expressed in the body frame; and IMU biases) and the positions of the tree trunks $\mathbf{x}_o$ (in the inertial frame).

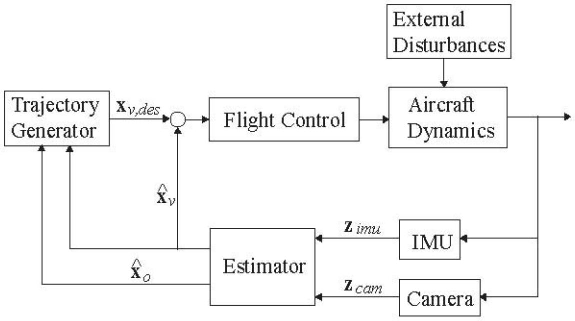

Fig. 2. Navigation system schematic.

IMU measurements $\mathbf{z}_{imu}$ (acceleration and angular rate in the body frame) and camera measurements $\mathbf{z}_{cam}$ (bearings to obstacles) are fused in the estimator, which computes estimates of the vehicle state $\hat{\mathbf{x}}_v$ and obstacle positions $\hat{\mathbf{x}}_o$ along with the associated covariances. These estimates are given to the trajectory generator, which computes a collision free trajectory to be followed by the aircraft. The difference between the desired trajectory $\mathbf{x}_{v,des}$ and the estimated aircraft state is used by the flight controller to generate control inputs.

C. State Estimation

Given noisy, biased measurements of acceleration and angular rate (from the IMU) and noisy measurements of bearings to obstacles (from the vision system), an estimator based on an Unscented Kalman Filter [8, 9] computes vehicle state, obstacle positions and the associated covariance [1]:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_o \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vo} \\ \mathbf{P}_{ov} & \mathbf{P}_{oo} \end{bmatrix} \tag{1}$$

where (for vehicle navigation in a plane)

$$\mathbf{x}_v = \begin{bmatrix} x & y & \psi & u & v & \boldsymbol{\alpha}^T & \mathbf{b}^T \end{bmatrix}^T \tag{2}$$

$$\mathbf{x}_o = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_m^T \end{bmatrix}^T \tag{3}$$

The covariance is a measure of the uncertainty in the estimated states.

As long as measurements to obstacles are available the uncertainty in position estimate is reduced. With this reduction in obstacle position uncertainty more efficient trajectories can be planned (meaning that the vehicle can operate in closer proximity to the obstacle while ensuring that the probability of collision is still small). It is this continuous improvement in obstacle position estimates that provides the motivation for adaptive receding horizon control: planned trajectories may quickly become invalid (or unnecessarily conservative, and thus expensive) as knowledge of the surrounding obstacles improves.

Since only bearing measurements are available, observability of states is highly dependent on motion of the camera (i.e. the trajectory flown by the UAV). During motion directly towards or away from an object (along the bearing) there is no new information which improves range estimate. Transverse motion is required to produce a useful estimate of object position. In the case of obstacle avoidance, transverse motion has the added benefit of ensuring that a collision is avoided, but the presence of multiple obstacles places conflicting demands on the trajectory which must be flown.

D. Adaptive Receding Horizon Control

The trajectory generator uses an adaptive receding horizon control policy to calculate local reference paths through the obstacle field. The main steps of the RHC approach are (i) calculating the optimal control sequence $\mathbf{u}^*[k+i] = \{u^*[k+1], \cdots, u^*[k+T_h]\}$ starting from state $\mathbf{x}_{uas}[k]$, over a finite *planning horizon* $2 \le T_h < \infty$; (ii) implementing the optimal control input over some *control horizon* $k \le t \le k + T_c$ where $1 \le T_c \le T_h$; and (iii) repeating steps 1 and 2 for $\mathbf{x}_{uas}[k+T_c]$ at time $k+T_c$.

The RHC approach is well suited for the vision-based navigation problem posed here because sensor constraints limit information about the environment beyond a certain range. Likewise, the estimated positions of the obstacles will change and the uncertainty in these estimates will decrease as the UAV moves through the environment. Thus it is unlikely that paths that extend significantly beyond the sensor range or that extend too far in time will remain optimal or even feasible (i.e. collision free) as the UAV proceeds. Computational effort spent planning paths beyond this critical time is therefore wasted. Instead, inclusion of an estimate of the cost-to-go beyond the planning horizon directs the local paths toward the goal location.

Even though the RHC approach calculates plans over a finite time horizon, computational cost is a major barrier to implementation on many systems. In order to calculate the optimal control input, the RHC trajectory generator must simulate the dynamics of the system for every candidate output it evaluates. For vision-based navigation in 2D that includes 10 vehicle states and 2*N* obstacle states. In order to minimize the computational effort spent planning useless segments of the local path, the control and planning horizons are adapted to the average uncertainty of the obstacle estimates. When the time rate of change of the average uncertainty is low, likelihood is higher that the world will remain relatively static over the planning horizon. Likewise, a high time rate of change in the average uncertainty implies that the world will change significantly over a shorter planning horizon and thus much of the path will be useless. Shortening the planning horizon during periods of high change in uncertainty also frees more resources for the estimator to process information.

III. Adaptive Receding Horizon Control

UAV guidance is achieved using a receding horizon control (RHC) policy based on the controller described in [7]. A key feature of this RHC policy is the ability to consider the information gain (i.e. reduction in uncertainty) of the obstacle positions over the planning horizon. The method presented here optimizes the control sequence for a low-level flight control system and these inputs are fed directly into the flight control system, eliminating the need for a path or trajectory tracking controller.

A. Adaptive Planning Horizons

The control and planning horizons of the RHC policy are adapted to the state of the aircraft and the time rate of change of the obstacle estimates. A re-plan is triggered by one of two conditions: (a) a new obstacle comes into view; (b) the end of the control horizon is reached.

For this work the control horizon is defined to be a fixed fraction of the planning horizon, i.e. $T_c = f \cdot T_h$ with $0 < f \le 1$.

The appearance of a previously unseen landmark triggers a re-plan of minimum time horizon. Otherwise the planning horizon is computed based on the sensor range and an effective vehicle speed:

$$T_{plan,k} = \max\left(w_1 \frac{r_{sense}}{u_{eff,k}}, \quad w_2 \frac{r_{sense}}{\hat{u}_k} \right) \qquad (4)$$

Here r_{sense} is the range of the vision sensor, $u_{eff,k}$ is an effective speed (defined below), $\hat{u}_k$ is the current estimate of vehicle speed and $w_{1,2}$ are weights with $w_1 \ge 1$ and $0 \le w_2 \le 1$. Choosing $w_1 \ge 1$ gives the UAV extra time to plan motion around obstacles in its vicinity and $w_2 \le 1$ defines the minimum planning horizon using the current estimate of vehicle speed and sensor range.

The effective speed $u_{eff,k}$ is a combination of estimated vehicle speed and a measure of the rate of change of uncertainty in the obstacle position estimate. It is a heuristic based on the realization that the rate of change of uncertainty of an obstacle's position estimate has the same dimension as vehicle speed:

$$u_{eff,k} = \hat{u}_k + w_3 \frac{d}{dt} \sqrt{Tr \mathbf{P}_{o,kk}} \qquad (5)$$

Here $\mathbf{P}_{o,kk}$ is the position error covariance for the obstacles (i.e. the uncertainty in the obstacle position) and w_3 is a weight. Hence effective speed contains both the estimated vehicle speed $\hat{u}_k$ and the average rate of change of the uncertainty of estimated obstacle positions:

it captures both how quickly the vehicle is moving and how quickly the knowledge of the environment is changing. A large rate of change of obstacle position uncertainty implies that the estimated state of the world will be significantly different at the end of the planning horizon and thus the resulting plan will not be optimal that far into the future.

Recall that obstacles are assumed to be stationary. Hence the position estimate and associated covariance of any obstacle that is not currently in view of the sensor will not change: only obstacles in view will contribute to the time rate of change of position covariance. Therefore

$$\frac{d}{dt}\sqrt{Tr\mathbf{P}_{o,kk}} \le 0 \qquad (6)$$

unless a previously unseen landmark appears in view. In other words, information about obstacle positions is not lost. Thus choosing $w_3 \le 0$ ensures that $u_{eff,k} \ge \hat{u}_k$. Clearly choosing $w_3 = 0$ results in a planning horizon which does not adapt to changing landmark uncertainty but plans with minimum horizon if a new landmark appears and maximum horizon if the end of the control horizon is reached.

B. Receding Horizon Control

The aircraft kinematic model used by the planner differs from that used by the estimator. The planner assumes a low-level autopilot provides the ARHC controller with the standard kinematic model:

$$\begin{aligned} \dot{x} &= V \cdot \cos\psi \\ \dot{y} &= V \cdot \sin\psi \\ \dot{\psi} &= u \quad |u| \le \omega_{\max} \end{aligned} \qquad (7)$$

where V is the known aircraft speed, u is the turn rate command, and $\omega_{\max}$ is the maximum allowable turn rate.

The control algorithm presented here uses an optimization designed to minimize the cost

$$J_{rhc}\left(\hat{\mathbf{x}}_v, \mathbf{u}, \hat{\mathbf{x}}_o\right) = J_u + J_{nav} + J_{unc} + J_{safe} \qquad (8)$$

where $\hat{\mathbf{x}}_v$ is the estimate of vehicle state and $\hat{\mathbf{x}}_o$ is the estimate of the obstacle locations at time *k*.

Components of J_{rhc} include: the control cost J_u, which limits control input; the navigation cost J_{na}, which encourages the UAV to move towards its current goal; the uncertainty cost J_{unc}, which encourages the UAV to improve its knowledge of the state of the N already observed obstacles in its environment; and finally the safety cost J_{safe}, which enables the RHC controller to perform obstacle avoidance [7].

In order to predict the evolution of the target estimate we assume an idealized (efficient) version of the bearings-only sensing problem. We use the Fisher Information Matrix (FIM), which is equivalent to the inverse of the Posterior Cramer-Rao Lower Bound, to describe the information gained about an obstacle by assuming the target estimate is exact over the finite planning horizon and we know exactly how the aircraft moves. The covariance of the aircraft position is added to the covariance of each obstacle estimate in order to focus on gaining information about the obstacles. The future covariance matrix is approximated from the FIM J_F by

$$\overline{\mathbf{P}}[k+i] = \mathbf{J}_F^{-1}[k+i]. \qquad (9)$$

See [7] for more details on calculation of the Fisher Information Matrix for this controller and on solution of the receding horizon optimization using a random search technique.

IV. Simulation Results

This section presents simulation results investigating effects of varying the parameters $(w_1 \quad w_2 \quad w_3)$, which define the maximum, minimum and adaptive time horizons, respectively. For all cases the control horizon is fixed at 30% of the planning horizon. A randomized path planner is used for RHC [7], therefore 50 runs were performed for each case and averages are reported below.

All simulations were conducted on a dual processor 3.0GHz Xeon PC using MatLab v7. A forest of 100 trees randomly distributed over a $280m \times 200m$ area was used for all simulations. The vehicle start position was at (-40,0), the goal location was at (40,0). Vehicle speed was 10m/s, maximum turn rate was 1 rad/s (this represents a 45 degree bank angle for an aircraft in coordinated flight and results in a minimum turn radius of 10m). Sensing range on the camera was 50m, field of view was 100 degrees. The mission is flight to the goal region (a circle of radius 5m around goal) followed by return to start position.

A. Constant planning horizon

Results for constant planning horizons (i.e. $w_1 = w_2$ and $w_3 = 0$) are given in Table 1. As the planning horizon increases the total number of plans required to complete the mission is reduced but the total CPU time required for planning increases significantly. This suggests that to minimize the time spent planning the planning horizon should be as short as possible, leading eventually to a purely reactive planner. Reactive planners, however, may lead the vehicle into dead ends which would have been avoided by a planner looking ahead.

B. Short horizon for new landmarks

In these cases there is a minimum horizon (used when a replan is triggered by the appearance of a new landmark) and a maximum horizon (used when a replan is triggered by the end of the control horizon). The maximum horizon parameter w_1 is constant and the minimum horizon parameter w_2 is varied from 0.3 to 1.0. Results are given in Table 2.

The number of replans triggered by the appearance of a new landmark is roughly constant. This is more a function of vehicle speed, sensor field of view and range and landmark spacing than planning horizon time parameters.

The number of replans triggered by reaching the end of the control horizon decreases steadily from an average of 7.0 replans to an average of 2.7 replans as the minimum planning horizon is increased. This is due to the increased length of the control horizon as the minimum planning horizon increases.

The number of replans also decreases with increasing minimum planning horizon, but the longer time required to generate a plan results in a significant increase in total planning time as w_2 is increased. This suggests that the minimum planning horizon should be as short as possible.

C. *Fully adaptive planning horizon*

In these cases the maximum horizon parameter was fixed at $w_1 = 1.5$ and the minimum horizon parameter was fixed at $w_2 = 0.5$. Full adaptation is enabled by setting $w_3 < 0$. Results are presented in Table 3.

As with the cases discussed above the number of replans triggered by the appearance of new landmarks is approximately constant. As expected the time required to generate a plan for this case is also approximately constant (since a plan of minimum horizon is generated).

As the magnitude of w_3 increases the per-plan CPU time is reduced. This is due to the shorter planning horizon caused by the larger effective velocity u_{eff}. The number of replans also increases, resulting in a minimum total planning time of 16.77 seconds when $w_3 = -2.0$.

Figure 3 shows a series of images from a single run. The mission consists of exploration to a goal followed by return to start through a previously unsurveyed forest. Initially planning horizons are short, reflecting the frequent appearance of new landmarks and the large rate of change of estimated landmark position uncertainty. During most of the return flight planning horizons are long, reflecting the existing knowledge of the environment and the small change in estimated obstacle position uncertainty.

V. Conclusion

This paper has presented an integrated system for navigation using vision as the primary exteroceptive sensor. Vehicle state and obstacle positions are estimated using inertial measurements (acceleration and angular rate) and bearings to obstacles obtained from the vision system. A receding horizon control algorithm is used for path planning in the presence of the uncertainties inherent to the navigation & mapping solution.

An Adaptive Receding Horizon Control (ARHC) strategy has been proposed based on an effective vehicle speed defined as a weighted sum of estimated speed and the rate of change of uncertainty in the obstacle position estimates. This strategy recognizes that plans may quickly become invalid as new information about the environment is obtained, reducing the computational resources spent (and potentially wasted) on planning during times of high information gain.

References

[1] J. Langelaan and S. Rock, "Navigation of small UAVs operating in forests," in Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference, Aug 16-19 2004, pp. 2031-2041.

[2] J. Langelaan and S. Rock, "Towards autonomous UAV flight in forests," in AIAA Guidance, Navigation, and Control Conference, 2005.

[3] S. Thrun, W. Burgard and D. Fox, "Real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in ICRA 2000: IEEE International Conference on Robotics and Automation, Apr 24-Apr 28 2000, pp. 321-328.

[4] S.B. Williams, G. Dissanayake and H. Durrant-Whyte, "Field deployment of the simultaneous localisation and mapping algorithm," in 15th IFAC World Congress on Automatic Control, 2002.

[5] T. Schouwenaars, J. How and E. Feron, "Receding horizon path planning with implicit safety guarantees," in Proceedings of the 2004 American Control Conference (AAC), Jun 30-Jul 2 2004, 2004, pp. 5576-5581.

[6] A. Watkins, A. Kurdila, R. Prazenica and G. Wiens, "RHC for vision-based navigation of a MWR in an urban environment," in AIAA Guidance, Navigation, and Control Conference, 2005.

[7] E.W. Frew, "Receding Time Horizon Control Using Random Search for UAV Navigation with Passive, Non-cooperative Sensing," in AIAA Guidance, Navigation, and Control Conference, 2005.

[8] S. Julier, J. Uhlmann and H.F. Durrant-Whyte, "New method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control,* vol. 45, pp. 477-482, 2000.

[9] R.D. Van Merwe, E.A. Wan and S.I. Julier, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion - Applications to integrated navigation," in Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference, Aug 16-19 2004, 2004, pp. 1735-1764.

Table 1: No adaptation (exploration phase)

mean (std dev.) for 50 runs		$w_1 = w_2 = 0.3$	$w_1 = w_2 = 1.0$	$w_1 = w_2 = 1.5$
plan triggered by new landmark	# replans	10.57 (0.87)	12.58 (2.84)	12.81 (3.03)
	Tplan	1.45 (0.03)	4.96 (0.09)	7.42 (0.11)
	CPU time/plan	0.57 (0.02)	1.81 (0.18)	2.65 (0.33)

plan triggered by end of horizon	# replans	18.04 (0.79)	3.78 (4.32)	1.17 (3.40)
	Tplan	1.45 (0.3)	4.96 (0.13)	2.69 (3.62)
	CPU time/plan	0.55 (0.01)	1.99 (0.41)	1.41 (2.09)
average number of replans		29	16.4	14
total CPU time for planning		15.97 (0.92)	32.30 (21.0)	40.68 (31.4)

Table 2: Adaptation to new landmarks only (i.e. $w_3 = 0$).

mean (std dev.) for 50 runs		$w_1 = 1.5 \quad w_2 = 0.3$	$w_1 = 1.5 \quad w_2 = 0.5$	$w_1 = 1.5 \quad w_2 = 1.0$
plan triggered by new landmark	# replans	10.84 (0.89)	11.22 (2.14)	11.89 (2.51)
	Tplan	1.45 (0.04)	2.45 (0.07)	4.93 (0.06)
	CPU time/plan	0.58 (0.03)	0.90 (0.08)	1.76 (0.16)
plan triggered by end of horizon	# replans	7.00 (0.90)	5.61 (3.21)	2.70 (2.08)
	Tplan	7.44 (0.15)	7.46 (0.15)	7.40 (0.14)
	CPU time/plan	2.40 (0.15)	2.29 (0.38)	2.80 (0.49)
average number of replans		17.84	16.83	14.59
total CPU time for planning		23.27 (4.08)	24.30 (17.31)	29.85 (16.56)

Table 3: Fully adaptive planning. Here $w_1 = 1.5 \quad w_2 = 0.5$ and $w_3 < 0$.

mean (std dev.) for 50 runs		$w_3 = -0.5$	$w_3 = -1$	$w_3 = -2$	$w_3 = -4$
plan triggered by new landmark	# replans	10.73 (0.73)	10.86 (1.01)	10.96 (0.92)	11.0 (1.02)
	Tplan	2.35 (0.46)	2.44 (0.03)	2.45 (0.04)	2.46 (0.04)
	CPU time/plan	0.87 (0.03)	0.90 (0.03)	0.90 (0.03)	0.90 (0.03)
plan triggered by end of horizon	# replans	4.82 (0.75)	4.82 (0.98)	4.78 (0.79)	6.37 (1.70)
	Tplan	6.04 (1.21)	5.46 (0.29)	4.48 (0.36)	3.96 (0.49)
	CPU time/plan	1.84 (0.12)	1.68 (0.18)	1.41 (0.21)	1.32 (0.24)
average number of replans		15.55	15.68	15.74	17.37
total CPU time for planning		18.35 (2.05)	18.01 (3.84)	16.77 (3.12)	18.68 (6.02)

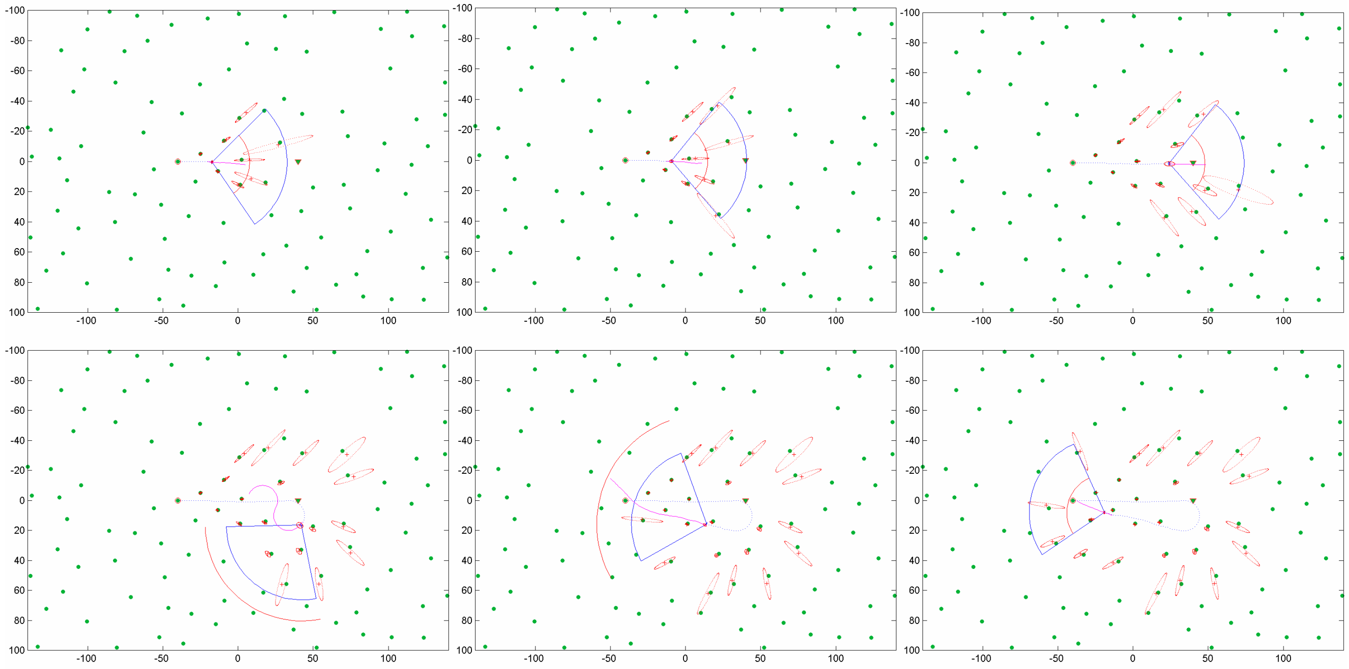

Fig. 3. Snapshots from a run. The red circle is the initial point and the inverted triangle is the goal location, trees are shown as green dots. The blue pie slice is the camera field of view, the red arc the plan horizon, the dotted blue line the path taken, the magenta line indicates the planned trajectory. Note the short planning horizon during the exploration phase of the flight (when there is significant gain in information about obstacle positions and new landmarks frequently come into view) and the longer planning horizon during the return phase of the flight (when obstacles have been accurately localized).