Guidance, Navigation and Control Conference, August 19-13, 2007, Hilton Head, South Carolina

Adaptive Planning Horizon Based on Information Velocity for Vision-Based Navigation

Eric W. Frew*

University of Colorado, Boulder, CO, 80302

Jack W. Langelaan†

Pennsylvania State University, University Park, PA, 16802

Maciej Stachura‡

University of Colorado, Boulder, CO, 80302

This paper presents a receding horizon planning algorithm for vision-based navigation using bearings-only SLAM that adapts the planning horizon to the velocity of the information gained about the environment. Bearings-only SLAM has an inherent *dynamic observability* property such that specific motion is needed in order to determine the relative positions between the camera and obstacles. Thus, the estimates of new obstacles are always highly uncertain and information about the obstacles is dependent on the relative motion between the camera and objects. Receding horizon planners are typically used in conjunction with SLAM algorithms because they continually adjust to new information about the world. We present a receding horizon planner that incorporates information metrics explicitly into the receding optimization cost function. Furthermore, an adaptive horizon is used based on the intuition that when the information about the world is rapidly changing, planning does not need to be too long. 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 receding horizon system for vision-based SLAM; illustrate the value of integrating information-theoretic costs into the objective function and the adaptive approach; and extension of the planning approach to other applications such as exploration, target intercept, and cooperative active sensing.

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

[*] Assistant Professor, Aerospace Engineering Sciences, eric.frew@colorado.edu, AIAA Member.

[†] Assistant Professor, Aerospace Engineering, jlangelaan@psu.edu, AIAA Senior Member.

[‡] Graduate Research Assistant, Aerospace Engineering Sciences, maciej.stachura@colorado.edu, Non-Member.

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[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 significant uncertainty in the 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.[1]

The interdependence between planning and estimation in SLAM and related applications such as target tracking has long been recognized.[5,6] The potential for improved state estimates which result from increasing the "look-ahead" distance been demonstrated,[7–9] but increasing the look-ahead for planning inevitably results in increased computational load.

Receding horizon control (RHC) has been used very successfully for control of unmanned aerial vehicles, especially in the context of obstacle avoidance.[10–12] The limited planning horizon and successive re-planning offers a built-in means of responding to a dynamic environment and serves to reduce the computational load associated with planning. However, the previous research has generally assumed that obstacles are localized perfectly as soon as they enter a local detection region.

Receding horizon control can be augmented to include the output of vision-based estimation.[10,11] This can consist of a learning algorithm that combines feature points into a terrain map of the environment[11] or inclusion of the estimate covariance matrices as extended obstacle boundaries.[10] 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.[10,13]

RHC approaches naturally raise a question of the optimal length of the planning horizon. In previous work we combined vision based estimation and RHC with an adaptive planning horizon.[13] The planning horizon was computed based on the rate of change of uncertainty of surrounding obstacles: in a fast-changing environment plans quickly become obsolete, hence computation spent on long planning horizons is likely to be wasted. In this paper we further develop the adaptive receding horizon control by evaluating the tradeoff between final trajectory cost and total computational effort for several different mission scenarios. Results show that the adaptive approach provides a balance in terms of computational effort between short and long (fixed) planning horizons while yielding better overall performance in terms of the final trajectory cost compared to either fixed case.

The remainder of this paper is organized as follows: Section II describes the vision-based navigation problem; Section III discusses dynamic observability and rates of change of information gain; Section IV summarizes the adaptive RHC controller; Section V provides simulation results; and conclusions are presented in Section VI.

II. Vision-Based Navigation

The concepts developed in this paper are motivated by navigation of a small UA through an unsurveyed, cluttered environment such as a forest (Fig. 1). An on-board camera provides bearings to obstacles and an inertial measurement unit provides accelerations and angular rates 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.

The block diagram in Fig. 2 shows a system that uses the given sensors to perform obstacle avoidance.[1,2] UA guidance is achieved by combining a high-level adaptive receding horizon trajectory generation algorithm with a low-level flight control system that directly controls the aircraft actuators. The trajectory generation and flight control algorithms are fed state variable estimates derived from the vision and inertial sensors.

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

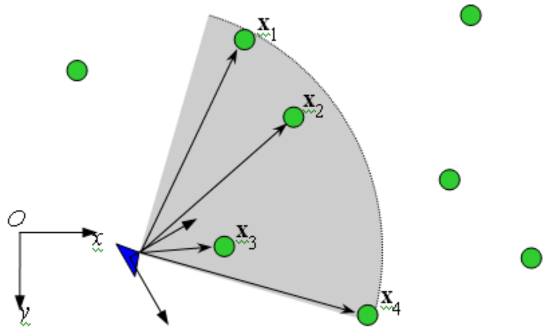


Figure 1. The navigation scenario. The grey region shows the field of view of the vision system, obstacles are show by green dots

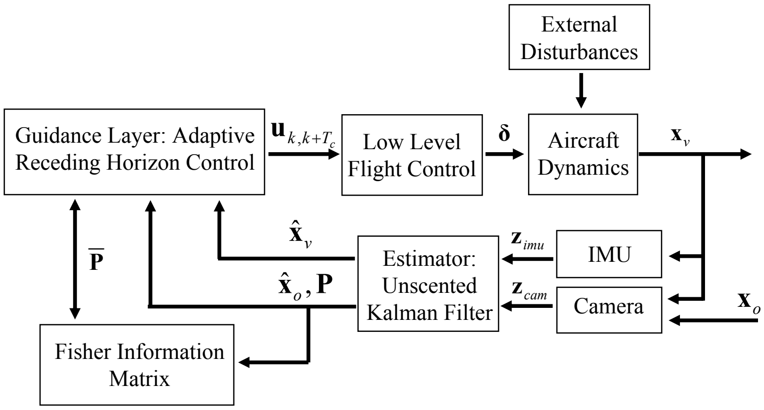


Figure 2. Navigation system schematic.

Filter[14, 15] was designed to compute 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), obstacle positions $\mathbf{x}_o$ (in the inertial frame) and the associated covariance:[1, 2]

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

The covariance is a measure of the uncertainty in the estimated states. The observability of the state estimation, its dependence on vehicle motion, and the implications for adaptive RHC are discussed in Section III.

UA guidance is achieved using an adaptive receding horizon control policy based on the controller described in earlier work.[10, 13] 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 that is fed directly into the guidance control layer of the aircraft flight control system, eliminating the need for a path or trajectory tracking controller. The trajectory generator assumes the low-level flight control system provides the ARHC controller with the standard kinematic model given in discrete form as:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + u_1 \cdot T_s \cdot \operatorname{sinc}(\phi)\cos(\psi_k + \phi) \\ y_k + u_1 \cdot T_s \cdot \operatorname{sinc}(\phi)\sin(\psi_k + \phi) \\ \psi_k + u_2 \cdot T_s \end{bmatrix} \qquad \begin{matrix} v_{min} \le u_1 \le v_{max} \\ \\ |u_2| \le \omega_{max} \end{matrix} \tag{2}$$

where u_1 is the aircraft speed command, u_2 is the turn rate command, v_{min} and v_{max} are bounds on airspeed, ω_{max} is the maximum allowable turn rate, $\phi = 0.5 \cdot u_2 \cdot T_s$, T_s is the sample time, and sinc(x) is the sine cardinal function

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(x)}{x} & x \neq 0 \\ 0 & x = 0 \end{cases} \tag{3}$$

The measurement model is

$$z_{k,i} = \arctan\left(\frac{y_i - y_k}{x_i - x_k}\right) - \psi_k + v_k \tag{4}$$

Receding horizon control is an optimization-based method that seeks to minimize some objective function continually over a finite control horizon. Let $\mathbf{u}_{k,k+T} = [\mathbf{u}_k, \cdots, \mathbf{u}_{k+T}]$ be a control sequence of length $T+1$ starting at discrete time k. Furthermore, let $J_{rhc}(\mathbf{x}_k, \mathbf{u}_{k,k+T})$ be an objective function evaluated with the current state $\mathbf{x}_k$ at time k and control sequence $\mathbf{u}_{k,k+T}$. The main steps of the RHC approach are

1. Calculating the optimal control sequence

$$\mathbf{u}^*_{k,k+T_h} = [u^*_k, \cdots, u^*_{k+T_h}] = \arg\min J_{rhc}(\mathbf{x}_k, \mathbf{u}_{k,k+T}), \tag{5}$$

   starting from state $\mathbf{x}_k$, over a finite *planning horizon* $2 \leq T_h \leq \infty$;
2. Implementing the optimal control input over some *control horizon* $k \leq t \leq k + T_c$ where $1 \leq T_c \leq T_h$;
3. Repeating steps 1 and 2 for $\mathbf{x}_{k+T_c}$ at time $k + T_c$.

The RHC algorithm casts navigation into a deterministic, relative problem in order to simplify analysis[a]. First, the certainty equivalence principle is invoked in order to use the current estimate of the UA state and obstacle positions in the optimization process. This enables minimization of a deterministic cost criterion as opposed to minimization of the expectation of a stochastic process. Second, the uncertainty in the UA state and object positions is transformed into a measure of the uncertainty of position $\mathbf{x}_{rel,i} = \mathbf{x}_v - \mathbf{x}_{o,i}$ of each object relative to the UA. This is accomplished by calculating the new covariance matrix $\mathbf{P}_{rel,i} = \mathbf{P}_{pp} + \mathbf{P}_{oo,i} - \mathbf{P}_{po,i} - \mathbf{P}^T_{po,i}$, where $\mathbf{P}_{pp}$ is the covariance of the position components of the UA state estimate, $\mathbf{P}_{oo,i}$ is the covariance of the i^{th} obstacle position, and $\mathbf{P}_{po,i}$ is the cross correlation between the UA position and the obstacle position. For the goal node, the UA position uncertainty is used to define a terminal set, as opposed to a single terminal state. Finally, safety is encoded by prohibiting UA motion within some uncertainty ellipsoid defined by $\mathbf{P}_{rel,i}$. The shape of this ellipsoid is defined by $\mathbf{P}_{rel,i}$ and its size is determined by a desired level of confidence.

The RHC algorithm used here minimizes the objective

$$J_{rhc}(\hat{\mathbf{x}}_k, \mathbf{u}_{k,k+T}) = J_u + J_{nav} + J_{info} + J_{safe}. \tag{6}$$

Given a vector $\mathbf{y} \in \Re^n$ and matrix $\mathbf{M} \in \Re^{nxn}$ let $\|\mathbf{y}\|^2_{\mathbf{M}}$ be the M-weighted 2-norm of the vector, i.e $\|\mathbf{y}\|^2_{\mathbf{M}} = \mathbf{y}'\mathbf{M}\mathbf{y}$. The control cost J_u and the navigation cost J_{nav} are the standard quadratic cost functions

$$J_u(\mathbf{u}_{k,k+T}) = \sum_{i=k}^{k+T_h} \|\mathbf{u_k}\|^2_{\mathbf{W_u}} \tag{7}$$

$$J_{nav}(\hat{\mathbf{x}}_k, \mathbf{u}_{k,k+T}) = \sum_{i=k}^{k+T_h} \|\hat{\mathbf{p}}_\mathbf{i} - \mathbf{p}_{goal}\|^2_{\mathbf{W_{nav}}} + \|\hat{\mathbf{p}}_{\mathbf{k+T_h}} - \mathbf{p}_{goal}\|^2_{\mathbf{W}_{goal}} \tag{8}$$

where $\mathbf{W}_u = W_u \cdot \mathbf{I}$, $\mathbf{W}_{nav} = W_{nav} \cdot \mathbf{I}$, and $\mathbf{W}_{goal} = W_{goal} \cdot \mathbf{I}$ are positive-definite weighting matrices, $\hat{\mathbf{p}}_k$ is the position component of the aircraft state estimate, $\mathbf{p}_{goal}$ is the current goal location, and $\hat{\mathbf{x}}_{k+i}, i \in [1, \cdots, T_h]$ is calculated from Equation 2. The information cost J_{info} encourages the UA to improve its knowledge of the state of the N obstacles observed in the environment

$$J_{info}(\hat{\mathbf{x}}_k, \mathbf{u}_{k,k+T}) = \frac{W_{info}}{N} \cdot \sum_{i=k}^{k+T_h} \sum_{j=1}^{N} \frac{1}{\eta_{A,j,i}} \tag{9}$$

[a] The "adaptive" portion of ARHC comes from a computation of the planning horizon T_h based on vehicle speed and the rate of change of uncertainty. This is described in Section IV.

where $\overline{\eta}_{A,j,i}$ is a measure of the expected information content in future measurements. Details on the function $\overline{\eta}_{A,j,i}$ are provided in Section III.

Finally, the safety cost J_{safe} enables the RHC controller to perform obstacle avoidance. Rather than include obstacle avoidance as a constraint, we use a penalty function in the optimization objective function. This leaves the control limits as the only constraints to the optimization problem. For the work presented here we consider only obstacles with circular or elliptical shape. If the j^{th} obstacle is located at position $\mathbf{p}_{o,j}$ then the safety cost is given by

$$J_{safe}(\hat{\mathbf{x}}_k, \mathbf{u}_{k,k+T}) = \mathbf{W}_{safe} \cdot \sum_{i=k}^{k+T_h} \sum_{j=1}^{N} b(\|\hat{\mathbf{p}}_\mathbf{i} - \hat{\mathbf{p}}_{o,j}\|^2_\mathbf{M}, r_{barrier}). \tag{10}$$

The function $b(x, r)$ is the barrier function

$$b(x,r) = \begin{cases} m \cdot (r - x) & x \leq r \\ 0 & x > r \end{cases} \tag{11}$$

where $m > 0$ is the slope of the barrier function. This value is typically set to $m = 1$ and the weighting parameter $\mathbf{W}_{safe}$ is used to set the steepness of the barrier. Setting $\mathbf{M} = \mathbf{1}$ (the identity matrix) in Equation 10 yields circular obstacles with radius $r_{barrier}$. Alternatively, setting $\mathbf{M} = \overline{\mathbf{P}}^{-1}_{rel,j,k}$ to the inverse of the approximation of the obstacle position error covariance matrix yields an elliptical confidence region defined by the parameter $r_{barrier}$. In this case the weighted norm in Equation 10 corresponds to the Mahalanobis distance between the target and the aircraft position estimates. 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 $\mathbf{I}_F$ (see Section III) to describe the information gained about an obstacle: by assuming the target estimate is exact over the finite planning horizon and assuming exact knowledge of aircraft motion[10] we can calculate

$$\overline{\mathbf{P}}_{rel,j,k} = \mathbf{I}^{-1}_{F,j,k}. \tag{12}$$

Figure 3 shows an example simulation run using the RHC approach without the information cost term. The solid line represents the actual path of the aircraft, the dotted line represents the estimated path, and the dashed line represents the predicted future path over the planning horizon. The black cone represents the field of view of the UA camera and the red line denotes the planning horizon.

A. RHC and ADP

Receding horizon control is an example of the broader class of model predictive control (MPC) algorithms. Bertsekas draws strong connections between MPC and approximate dynamic programming (ADP), essentially showing that both concepts are based on the dynamic programming idea of policy iteration and rollout.[16] Although the stability of MPC has been addressed extensively in the literature (c.f 17), we use the language of ADP here in order to motivate the use of adaptive planning and control horizons.

Consider the discrete time dynamical system whose state vector evolves according to the equation

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{13}$$

where k is the discrete sample time, $\mathbf{x}_k \in X_k$ is the state vector constrained to some finite set, $\mathbf{u}_k \in U_k$ is the vector of control inputs selected from a finite set, and $\mathbf{w}_k$ is a random disturbance. Starting at the state $\mathbf{x}_k$, the general dynamic programming problem is to find

$$J^*_k(\mathbf{x}_k) = \min_{\mathbf{u}_i \in U_i} J_k(\mathbf{x}_k) \tag{14}$$

where

$$J_k(\mathbf{x}_k) = E\left\{ \sum_{i=k}^{N-1} g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i) + g_N(\mathbf{x}_N) \right\} \tag{15}$$

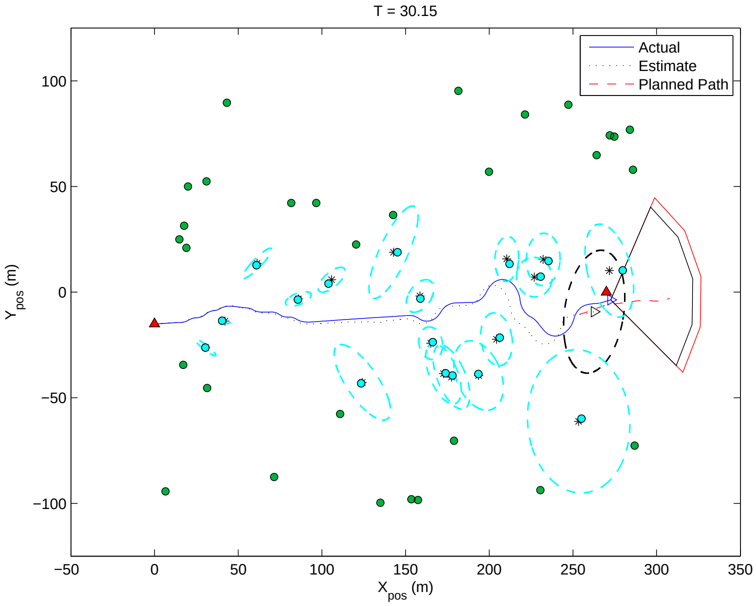


Figure 3. Example simulation run using ARHC. The solid line is path flown, solid circles are obstacles, and ellipses show estimated obstacle positions.

is the cost-to-go function for the problem with $g_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i)$ representing the cost incurred at each stage (over each discrete time interval). One of the key concepts of dynamic programming is the recognition that the optimal cost-to-go function Equation 14 satisfies the recursion

$$J_k^*(\mathbf{x}_k) = \min_{\mathbf{u}_i \in U_i} E\left\{g_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) + J_{k+1}^*(f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k))\right\} \tag{16}$$

with the initial condition $J_N^*(\mathbf{x}_N) = g_N(\mathbf{x}_N)$.

In practice, solving the stochastic optimization of Equation 16 with imperfect state information and random disturbances is computationally prohibitive and suboptimal approximations are applied. In order to replace the stochastic problem with a deterministic one the *certainty equivalence principle* is often invoked to replace the imperfect state with its estimate and to use a nominal value for the disturbance (which we take to be zero here). Further, an *implicit cost-to-go approximation* is used to substitute $J_{k+1}(f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k))$ in Equation 16 with the function $\tilde{J}_{k+1}(f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k))$ computed online from some heuristic/suboptimal base policy. Thus the new problem becomes solving

$$\hat{J}_k(\hat{\mathbf{x}}_k) = \min_{\mathbf{u}_i \in U_i}\left[g_k(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \tilde{J}_{k+1}(f_k(\hat{\mathbf{x}}_k, \mathbf{u}_k))\right] \tag{17}$$

at each stage.

Comparing Equation 17 to the Equation 6 - Equation 10 we see that the receding horizon control objective function has the same form as the approximate cost-to-go function with

$$\tilde{J}_{k+1}(\hat{\mathbf{x}}_{k+1}) = \left\{\sum_{i=k+1}^{k+T_h} g_{rhc,i}(\hat{\mathbf{x}}_i, \mathbf{u}_i) + \tilde{J}_{ctg}(\hat{\mathbf{x}}_{k+T_h})\right\}. \tag{18}$$

This corresponds to *multi-step look ahead* where $\tilde{J}_{ctg}(\hat{\mathbf{x}}_{k+T_h})$ is the approximately cost-to-go function which corresponds to the (infeasible) base policy of "heading toward the goal in a straight line".

Convergence of ADP (and stability of RHC) is dependent on an assumption of cost improvement, specifically that

$$\tilde{J}_k(\hat{\mathbf{x}}_k) \geq \min_{\mathbf{u}_i \in U_i}\left[g_k(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \tilde{J}_{k+1}(f_k(\hat{\mathbf{x}}_k, \mathbf{u}_k))\right]. \tag{19}$$

This assumption puts constraints on the RHC problem. For example, the implementation described here cannot properly account for dead-end paths and must assume obstacles can be avoided individually. Furthermore, when an approximate cost-to-go function satisfies Equation 19 the actual cost $\overline{J}_k(\mathbf{x_k})$ obtained as a result of solving Equation 17 at each stage k is bound by the approximation, ie.

$$\overline{J}_k(\mathbf{x_k}) \leq \hat{J}_k(\hat{\mathbf{x}}_k). \tag{20}$$

Thus, the closer $\hat{J}_k(\hat{\mathbf{x}}_k)$ is to the optimal cost $J_k^*(\hat{\mathbf{x}}_k)$, the better the final trajectory.

Although the result of Equation 20 may seem intuitive, it has important ramifications since $\hat{J}_k(\hat{\mathbf{x}}_k)$ is computed online and we never know how closely it approximates $J_k^*(\hat{\mathbf{x}}_k)$. Instead, we will infer properties of $\hat{J}_k(\hat{\mathbf{x}}_k)$ based on our perception of the environment and the uncertainty in that perception. In particular, when the environment is changing rapidly we have no reason to assume that the projected stage costs at some future time will reflect the true cost of the stage. This suggests that the length of the planning horizon should depend on how quickly the environment changes.

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 $2N$ 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 (see Section IV). 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. Dynamic Observability, Active Sensing, and Information Acceleration

A fundamental characteristic of vision-based localization and mapping is the dependence of the observability of the system states on motion of the camera (i.e. the trajectory flown by the UA). During motion directly towards or away from an object (along the bearing) there is no new information to improve the range estimate. Thus, 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.

In general, this dependence of estimation performance on system inputs is referred to as *dynamic observability* and forms the basis of *active sensing*. The main goal of active sensing systems is to choose the inputs (e.g. UA motion) that optimize estimation performance. Inputs which optimize estimator performance are found by maximizing some scalar measure η of the Fisher Information Matrix (FIM) $\mathbf{I}_{F,k}$ which is equivalent to the inverse of the Posterior Cramer-Rao Lower Bound,[18] i.e.

$$\mathbf{I}_{F,k}^{-1} = \mathbf{P}_{CRLB,k} \leq \mathbf{P}_k. \tag{21}$$

The FIM represents the sensitivity of the estimation process at a given point to the unknown parameter. Better sensitivity means a measurement is more likely to add new information or reduce the uncertainty of the unknown parameters.

Consider a discrete time nonlinear estimation problem. Let the state vector $\mathbf{x}_k$ evolve according to the state equation Equation 13 where the noise $\mathbf{w}_k$ is zero-mean, spatially uncorrelated, white Gaussian process noise with covariance matrix $E[\mathbf{w}_k \mathbf{w}_k'] = \mathbf{Q_k}$. Furthermore, let $\mathbf{z}_{k,i}$ denote the i^{th} measurement of the system with

$$\mathbf{z}_{k,i} = h_i(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_{k,i} \tag{22}$$

where $\mathbf{v}_k$ is zero-mean, spatially uncorrelated, white Gaussian measurement noise with covariance $E[\mathbf{v}_{k,i}\mathbf{v}_{k,i}^T] = \mathbf{R}_{k,i}$. The FIM for this system can be recursively calculated as[19]

$$\mathbf{I}_{F,k} = \left[\mathbf{\Phi}_{k-1}\mathbf{I}_{F,k-1}^{-1}\mathbf{\Phi}_{k-1}^T + \mathbf{\Gamma}_{k-1}\mathbf{Q}_{k-1}\mathbf{\Gamma}_{k-1}^T\right]^{-1} + \mathbf{I}_{z,k} \tag{23}$$

$$\mathbf{I}_{z,k} = E\left[\sum_{i=1}^{n} \mathbf{H}_{k,i}^T \mathbf{R}_{k,i}^{-1} \mathbf{H}_{k,i}\right] \tag{24}$$

$$\mathbf{\Phi}_k = \frac{\delta \mathbf{f}}{\delta \mathbf{x}}(\mathbf{x}_k, \mathbf{u}_k, 0) \tag{25}$$

$$\mathbf{\Gamma}_k = \frac{\delta \mathbf{f}}{\delta \mathbf{w}}(\mathbf{x}_k, \mathbf{u}_k, 0) \tag{26}$$

$$\mathbf{H}_{k,i} = \frac{\delta \mathbf{h}}{\delta \mathbf{x}}(\mathbf{x}_k, \mathbf{u}_{k,i}, 0) \tag{27}$$

where $\mathbf{\Phi}_k$, $\mathbf{\Gamma}_k$, and $\mathbf{H}_{k,i}$ are the linearized state transition matrix, noise input matrix, and measurement matrix respectively; n refers to the number of independent sensors measuring the system; and the expectation in Equation 24 is over the unknown system states. The recursion is initialized by the inverse of the covariance matrix

$$\mathbf{I}_{F,0} = \mathbf{P}_0^{-1}. \tag{28}$$

The active sensing literature contains several popular choices for the scalar function including

1. The D-optimality (determinant) criterion

$$\eta_D(\mathbf{I}_{F,k}) = \det(\mathbf{I}_{F,k}). \tag{29}$$

2. The A-optimality (trace) criterion

$$\eta_A(\mathbf{I}_{F,k}) = \mathrm{trace}(\mathbf{I}_{F,k}). \tag{30}$$

3. The E-optimality (eigenvalue) criterion

$$\eta_E(\mathbf{I}_{F,k}) = \lambda_{min}(\mathbf{I}_{F,k}). \tag{31}$$

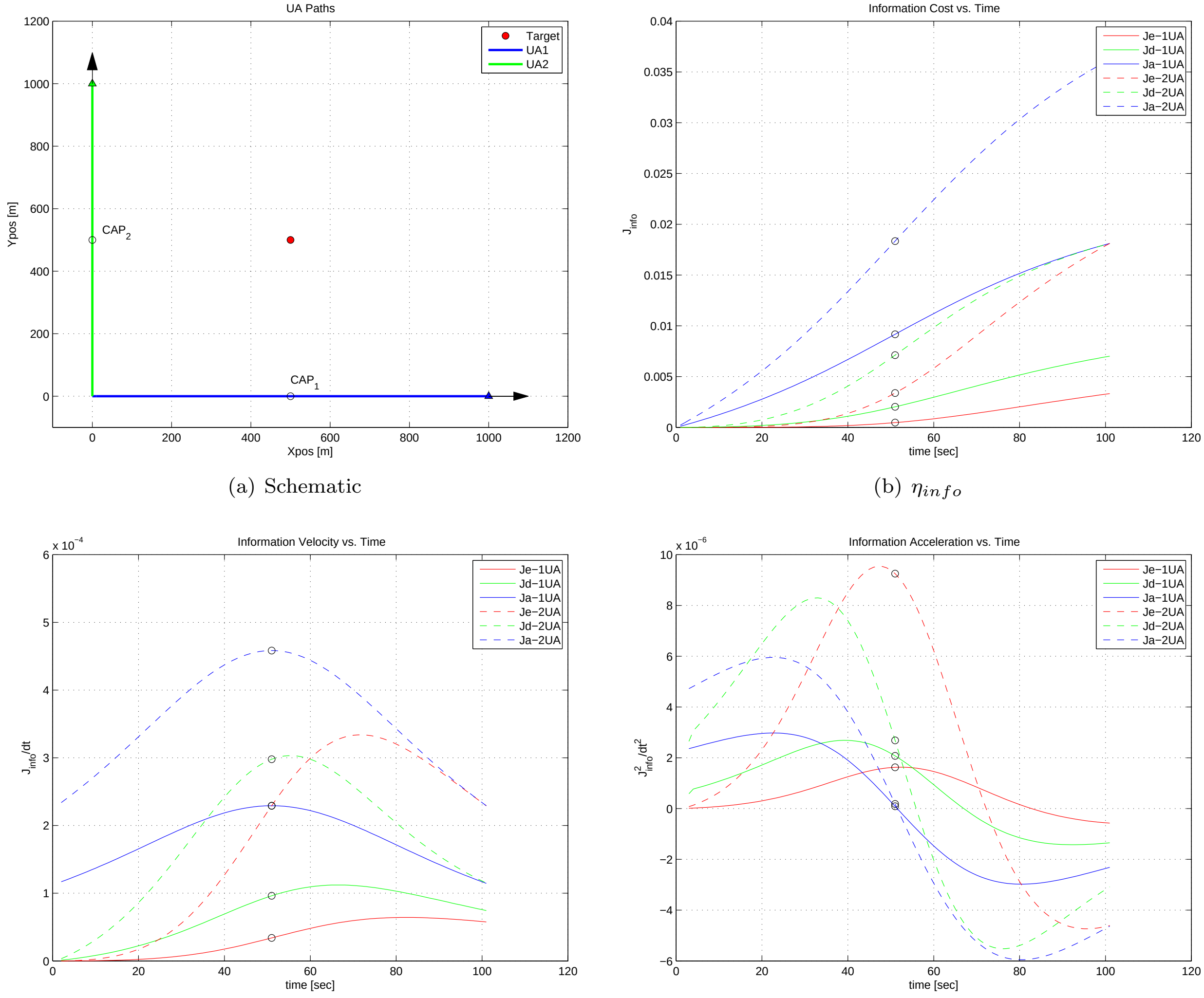
where $\lambda_{min}(\mathbf{M})$ is the minimum eigenvalue of the matrix $(\mathbf{M})$. For simplicity we will write $\eta_{A,k}$ to represent $\eta_A(\mathbf{I}_{F,k})$.

Dynamic observability can now be restated as the dependence of the scalar criterion η (which we will refer to as measures of the information gained by the system) on the system state and the system input (and additional nuisance parameters like target motion that cannot be controlled). As a result of this relationship, the time rate of change of the optimality criteria varies with time and the estimation system has state- and input-dependent information acceleration, i.e. $\dot{\eta}_k = f(\eta_k, x_k, u_k)$. In contrast, the FIM for a linear system is a function only of the initial estimate error covariance and the system matrices and thus $\dot{\eta}_k = f(x_0, P_0)$.

Figure 4 shows the information and its velocity and acceleration for a simple BOT example.[20] Two UA start at the origin moving with velocities [10, 0] m/s and [0, 10] m/s. They take bearing measurements of a stationary target located at [500, 500] m. The plots show the information (and rates) using the three different optimality criterion (Equation 29 - Equation 31) and for estimation based on measurements from the first UA only and from both. The open circles represent the closest approach point (CAP) of each UA to the target and the corresponding measurements of information (and rates) at that time.

The simple BOT example shows several important results. (Note, comparison of the magnitudes of the information and information rates across different criteria is not meaningful since they have varied interpretations.) As expected, information increases throughout the simulation (i.e. the information velocity is always positive) since the targets are always in view and there is no uncertainty in their motion (they are static). Likewise, for a given optimality criterion information is gained at a faster rate for cooperative estimation compared to the single-vehicle case. In every case the information velocity and acceleration peak when the vehicles are at or just past the closest approach point, which occurs at t = 51.0s for both examples. In all cases the information velocity decreases after some point and the information metric flattens.

The results have several implications for navigation and choosing the planning horizon. From the ARHC planner's perspective, an uncertain obstacle is treated the same as a large certain one. Thus, the magnitude of the information matrix (and the target uncertainty) is not as important as the information rate. When the information velocity is large the planner can reasonably expect the world to change significantly while it



(a) Schematic

(b) η_{info}

(c) $\dot{\eta}_{info}$

(d) $\ddot{\eta}_{info}$

Figure 4. Information, information velocity, and information acceleration versus time. Note that sensor field of view is 360 degrees, hence the target is always in view.

plans, so computation far into the future is useless. Furthermore, the information velocity itself is dynamic and changes as the system evolves. Therefore a specific fixed planning horizon (e.g. short for situations with large information velocity or long for static environments) would not always be appropriate. Finally, the results show that the A-optimality criterion is intuitively the most appealing, since information velocity peaks at CAP. This is ideal for the planning horizon computation since we can start planning longer trajectories as soon as we pass an obstacle.

It is important to note that the Fisher Information Matrix and Cramer Rao Lower Bound are calculated based on the *true* state variable, which in practice is never known. Thus, the planner creates an estimate $\bar{\mathbf{I}}_{F,k}$ of the FIM based on the state estimate $\mathbf{x}_k$. Furthermore, calculation of the FIM for the full SLAM problem is computationally prohibitive so we calculate the FIM for a simplified bearings-only problem that decouples the FIM of each individual obstacle. In particular, we assume the vehicle knows its state exactly and only consider the relative position between the aircraft and each target. We assume a static target ($\mathbf{\Phi}_{k,i} = \mathbf{I}$ and $\mathbf{Q}_{k,i} = \mathbf{0}$), measurement of the target bearing with noise variance $E[v^2] = \sigma_\theta^2$, and initial information given by the inverse of the relative position covariance $\mathbf{P}_{rel,i}$. If the planner is evoked at time k with state estimate $\hat{x}_k$, then the (estimated) FIM for the i^{th} obstacle at time $k + t$ with $t \in [1, T_h]$ is

$$\bar{\mathbf{I}}_{F,k+t,i} = \mathbf{P}_{rel,k,i}^{-1} + \sum_{j=k+1}^{k+t} \begin{bmatrix} \frac{\sin^2 \hat{\theta}_{j,i}}{\hat{r}_{j,i}^2 \sigma_\theta^2} & -\frac{\cos \hat{\theta}_{j,i} \sin \hat{\theta}_{j,i}}{\hat{r}_{j,i}^2 \sigma_\theta^2} \\ -\frac{\cos \hat{\theta}_{j,i} \sin \hat{\theta}_{j,i}}{\hat{r}_{j,i}^2 \sigma_\theta^2} & \frac{\cos^2 \hat{\theta}_{j,i}}{\hat{r}_{j,i}^2 \sigma_\theta^2} \end{bmatrix} \tag{32}$$

where $\hat{\theta}_{j,i}$ and $\hat{r}_{j,i}$ are predicted estimates of the relative bearing and range from aircraft to target, respectively. Finally, the A-optimality criteria for the entire forest is calculated by summing the trace of each individual FIM (this is equivalent to the trace of the larger matrix with individual FIMs as diagonal elements)

$$\overline{\eta}_{A,k+t} = \sum_{i=1}^{N} \text{trace}(\bar{\mathbf{I}}_{F,k+t,i}) = \sum_{j=1}^{N} \overline{\eta}_{A,k+t,i} \tag{33}$$

IV. Adaptive Planning Horizon

The main motivation for adaptive receding horizon control is the fact that dynamically observable systems exhibit information velocity and acceleration. From the perspective of a planning system, obstacles with large uncertainty can easily be treated as larger objects and with the reduction in obstacle position uncertainty more efficient trajectories can be planned. Information velocity and acceleration imply that the state of the world will change significantly over the control horizon of the receding horizon controller. Planned trajectories may quickly become invalid (or unnecessarily conservative, and thus expensive) as knowledge of the surrounding obstacles improves. Thus, adapting the planning horizon to the information velocity seeks to minimize the amount of computational resources devoted to portions of a plan that will never be used.

For the algorithm presented here, 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. The control horizon is defined to be a fixed fraction of the planning horizon, i.e. $T_c = f \cdot T_h$ with $0 < f \leq 1$.

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:

$$T_{plan,k} = \max\left(w_1 \frac{r_{sense}}{u_{eff,k}}, w_2 \frac{r_{sense}}{\hat{u}_k} \right). \tag{34}$$

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 $w_1 \geq 1$ and $0 \leq w_2 \leq 1$. Choosing $w_1 \geq 1$ gives the UA extra time to plan motion around obstacles in its vicinity and w_2 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 intuition that rate of change of uncertainty has the same dimension as vehicle speed:

$$u_{eff,k} = \hat{u}_k + w_3 \frac{d}{dt} \sqrt{\text{trace}(\mathbf{P}_{rel})} \tag{35}$$

Here $\mathbf{P}_{rel}$ is the relative position error covariance for the obstacles (i.e. the uncertainty in the obstacle position relative to the aircraft) 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

$$\frac{d}{dt} \sqrt{\text{trace}(\mathbf{P}_{rel})} \leq 0 \tag{36}$$

unless a previously unseen landmark appears in view. In other words, information about obstacle positions is not lost. Thus choosing $w_3 \leq 0$ ensures that $u_{eff,k} \geq \hat{u}_k$. 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.

V. Results

This section presents simulation results investigating effects of the adaptive time horizon on the performance of the resulting trajectories. In previous work we investigated the computational aspects of the adaptive receding horizon controller in the context of a single navigation scenario,[13] while here we look at the trade off between computational time and final cost for several different scenarios that include navigation, exploration, target intercept, and cooperative tracking.

For all cases the control horizon is fixed at 20% of the planning horizon. Vehicle speed was 10m/s and maximum turn rate was 1.0 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 is 50m, field of view is 100 degrees, and the bearing measurement noise variance is $\sigma_v = 0.01$rad. IMU measurements are collected at 20 Hz while vision measurements are taken at 2 Hz. The simulation itself is integrated at 20 Hz although the sample time used by the ARHC planner is the vision rate of 2 Hz. Optimization of the ARHC objective function is performed using sequential quadratic programming (the *fmincon* function in Matlab) from an initial condition determined by random search.[10] Figure 3 shows an example trajectory through the forest.

Since the time horizon over which the ARHC objective function is evaluated changes due to the adaptation, we scale the control, navigation, and information costs by the number of discrete stages within the planning horizon. In other words,

$$W_u = \frac{T_s}{T_h} \cdot W_{u,0} \qquad W_{nav} = \frac{T_s}{T_h} \cdot W_{nav,0} \qquad W_{info} = \frac{T_s}{T_h} \cdot W_{info,0} \tag{37}$$

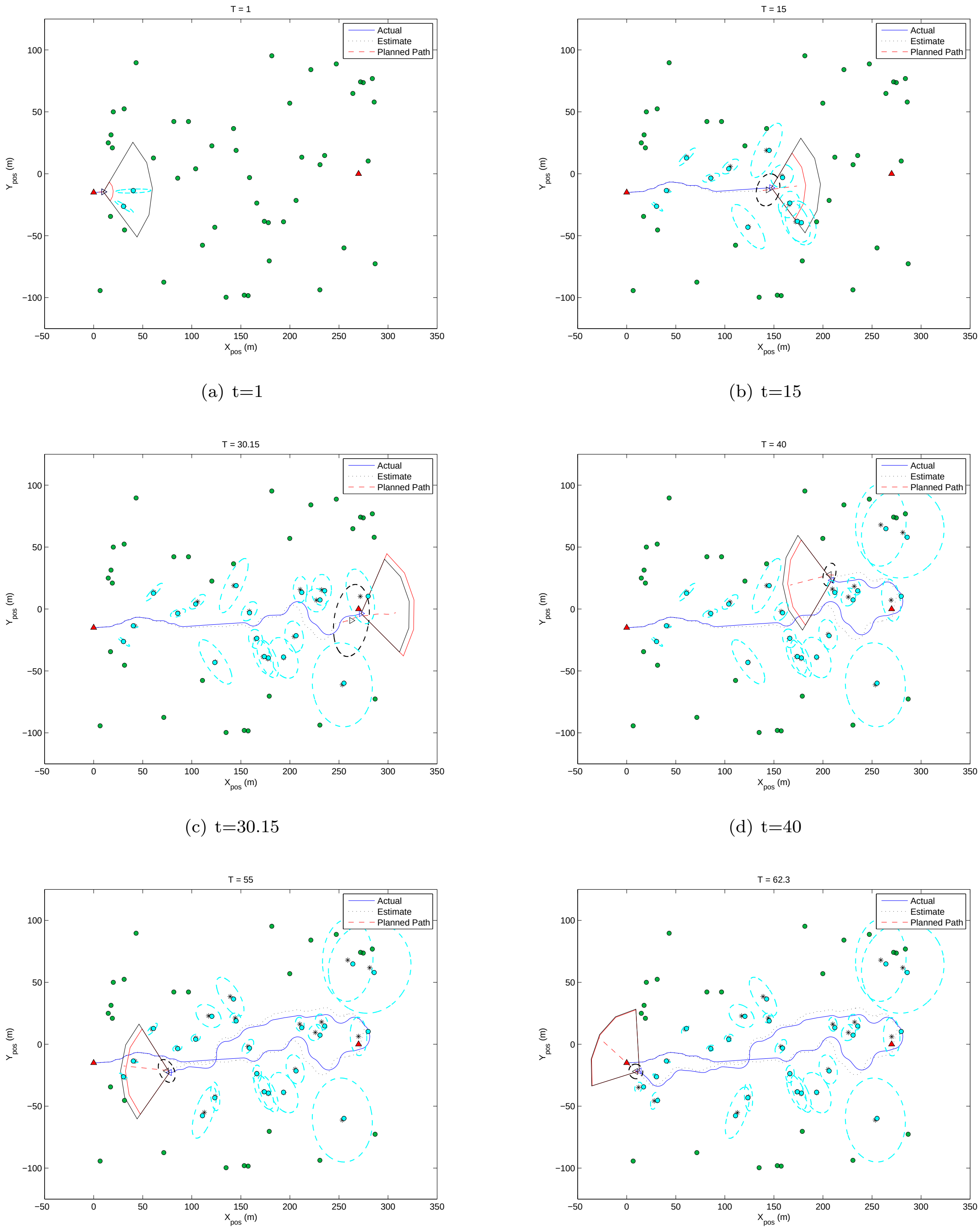
where T_h is the planning horizon, T_s is the sample time (0.5 s in our case), $W_{u,0} = \frac{1}{2 \cdot \omega_{max}} = 0.5$, $W_{nav,0} = \frac{1}{\|\mathbf{p}_{goal}\|}$, and $W_{info,0} = \frac{0.1}{2 \cdot r_{cam}^2} = 2 \cdot 10^{-5}$. Each weight is scaled by the maximum possible value at a single instant in time in order to make their values comparable. The goal cost (when used) is taken to be $W_{goal} = W_{nav,0}$ and the safety cost is always $W_{safe} = 10^{10}$.

A. Navigation

The *navigation* mission is flight through an unknown, cluttered forest to a goal position followed by return to the start position. This scenario provides two distinct phases which are analyzed separately. On the way toward the goal the positions of the trees are highly uncertain since they are discovered as the aircraft flies. On the return leg most of the trees have already been detected and the uncertainty in the world is much lower. A forest of 100 trees randomly distributed over a 300m by 200m area was used for all simulations (see Fig. 3). The vehicle start position was at (0,-15)m, the goal location was at (270,0)m with the goal region being a circle of radius 4m around the goal. Results are averaged over 100 runs.

1. *Control and Navigation Cost Only, Fixed Range Safety Cost*

For the first navigation scenario we consider the control and navigation costs only (the safety cost is always used so we do not distinguish it) with obstacles set as circles of fixed radius ($\mathbf{M} = \mathbf{I}$ and $r_{barrier} = r_{safe} = 4$m in Equation 10) around the estimated target positions. Table 1 shows the cost weights used in this case. Six different sets of adaptive parameters are considered (see Table 2). The first two correspond to short and long fixed horizons, the third corresponds to the case when the long horizon is used except when a new tree is detected in which case the short horizon is used, and the remaining three adapt the horizon based on the effective velocity. Figure 5 shows a sequence of snapshots from one simulation run. On the way toward the goal region (red triangle icon) the planning horizon is short since new, uncertain obstacles are continually detected. On the return leg the uncertainty is constant (and low) so the planning horizon lengthens. The value of the adaptive planning horizon can be seen by looking at the trade-off between computational effort and final trajecory cost for the different scenarios given by Table 2. Figure 6 shows plots of (average) final cost versus (average) total computational time for each scenario during the outbound



(a) t=1

(b) t=15

(c) t=30.15

(d) t=40

(e) t=55

(f) t=62.3

Figure 5. Example simulation results for Adapt1.

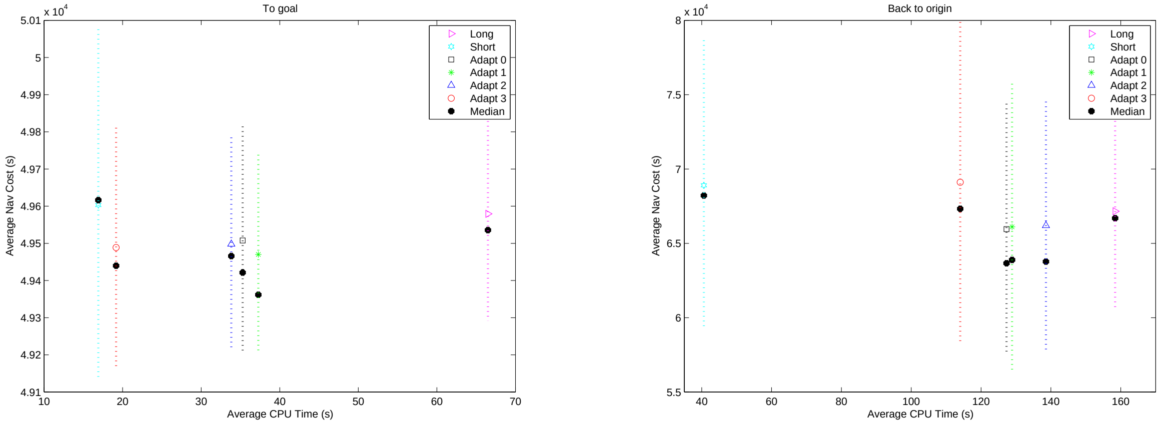
Table 1. Weights used in ARHC Objective Function for Navigation with Control and Navigation costs only

$W_{u,0}$	$W_{nav,0}$	W_{goal}	W_{safe}	W_{info}
0.5	0.0037	0.0037	10^{10}	0

Table 2. Adaptive parameters

Run	w_1	w_2	w_3	Avg cost (there)	Avg comp.(there)	Avg cost (back)	Avg comp.(back)
Short	0.3	0.3	0	49603.14349	16.9078	68901.87951	40.6015
Long	1.0	1.0	0	49579.29843	66.5034	67156.60450	158.3957
Adapt0	0.3	1.0	0	49507.74310	35.2758	65953.23309	127.2858
Adapt1	0.3	1.0	-0.1	49469.96601	37.2773	66109.01196	128.8650
Adapt2	0.3	1.0	-1.0	49497.94139	33.8358	66198.47719	138.5768
Adapt3	0.3	1.0	-10.0	49488.86392	19.1661	69108.62187	114.0332

and return legs, respectively (error bars show 3 standard deviations around the mean, colored marker shows the mean, dark marker shows the median). As expected, the short planning horizon uses the least amount of total processing, the long horizon uses the most, and the adaptive approach falls between the two. Likewise, the long horizon achieves a better (lower) final cost than the short horizon. Surprisingly, the adaptive horizon achieves a better final cost than either fixed horizon. This implies that by adapting the planning horizon to the information velocity we are able to approximate the cost-to-go more acccurately over the course of the entire run. The adaptive horizon also has an order of magnitude better effect on the return leg compared to the outbound case (the axis on the subplot of Fig. 6 are different). From the sequence in Fig. 5 we see that the aircraft aminly encounters obstacles it has seen before. These obstacles have an initially high information velocity (shorter horizon) after which they are well localized and a longer horizon is used to complete the run. Thus, we see an example where neither the short nor long horizons would be applicable and the adaptive control approach prevails. The Adapt3 case on the return leg gives the only anamolous result for this scenario. It is unclear why the large adaptive gain led to an increase in the final trajecory cost.



(a) outbound

(b) return

Figure 6. Final trajectory cost versus total computational effort for navigation with fixed-radius obstacles.

2. Control and Navigation Cost Only, Mahalanobis Distance Safety Cost

The second scenario repeats the first navigation scenario (same weight and parameter values in Table 1 and Table 2) with the exception of how the safety cost is calculated. In this scenario the position estimate error covaraince matrix is used to determine the M-weighted norm in Equation 10. This is done by taking the covariance matrix that gives a specified confidence level (i.e. the "n-σ" ellipse) and expanding it by the safety distance r_{safe}. Let the i^{th} relative position estimate error covariance have singular value decomposition

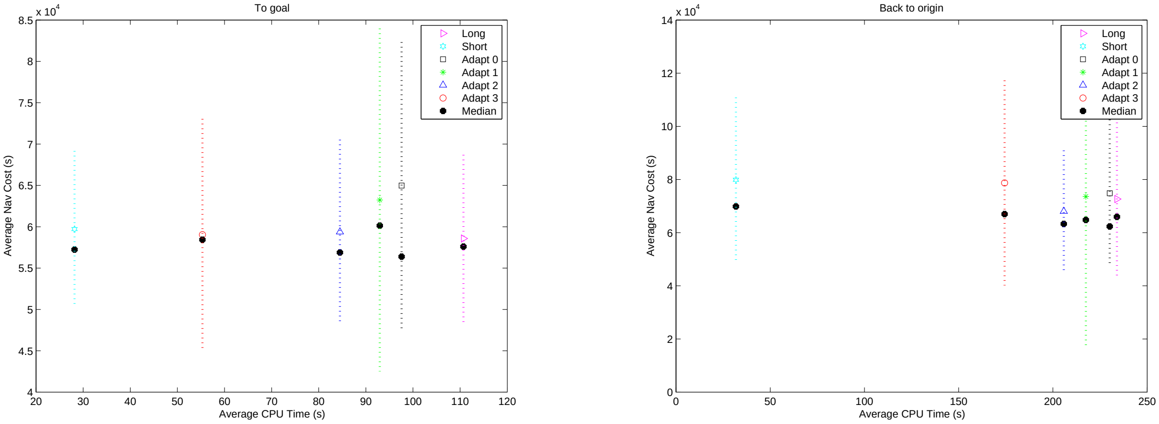
$$\mathbf{P}_{rel,i} = \mathbf{U}\left[\begin{array}{cc} \sigma_1 & 0 \\ 0 & \sigma_2 \end{array}\right]\mathbf{V}. \tag{38}$$

The singular values of $P_{rel,i}$ represent the lengths of the major and minor axis of a bounding ellipse while the orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$ define its orientation. The specified safety distance is added to each singular value and the transformation matrix $\mathbf{M}_i$ is created by re-multiplying by the orthonormal matrices to yield

$$\mathbf{M}_i = \mathbf{U}\left[\begin{array}{cc} n_\sigma^2\sigma_1 + r_{safe}^2 & 0 \\ 0 & n_\sigma^2\sigma_2 + r_{safe}^2 \end{array}\right]\mathbf{V} \tag{39}$$

where n_σ is a function of the desired confidence level (set to 3 here). The net effect is to stretch the bounding ellipsoid by the distance r_{safe} in the direction of both axes of the ellipse. The safety cost (Equation 10) is then called using the matrix $\mathbf{M}_i$ and $r_{barrier} = 1$.

Figure 7 shows the results of the simulation runs using variable-sized obstacles. Unlike the previous example using fixed obstacle size, use of the expanded 3-σ uncertainty ellipsoid $\mathbf{M}$ can cause dead-end regions which the aircraft cannot pass through. Thus the planner backtracks before moving forward, causing performance to degrade. The presence of outliers due to this backtracking behavior skews the resulting distributions of computational time and final trajectory cost. The skewness in the distributions can be inferred by the differences between the mean and median values on the plots. The computational times vary as expected across the different planning horizons, while the trajectory cost distributions do not always show improved performance for the adaptive cases.



(a) outbound

(b) return

Figure 7. Final trajectory cost versus total computational effort for navigation with variable-sized obstacles.

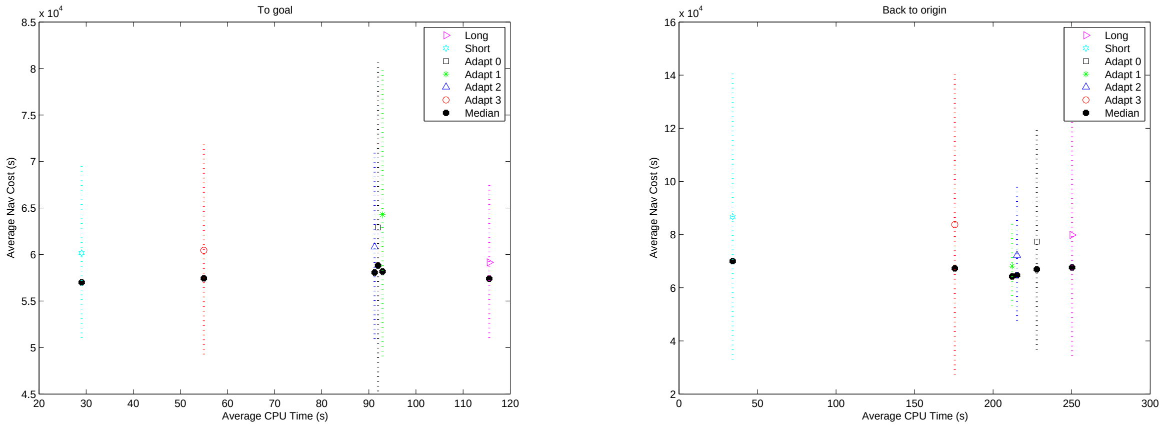
3. Control, Navigation, and Information Cost

The final navigation scenario includes the information cost in order to encourage the aircraft to explore the environment while heading toward the goal. The expectation is that by gaining more information early about the obstacle positions, a better overall trajectory can be planned. The same adaptive weights as Table 2 are used as well as the same safety cost as the previous scenario. The objective function weights used here are given in Table 3.

Table 3. Weights used in ARHC Objective Function for Navigation with Information Cost

$W_{u,0}$	$W_{nav,0}$	W_{goal}	W_{safe}	W_{info}
0.5	0.0037	0.0037	10^{10}	$2 \cdot 10^{-5}$

Figure 8 shows the results of the simulation runs for this case. Like the previous two cases, the computational effort using the adaptive control horizon falls between the short and long horizons. Also like the previous case, outliers due to dead-ends in the planning space cause a large dispersion in the final trajectory costs and the adaptive horizon does not always perform better than the fixed cases. The intuition that the information cost encourages exploration that ultimately benefits navigation performance is not supported here. Table 4 shows that the navigation performance (navigation plus control costs only) degrades when the information cost is included. It is possible that the information cost could help for different sets of relative weights or for forests of greater density.



(a) outbound

(b) return

Figure 8. Final trajectory cost versus total computational effort for navigation including the information cost with variable-sized obstacles.

Table 4. Navigation plus control costs when also using information cost

	with info (there)	with info (back)	w/o info (there)	w/o Info (back)
Short	60165.4129	86761.8653	59723.3446	79918.5642
Long	59162.0455	79866.1791	58575.1626	72661.2957
Adapt0	62908.9870	77391.6527	64982.1420	74825.1412
Adapt1	64293.4725	68142.6702	63237.3960	73619.9793
Adapt2	60818.6930	72252.9765	59380.8093	68034.3710
Adapt3	60447.7105	83779.7672	59040.0368	78702.0819

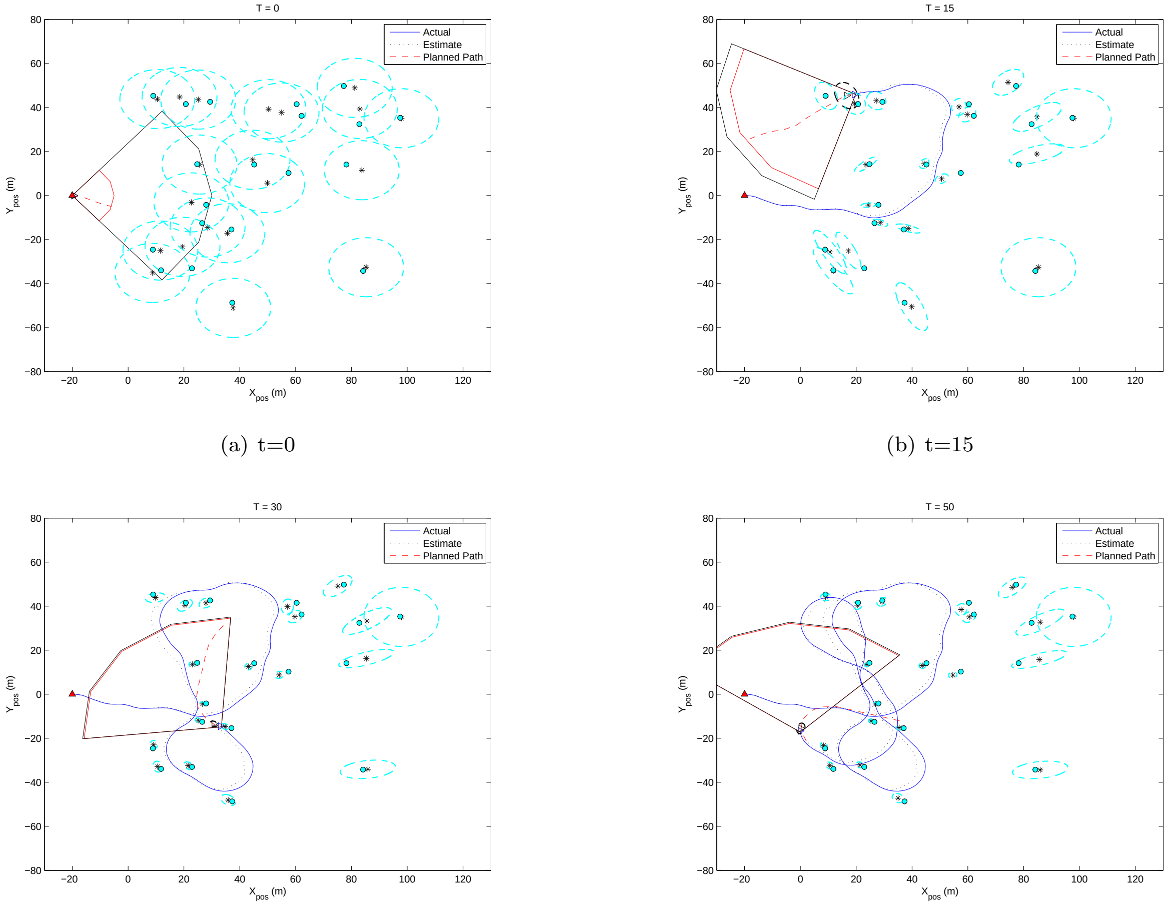
B. Exploration

The *exploration* mission is to determine the positions of uncertain targets in a given amount of time. Here the only costs are the safety cost and the information cost since there is no explicit goal location. A forest of 50 trees randomly distributed over a 300m by 200m area was used for all simulations (see Fig. 9 for a sequence of plots from a representative run). Uncertain initial positions are given for every tree (e.g. from satellite imagery taken in advance). The initial position estimates are determined by adding a random offset taken from a zero-mean normal distribution with covariance $\mathbf{P}_0 = \sigma_0^2 \cdot \mathbf{I}$ where $\sigma_0^2 = 40$m. The estimator is initialized with the covariance matrix $\mathbf{P}_0$. The vehicle start position is at (0,-15)m. Results are averaged over 100 runs. Table 5 gives the weights used in the objective function for these runs.

Table 5. Weights used in ARHC Objective Function for Exploration

$W_{u,0}$	$W_{nav,0}$	W_{goal}	W_{safe}	W_{info}
0	0	0	10^{10}	$2 \cdot 10^{-5}$

Figure 9 shows a sequence of plots from a representative run. In this example the aircraft is able to explore all but the upper right quadrant of the environment, where it is heading when the simulation runs. Figure 10 shows the information cost for this run. Surprisingly, this cost is not monotonically decreasing even though the error in the target locations is decreasing monotonically. This is due to the fact that the information cost uses the relative position covariance. When the aircraft turns away from the target the uncertainty in vehicle position causes the relative position uncertainty to grow. This implies that the global target (and aircraft) position error covariances should be used for exploration missions while the relative error is more appropriate for navigation missions.



(a) t=0

(b) t=15

(c) t=30

(d) t=50

Figure 9. Example exploration run for Adapt1.

C. Target Intercept

The goal of *target intercept* or *target prosecution* is to bring the aircraft to one of the target locations whose position is being estimated during the run. This is accomplished using the navigation costs and setting the

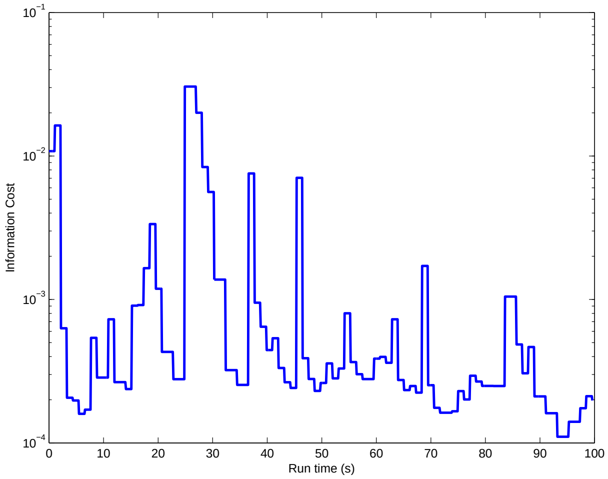


Figure 10. Exploration (information) cost versus time for typical run.

goal location to the target location, i.e. $\mathbf{p}_{goal} = \hat{\mathbf{p}}_o$. In these simulations we only consider a single target obstacle which we want to hit. Although this configuration causes the absolute position errors of the aircraft and target to diverge relatively quickly, this divergence is correlated and the relative position errors are still reduced by the SLAM filter during the run.

Table 6 summarizes the results of the target intercept scenario using a combination of information and navigation costs. The four different cases use the same adaptive parameter values of $w_1 = 0.3$, $w_2 = 1.0$, and $w_3 = -1.0$. The aircraft starts at$[0, -15]$m, the target has initial position $[100, 50]$m, and the camera range is 110m. The simulation is run for 13 seconds and the closest approach point is reported. Figure 11 shows the resulting paths for the four different cases. Because information is optimized when the aircraft is over the target, the information cost only (Case1) still results in interception of the target. Increasing the navigation cost straightens the trajectory until it heads straight at the target (Case4) with no information gathering maneuvers. Since only one result is presented for each case, the trajectories do not capture the random distribution of the ARHC performance for this scenario. However, the results still validate use of the ARHC approach for target prosecution.

Table 6. Target prosecution results

	$W_{info,0}$	$W_{nav,0}$	CAP [m]	trace($\mathbf{P}_{final}$) [m^2]
Case1	$2 \cdot 10^{-5}$	0	1.087732115	33.80919093
Case2	$2 \cdot 10^{-5}$	$3.7 \cdot 10^{-7}$	2.293602561	80.99330756
Case3	$2 \cdot 10^{-5}$	$3.7 \cdot 10^{-5}$	5.791082359	172.9369641
Case4	$2 \cdot 10^{-5}$	$3.7 \cdot 10^{-3}$	0.175401541	158.5912546

D. Cooperative Standoff Tracking

The final mission scenario is *cooperative standoff tracking* in which two aircraft follow a moving target while maintaining a specified distance from the target. Here we are interested in reducing the uncertainty in the target position estimate and we use the same weights as the exploration example. Since the SLAM algorithms described above only apply to estimation of static targets by a single vehicle, we must make several simplifications here. Rather than performing estimation, we use the true target state and the Fisher Information in place of the estimated values. This simplification allows us to focus solely on the planning algorithms for this scenario.

The ground target is assumed to move at a constant-velocity with no acceleration (i.e. no process noise).

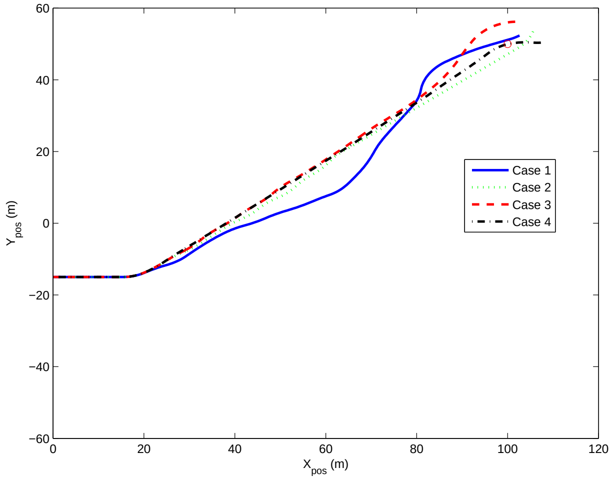


Figure 11. Final paths for target intercept scenario using an adaptive horizon.

Since this model is deterministic the expectation in the measurement term (Equation 24) of the FIM is removed and we use the recursion of Equation 23 with $\mathbf{Q}_k = \mathbf{0}$ and

$$\mathbf{\Phi}_k = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{40}$$

where the target state is now $\hat{\mathbf{x}}_v = [x, y, \dot{x}, \dot{y}]$.

The cooperative tracking task is an example of a distributed system with coupled dynamics (the target uncertainty) and coupled constraints (collision avoidance). The latter coupling is ignored here by assuming the vehicles use altitude separation. In order to address the dynamic coupling, we use a cascaded control approach. Rather than solving the full state optimization (which would require centralization) each aircraft solves its own local ARHC optimization. The cascade approach lets the first aircraft plan its path ignoring the second one, while the second aircraft plans its trajectory *given* the planned trajectory of the first. In other words

$$\mathbf{u}^*_{1,k,k+T_h} = \arg\min J_{rhc}(\mathbf{x}_{1,k}, \mathbf{u}_{1,k,k+T}), \tag{41}$$

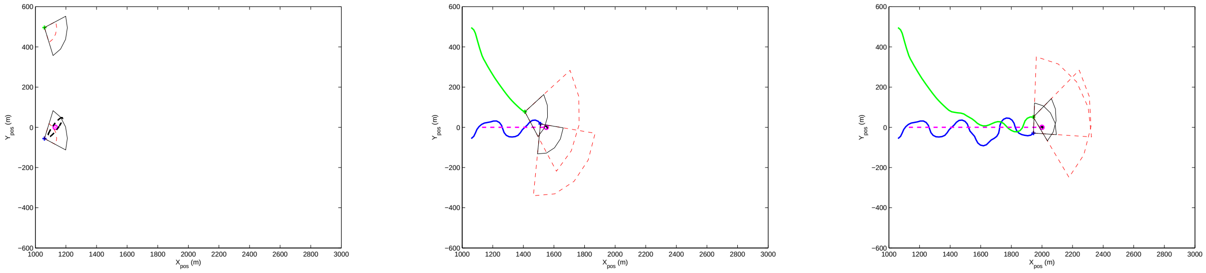
and

$$\mathbf{u}^*_{2,k,k+T_h} = \arg\min J_{rhc}(\mathbf{x}_{2,k}, \mathbf{u}_{2,k,k+T} | \mathbf{x}_{1,k}, \mathbf{u}_{1,k,k+T}). \tag{42}$$

Cooperative behavior emerges from this approach since the objective functions are still coupled by the target uncertainty. Both vehicles use the same errror covariance matrix (and therefore also calculate the same adaptive horizons) and the second vehicle includes the first vehicles contribution to the FIM of the target while determining its trajectory. Standoff tracking is achieved using the safety cost (either fixed-radius or variable-size) to prevent the aircraft from approaching too close to the target.

For the simulation run here the aircraft started at initial conditions $\mathbf{x}_1 = [1040, -60]$m and $\mathbf{x}_2 = [1040, 500]$m. The target begins at $[1100, 0]$m, moves with constant velocity $[15, 0]$m/s, and has initial position and velocity variances of $\sigma_p^2 = 10\text{m}^2$ and $\sigma_v^2 = 1\text{m}^2/\text{s}^2$, respectively. The adaptive parameter values used here were $w_1 = 0.5333$, $w_2 = 2.6667$, and $w_3 = -10.0$. The entire simulation was run for 100s.

Figure 12 shows a sequence of plots from the simulation run. The field of view limitations are not explicitly considered in the planning process, but since no information is gained when the target leaves the field of view, use of the information cost still keeps the target in view most of the time. Figure 13 shows the performance of the ARHC planner during the simulation by overlaying the information cost and planning horizon as functions of time. The initial uncertainty decreases quickly so the planning horizon is short at the beginning of the run. As the estimate improves and the uncertainty reaches steady state, the planning



(a) t=2

(b) t=30

(c) t=60

Figure 12. Cooperative standoff tracking simulation.

horizon becomes long. In this example position information can be lost when the target leaves the field of view of both cameras due to the uncertain velocity, so the information cost does not monotonically decrease. Because the uncertainty grows when the target leaves the field of view, the planning horizon will shrink when this happens. Unlike the navigation and exploration tasks, this could be detrimental to mission performance since the system needs to turn back toward the target as quickly as possible.

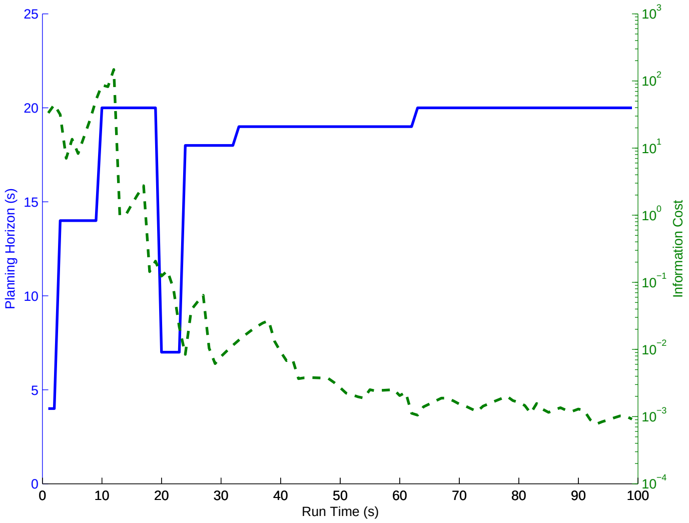


Figure 13. Final trajectory cost versus total computational effort for exploration.

VI. Conclusion

This paper has presented an integrated system for navigation using vision as the primary exteroceptive sensor. Bearings-only navigation has an inherent dynamic observability property such that specific motion is needed in order to determine the relative positions between the camera and obstacles. Thus, the estimates of new obstacles are always highly uncertain and information about the obstacles is dependent on the relative motion between the camera and objects. We present an adaptive receding horizon planner that incorporates information metrics explicitly into the receding optimization cost function. The adaptive horizon is based on the intuition that when the information about the world is rapidly changing, planning does not need to be long. 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 demonstrated the validity of the adaptive horizon approach by comparing the adaptive algorithm to short and long fixed horizon controllers. As expected, the computational effort required by the adaptive controller always fell between the short and long horizon cases. For the baseline navigation experiment with fixed-size obstacles the adaptive case also led to trajectories with lower (better) final cost. This supports the claim that the adaptive horizon yields more accurate estimates of the cost-to-go of the system over the course of a run. Validation for the other navigation scenarios was complicated by outliers in the data due to deadend paths in the planning space. The adaptive receding horizon controller was also applied to several other missions including exploration, target intercept, and cooperative tracking. Although performance was not always ideal for these examples, the simulations showed that the adaptive receding horizon control framework integrates a wide variety of missions into a single guidance layer control system.

References

[1] Langelaan, J. and Rock, S., "Towards autonomous UAV flight in forests," *AIAA Guidance, Navigation, and Control Conference*, Aug. 2005, AIAA-2005-5870.

[2] Langelaan, J. and Rock, S., "Navigation of small UAVs operating in forests," *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, Vol. 3, American Institute of Aeronautics and Astronautics, Providence, RI, United States, August 16-19 2004, pp. 2031–2041.

[3] Thrun, S., Burgard, W., and Fox, D., "Real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," *ICRA 2000: IEEE International Conference on Robotics and Automation*, Vol. 1, Institute of Electrical and Electronics Engineers, San Francisco, CA, USA, Apr 24-Apr 28 2000, pp. 321–328.

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

[5] Makarenko, A. A., Williams, S. B., Bourgault, F., and Durrant-Whyte, H. F., "An Experiment in Integrated Exploration," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Piscataway, New Jersey, October 2002.

[6] Frew, E. W., *Observer Trajectory Generation for Target-Motion Estimation Using Monocular Vision*, Ph.D. thesis, Stanford University, August 2003.

[7] Huang, S., Kwok, N. M., Dissanayake, G., Ha, Q. P., and Fang, G., "Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity," *IEEE International Conference on Robotics and Automation*, Piscataway, New Jersey, April 2005.

[8] Nygards, J., Skoglar, P., Karlholm, J., Bjorstrom, R., and Ulvklo, M., "Towards Concurrent Sensor and Path Planning," Tech. rep., FOI, May 2005.

[9] Bertuccelli, L. F. and How, J. P., "Bayesian forecasting in multi-vehicle search operations," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2006-6460, American Institute of Aeronautics and Astronautics, Reston, Virginia, August 2006.

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

[11] Watkins, A., Kurdila, A., Prazenica, R., and Wiens, G., "RHC for vision-based navigation of a MWR in an urban environment," *AIAA Guidance, Navigation, and Control Conference*, Aug. 15 - 18 2005.

[12] Schouwenaars, T., How, J., and Feron, E., "Receding horizon path planning with implicit safety guarantees," *Proceedings of the 2004 American Control Conference (ACC)*, Vol. 6, Institute of Electrical and Electronics Engineers, Boston, MA, United States, Jun 30-Jul 2 2004, pp. 5576–5581.

[13] Frew, E. W., Langelaan, J., and Joo, S., "Adaptive Receding Horizon Control for Vision-Based Navigation of Small Unmanned Aircraft," *Proceedings 2006 American Control Conference*, June 2006.

[14] Julier, S., Uhlmann, J., and Durrant-Whyte, H. F., "New method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, 2000, pp. 477–482.

[15] Merwe, R. D. V., Wan, E. A., and Julier, S. I., "Sigma-point kalman filters for nonlinear estimation and sensor-fusion - Applications to integrated navigation," *AIAA Guidance, Navigation, and Control Conference*, Vol. 3, American Institute of Aeronautics and Astronautics, Providence, RI, United States, Aug 16-19 2004, pp. 1735–1764.

[16] Bertsekas, D. P., "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European Journal of Control*, Vol. 11, No. 4-5, 2005, pp. 310 – 334.

[17] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality," *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814.

[18] Tichavsky, P., Muravchik, C., and Nehorai, A., "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, 1998, pp. 1386 – 96.

[19] Hernandez, M. L., "Optimal sensor trajectories in bearings-only tracking," *Proceedings of the Seventh International Conference on Information Fusion*, Vol. 2, Stockholm, Sweden, Jun 28 - Jul1 2004, pp. 893–900.

[20] Frew, E. W., "Sensitivity of Cooperative Geolocalization to Orbit Coordination," *AIAA Guidance, Navigation, and Control Conference*, August 2007.