Information-theoretic tracking control based on particle filter estimate

Allison Ryan*, *University of California, Berkeley, CA, 94720, United States*

The contribution of this work is a control formulation for a mobile sensor to track a target using an information-theoretic cost function based on a particle filter estimate of the target state. The particle filter representation fully models the non-linearity and limited field of view of the sensor and is able to search for a lost target by updating the estimate to eliminate areas which have been searched. An entropy calculation is developed which reflects the uncertainty of the particle filtering density for the purpose of tracking, and is then combined with a sampling method to predict the expected entropy of the target state estimate under a proposed control.

When sensor motion is constrained, such as for a fixed-wing aircraft, a long planning horizon can provide better performance than single step planning approaches. Exact prediction of information-theoretic costs for non-linear models is not generally feasible in real time, and so approximate methods will be required to predict the expected estimate entropy for receding horizon control. Simulation results demonstrate the accuracy of the prediction method and the effectiveness of the information-theoretic control. Initial experimental results verify the appropriateness of the particle filter for tracking a mobile target from an unmanned aircraft.

Nomenclature

$(\ )_k$	time subscript
x_k, X_k	target state and domain
z_k, Z_k	observation and domain
y_k	sensor platform state
u_k	sensor platform control input
$p(x_{k+1}\|x_k)$	target motion model as state transition probability
$p(z_k\|x_k)$	sensor model, conditional probability
$z_k = h(x_k, y_k)$	sensor model, stochastic function
$y_{k+1} = g(y_k, u_k)$	platform motion model
$p(X_k\|z_1, \ldots, z_k)$	state estimate distribution conditioned on all observations
$w_k^i,\ x_k^i$	weight and state of particle i, time k
$\mathcal{N}(x; \mu, \Sigma)$	normal distribution on x with mean μ and covariance Σ
$H(p(X))$	entropy of distribution
$\mathbf{H}(p(X_k\|Z_1, \ldots, Z_k))$	expected entropy of filtering density
z_k^s	a sample drawn from $p(Z_k)$

I. Introduction

Information-theoretic control for tracking seeks to minimize uncertainty in the estimate of the target state by controlling sensor position. Such a formulation explicitly incorporates probabilistic target motion and sensor models and is therefore especially suited for use with economical sensors or when the target cannot be continuously observed.

*Graduate student researcher, Center for Collaborative Control of Unmanned Vehicles, 2105 Bancroft way

Much of the extensive previous tracking literature is based on deterministic formulations. For example Spry et. al.[1] formulate a control specifically for a fixed wing unmanned air vehicle (UAV) with a downward looking camera to optimally revisit a target with constant velocity which is less than the UAV's stall speed. This control does not model uncertainty in the position of the target and so is not robust to unexpected target maneuvers. It is also based on implicit sensor modeling (outside of the control design framework), and so is not adaptable to changes in sensor mounting.

Probabilistic tracking formulations often assume that the target will be observed continuously. The most computationally efficient algorithms are based on Kalman filtering, but are limited in their ability to model non-linear and/or range-limited sensors. Campbell and Wheeler[2] include extensive modeling of uncertainty in both target motion and sensor observations, but assume that the presence of a gimballed camera system will guarantee availability of detections. A sigma point filter representation allows non-linear modeling, but is not able to update the estimate when the target is not detected. Kalman filter-based approaches are also common in multiple target tracking applications with probabilistic data association,[3] which require extremely efficient representation of a single target track due to the additional complexity of the data association problem.

A general theme in probabilistic tracking control is the balance between computational efficiency and the need to model a non-Gaussian, possibly multi-modal target estimate distribution. This need arises from the fact that when the sensor platform is kinematically constrained and the target motion is either very agile or uncertain, the target may be lost. In order to search for and reacquire the target, observations in which it does not appear must be included in the estimate. Such 'no-detection' observations cause the estimate to become extremely non-Gaussian, ruling out Gaussian-based filtering techniques such as the Kalman filter, extended Kalman filter, and sigma point filter.

Webb and Furukawa[4] develop a search and tracking controller for a camera mounted on a robot manipulator which represents the target estimate distribution by discretization over a fixed grid, and is therefore able to include the 'no-detection' observation. The objective is to maximize the probability of detecting the target over a single-step horizon. The tradeoff between modeling and computational efficiency is seen from the fact that the computational cost of calculating the control grows with the density of the grid, limiting the accuracy of the estimate. The fixed grid is appropriate in this case because the search area is small, fixed and known a priori. For tracking mobile targets, Lavis et. al.[5] suggest a reconfigurable mesh which expands to contain the target's forward reachable set and thereby maintain the validity of the estimate PDF. Mesh elements which no longer contribute to the PDF are removed, allowing the search space to move with the target rather than growing indefinitely. However, this is quite computationally intensive.

The tradeoff between flexible representation and computational efficiency will be addressed in this paper by choosing a particle filter representation for the target state estimate. The particle filter is able to represent non-linear models and non-Gaussian multi-model estimate distributions by moving particles to areas of high probability. The advantages of particle filtering for probabilistic tracking are well-known, but the interpretation of a particle set representation for information-theoretic control is an area of current research. Particle filtering will be further discussed in section II-B.

In section II the information-theoretic control formulation is developed in two parts: control and estimation. The control formulation concentrates on derivation of an appropriate objective function for the sensor platform motion control. The estimation section introduces the particle filter. Section III addresses real-time feasible methods for approximating the objective function. Simulation and experimental results are presented in section IV regarding both accuracy of the approximations from section III and control performance, followed by conclusions in section V.

II. Control and Estimation Formulation

In traditional deterministic tracking formulations[1,6] the output of the sensing algorithm is an estimate of the target position. This estimate is regarded as truth from the standpoint of control and the desired sensor platform position is calculated based on heuristics which attempt to produce continued observations. For example, the heuristic for the downward-looking camera on a fixed-wing UAV is to fly over the target position with zero roll angle. The control objective in the traditional tracking formulation depends only on the sensor platform state and the best estimate of the target state, as shown in figure 1.

The information-theoretic control formulation is distinguished by the fact that the objective function is an information measure on the target state estimate distribution. Rather than separating the estimation

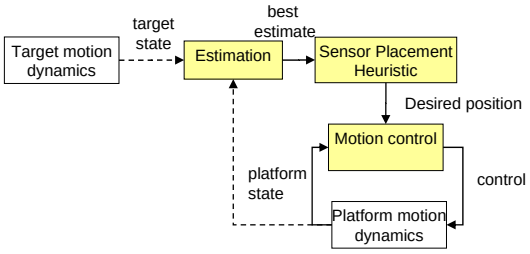

Figure 1. Traditional tracking control loop

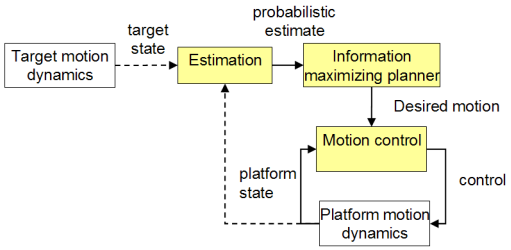

Figure 2. Information-theoretic tracking control loop

and control aspects of the system by controlling based on the best estimate (a certainty equivalence control), the control is based on the full probability density function (PDF) of the target estimate and its relation to platform motion, as shown in figure 2. Probabilistic models of target motion, sensor information gain, and sensor platform motion are explicitly included. Modeling the effect of sensor platform control on the evolution of the estimate PDF is challenging due to the many introduced uncertainties and the non-linear nature of range-limited sensors. Andrieu et. al. discuss the complexity of this type of control, as well as the application of sampling methods, in a very general formulation based on a partially observable Markov decision process.[7]

The notation for estimation and control will be based on variables x related to the unknown target state, y related to the platform state, and sensor observations z. The target state x_k at discrete time k is an unknown random vector in the continuous domain X_k. The estimate of x_k is a conditional PDF $p(X_k|z_1,\ldots,z_k)$ defined on X_k, also referred to as the filtering density. The observation z_k is defined on the domain Z_k, which includes the observation $z_k = \emptyset$, in which the target is not detected. The stochastic observation model $z_k = h(x_k, y_k)$ is a function of both the current target state and the current sensor platform state y_k, and generates the conditional distribution $p(Z_k|X_k)$. The platform motion is governed by a known deterministic model $y_{k+1} = g(y_k, u_k)$, where u_k is the control input. The goal considered in this work is to minimize the filtering density uncertainty.

Estimate uncertainty will be measured using information entropy as defined by Shannon,[8] which represents the uncertainty or randomness in the estimate of x.

$$H(p(X)) = -\int_X p(x)\log p(x)dx \tag{1}$$

$$\mathbf{H}(p(X|Z)) = \int_Z p(z)H(p(X|z))dz \tag{2}$$

The uncertainty measure on the estimate PDF will be defined by the expected entropy, $\mathbf{H}(p(X_k|Z_1,\ldots,Z_k))$.

A. Receding Horizon Information-Theoretic Control

The tracking objective is to provide an accurate estimate of the target position at all times. All knowledge of the current target location at time k is represented by the filtering density $p(X_k|z_1,\ldots,z_k)$ provided by any appropriate filter implementation. For a finite time interval from $k=0$ to $k=N-1$ an optimal control sequence $[u_0,\ldots,u_N]$ solves the following minimum entropy problem.

$$[u_0^*,\ldots,u_{N-1}^*] = \arg\min \sum_{k=1}^{N} \mathbf{H}(p(x_k|Z_1,\ldots,Z_k)) \tag{3}$$

$$\text{subject to}$$

$$y_{k+1} = g(y_k, u_k)$$

$$z_k = h(x_k, y_k) \qquad k = 0,\ldots,N$$

However, real-time optimization over the duration of a task is not generally feasible, and a receding horizon approach is often selected. In some cases, stability and performance guarantees can be produced for a

receding horizon optimization by development of a Lypunov function based on the cost. Although there is a large body of work in this area,[9] it will not be attempted in the information-theoretic formulation due to the complexity of the cost function.

A common approach in information-theoretic control problems is model-based receding horizon control with a horizon of one step.[10–12] This approach is referred to as information surfing because the sensor platform follows the local gradient of information gain. However, a longer planning horizon has the potential to provide better performance for sensors with limited range and constrained motion, especially when non-minimum phase characteristics are present. For example, to move the downward-looking sensor footprint of a fixed-wing UAV to the right, the footprint must first move left due to the aircraft bank angle dynamics. This behavior is not accounted for by a single-step plan. Therefore, we consider a receding horizon of $T-1$ control steps.

At the current time k, the entropy of the conditional distribution $p(X_{k+i}|z_1,\ldots,z_{k+i})$ depends on the (unknown) future observations $\{z_{k+1}\ldots z_{k+i}\}$. Thus, the control objective must be considered in expectation over the sequence of future observations $\{Z_{k+1},\ldots,Z_{k+T}\}$. The control objective considered over the remainder of the paper is to minimize the expected entropy of the estimate distribution over the receding horizon.

$$u^* \equiv [u_k^*,\ldots,u_{k+T-1}^*] = \begin{array}{c} \arg\min \\ \text{subject to} \\ y_{k+1} = g(y_k,u_k) \\ z_k = h(x_k,y_k) \end{array} J_k = \sum_{i=k+1}^{k+T} \mathbf{H}(p(X_i|z_1,\ldots,z_k,Z_{k+1},\ldots,Z_i)) \tag{4}$$

For a single sensor platform tracking a single target, the cost function can be minimized using non-linear gradient search algorithms such as sequential quadratic programming.[13] Due its complexity, it is assumed that the cost function will be minimized using an appropriate optimization package using repeated function evaluations for numerical approximation of the gradient. Therefore, the cost function must be calculated efficiently for real-time optimal control.

B. Estimation Using Particle Filter

The proposed control formulation assumes the presence of a target state estimate PDF. A Sampling Importance Resampling (SIR) particle filter[14] is chosen in order to represent arbitrary target estimate PDFs and non-linear sensor and target motion models. The particle filter is a sub-optimal Bayes estimator based on sequential Monte Carlo sampling, and is very popular for target tracking and simultaneous localization and mapping applications.[15,16] Unlike a grid or mesh representation which must either extend over all possible target states or be reconfigured outside of the filtering algorithm,[5] the particle set efficiently moves to follow the target as a consequence of the filtering process.

In an optimal recursive Bayes filter, the prediction step applies a motion model $p(X_{k+1}|X_k)$ to the current estimate PDF $p(X_k|z_1,\ldots,z_k)$ via the Chapman-Kolmogorov equation.

$$p(x_{k+1}|z_1,\ldots,z_k) = \int_{X_k} p(x_{k+1}|x_k)p(x_k|z_1,\ldots,z_k)dx_k \tag{5}$$

The estimate is then updated according to observation z_{k+1} using Bayes rule and the sensor model $p(Z_k|X_k)$.

$$p(x_{k+1}|z_1,\ldots,z_{k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|z_1,\ldots,z_k)}{p(z_{k+1}|z_1,\ldots,z_k)} \tag{6}$$

Recursion of the prediction and update steps form the optimal estimate by providing the conditional distribution $p(X_k|z_1,\ldots,z_k)$ from the sequence of observations from time 1 to k. The recursion starts from a prior distribution $p(X_0)$ which includes prior knowledge of the initial target state. The optimal filter can be implemented exactly in restricted variations such as the Kalman filter or estimation of on a discrete domain with a finite number of possible values.[14] For arbitrary models, the optimal filter equations cannot be solved analytically, but could be implemented approximately using quadrature techniques. However, more efficient methods are required for real-time filtering and control.

A particle set represents a distribution $p(X)$ as a set of N weighted particles in the form of a sum of Dirac delta functions located at the particle locations $\{x^i\}$, where each particle represents a hypothesis from the state space.

$$p(x) = \sum_{i=1}^{N} w^i \delta(x - x^i) \tag{7}$$

The particle filter approximates optimal recursive Bayes filtering by representing the distributions $p(X_k|z_1, \ldots, z_k)$ and $p(X_{k+1}|z_1, \ldots, z_k)$ as sets of particles and applying the prediction and update steps (5,6). The prediction step of the filter is implemented by resampling the particle locations according to the target motion model. The new location x^i_{k+1} of particle i is a random sample drawn from the distribution $p(x_{k+1}|x^i_k)$ conditioned on the current particle location x^i_k. The update step is implemented by updating the particle weights using Bayes rule. The resampling step does not follow directly from the optimal Bayes filter, but is required to avoid particle set degeneracy[14] and has the effect of eliminating very unlikely particles and replicating likely particles.

Although the particle filter is well-suited for estimating target motion, it is a challenging representation for calculating information measures which will form the basis for control.

III. Approximation of Objective Function

The current cost J_k consists of expected entropy terms $\mathbf{H}(p(X_i|z_1, \ldots, Z_i))$ for future times i. The observations $\{z_1, \ldots, z_k\}$ are known, and the expectation is taken over the future observations $\{Z_{k+1}, \ldots, Z_i\}$. The cost is expanded to explicitly show the expectation over future observations.

$$J_k = \sum_{i=k+1}^{k+T} \int_{Z_{k+1},\ldots,Z_i} p(z_{k+1}, \ldots, z_i|z_1, \ldots, z_k) H(p(X_i|z_1, \ldots, z_i)) dz_{k+1} \ldots dz_i \tag{8}$$

As described in the previous section, selection of an optimal control requires evaluations of the cost function for control sequences $[u_k, \ldots, u_{k+T}]$.The expectation over $\{Z_{k+1}, \ldots, Z_i\}$ cannot be calculated exactly due to lack of analytical expression for the joint distribution $p(Z_{k+1}, \ldots, Z_i|z_1, \ldots, z_k)$. To calculate an expectation over a single observation, one could partition the domain Z_{k+1} into N_z discrete regions, calculate their respective probabilities, and evaluate the entropy conditioned on a representative sample from each region. However, the cost of this method grows exponentially with the length T of the observation sequence as N_z^T, and is therefore not feasible for real-time control. It is extremely inefficient because data points are drawn uniformly from the observation domain.

Instead, we approximate the expectation by drawing random samples from the distribution $p(Z_{k+1}, \ldots, Z_i|z_1, \ldots, z_k)$ and replacing the expectation with the sample mean. This method allows a multiple step expected entropy prediction for tracking a moving target with a non-linear sensor, which has not been seen in literature review. The method for drawing random samples from the future observation sequence is described in section A and the entropy calculation for a particular sample is developed in section B.

A. Expected entropy calculation by drawing samples from predicted observation sequence

Each expected entropy term $\mathbf{H}(p(X_i|z_1, \ldots, Z_i))$ in the cost function will be replaced by a sample mean based on random samples from $p(Z_{k+1}, \ldots, Z_i|z_1, \ldots, z_k)$: observations between the current time and the time i. The sampling procedure will make use of the hidden Markov problem structure shown in figure 3, where state x_{k+1} is conditioned only on state x_k and observation z_k is conditioned only on state x_k. A single sample from $p(Z_{k+1}, \ldots, Z_i|z_1, \ldots, z_k)$ is generated using the chain rule by first drawing a sample z^s_{k+1} from $p(Z_{k+1}|z_1, \ldots, z_k)$, followed by a sample z^s_{k+2} from $p(Z_{k+2}|z_1, \ldots, z_k, z^s_{k+1})$, and so on. The hidden Markov structure shows that the unknown state x must be marginalized at each step in the sample generation.

$$\begin{aligned} p(Z_{k+1}, \ldots, Z_i|z_1, \ldots, z_k) &= p(Z_{k+1}|z_1, \ldots, z_k) \ldots p(Z_i|z_1, \ldots, Z_{i-1}) \\ p(Z_i|z_1, \ldots, z_{i-1}) &= \int_{X_i} p(Z_i, x_i|z_1, \ldots, z_{i-1}) dx_i = \int_{X_i} p(Z_i|x_i) p(x_i|z_1, \ldots, z_{i-1}) dx_i \end{aligned} \tag{9}$$

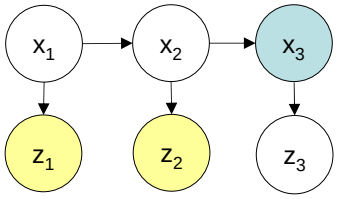

Figure 3. Hidden Markov model for target state and observations. Node x_3 is marginalized to calculate $p(Z_3|z_1, z_2)$ when z_1 and z_2 are observed.

The conditional state estimate $p(X_i|z_1, \ldots, z_{i-1})$ is obtained from a particle filter estimate conditioned on the preceding sampled observations; essentially a simulation of the estimation process. This process is outlined in Algorithm 1, which draws a single random sample from $p(Z_{k+1}, \ldots, Z_{k+T}|z_1, \ldots, z_k)$ and calculates the associated sample cost J_k^s.

$$\begin{aligned} J_k &\approx \sum_{i=k+1}^{k+T} \frac{1}{N_s} \sum_{s=1}^{N_s} H(p(X_i|z_1^s, \ldots, z_i^s)) = \frac{1}{N_s} \sum_{s=1}^{N_s} J_k^s \\ J_k^s &= \sum_{i=k+1}^{k+T} H(p(X_i|z_1^s, \ldots, z_i^s)) \end{aligned} \tag{10}$$

Algorithm 1 Sample generation from observation sequence of length T

1: Begin with particle set representation of prior $p(X_k|z_1, \ldots, z_k)$
2: i = 1
3: **while** $i \leq T$ **do**
4: Predict particle filter. Result is $p(X_{k+i}|z_1, \ldots, z_{k+i-1}^s)$.
5: Sample z_{k+i}^s from $p(Z_{k+i}|z_{k+1}^s, \ldots, z_{k+i-1}^s)$ using (9).
6: Update particle filter based on sample observation z_{k+i}^s using (6). Result is $p(X_{k+i}|z_1, \ldots, z_{k+i}^s)$.
7: Calculate entropy from particle set. Result is $H(p(X_{k+i}|z_1, \ldots, z_{k+i}^s))$.
8: Resample particle filter
9: i = i+1
10: Sample from observation sequence is $[z_{k+1}^s, \ldots, z_{k+T}^s]$
11: Sample cost is $J_k^s = H(p(X_{k+1}|z_1, \ldots, z_{k+1}^s)) + \ldots + H(p(X_{k+T}|z_1, \ldots, z_{k+T}^s))$

The two challenges in Algorithm 1 are the entropy calculation in line 7 and drawing the sample in line 5. A sample is required from $p(Z_{k+i}|z_k, \ldots, z_{k+i-1})$, which is written in terms of the sensor model $p(Z_k|X_k)$ and the particle filter estimate conditioned on the previous estimations. Substituting the particle set representation from (7) replaces an integral over the target state domain with a summation over N particles, where the particle filter locations and weights reflect all preceding observations.

$$p(z_k|z_1, \ldots, z_{k-1}) = \sum_{i=1}^{N} w_k^i p(z_k|x_k^i) \tag{11}$$

Many sensor models can be written in the form $p(z|x) = \mathcal{N}(z; \mu(x), \Sigma(x))$: a normal distribution with mean and covariance dependent on x. In this case, the distribution $p(Z_k|z_1, \ldots, z_{k-1})$ is a Gaussian mixture model (12), for which sampling techniques exist.[17]

$$p(z_k|z_1, \ldots, z_{k-1}) = \sum_{i=1}^{N} w_k^i \mathcal{N}(z_k; \mu(x_k^i), \Sigma(x_k^i)) \tag{12}$$

Having drawn N samples from $p(Z_{k+1}, \ldots, Z_{k+T}|z_1, \ldots, z_k)$ and calculated the resulting entropies, the result is N samples from the distribution of $\sum \mathbf{H}(p(X_i|Z_1, \ldots, Z_i))$. The expectation μ of the true distribution has been defined as the cost function J_k. Under proper (independent identically distributed) sampling

conditions, it is well-known that the sample mean μ_N is an unbiased estimate of μ and with standard deviation $\hat{\sigma}_N = \sigma_N/\sqrt{N-1}$, where σ_N is the standard deviation of the sample set. A similar analysis can be used to predict 3σ bounds for the realized conditional entropy relative to the predicted expected entropy. These bounds are much wider because the uncertainty in the realization of each observation does not decrease with the number of samples, as does the uncertainty in the cost prediction. Uncertainty bounds due to sampling are shown in the simulations section.

The 3σ error bound for the true cost relative to cost estimated by sampling is required in order to gauge the accuracy of the optimization. If the error bound is large compared to the cost difference between control choices, then the control with the lowest predicted cost may not be optimal because the selection may be dominated by the uncertainty. If a set of candidate controls $\{u^i\}$ are to be evaluated in order to select the minimizer for $J(u)$, $\hat{\sigma}_N$ should be small compared to the differences between the costs $\{J(u^i)\}$. Since this set of costs is the result of the sampling procedure, an iterative process may be required to increase the number of samples until the desired accuracy is reached.

B. Entropy from particle set representation

The second challenge in Algorithm 1 is to calculate the entropy of the filtering distribution represented by the particle set. A particle set approximates a continuous distribution $p(X_k|z_1,\ldots,z_k)$ by a set of samples (weighted dirac delta functions) such that the two distributions have similar cumulative density functions. However, the particle set and the continuous distribution are very different in the context of the entropy calculation.

An entropy calculation on any resampled particle filter will result in $H = N\log N$ due to the uniform weights on the N particles. Similarly, an entropy calculation on the particle set before resampling will depend only on the distribution of particle weights, corresponding to the most recent likelihood function $p(z_k|X_k)$ rather than the entropy of the cumulative estimate. To calculate the entropy of the cumulative state estimate, it is useful to form a continuous distribution which corresponds to the particle set representation.

Gaussian smoothing is a method for calculating information measures from a particle distribution by convolving the particle distribution with a small-covariance Gaussian kernel to produce a continuous distribution. For Gaussian motion models, the Gaussian smoothing method may produce a result similar to the one developed here. However, in Gaussian smoothing the covariance of the Gaussian kernel must be selected as a design parameter. The method presented here follows directly from the optimal Bayes filter and thereby avoids the need to tune the choice of smoothing parameters. Instead, a continuous PDF at time k can be realized very intuitively by applying optimal Bayes updates to the preceding particle set at time $k-1$.

If the filtering density $p(X_{k-1}|z_1,\ldots,z_{k-1})$ is given by the particle set $\{w^i_{k-1}, x^i_{k-1}\}$, application of optimal prediction and update steps (5,6) results in the following expression for a continuous PDF at time k.

$$p(X_k|z_1,\ldots,z_k) = \frac{p(z_k|X_k)}{p(z_k|z_1,\ldots,z_{k-1})}\sum_{i=1}^{N} w^i_{k-1}p(X_k|x^i_{k-1}) \tag{13}$$

Entropy calculation from this PDF would require numerical integration of an arbitrary function over the domain X_k. The cost of quadrature grows quickly with the dimension of the domain, and may devote large computational effort to integration over areas of small contribution. Instead, a piecewise linear approximation of (13) over appropriately chosen regions could allow entropy calculation as a sum of analytical contributions from each linear region.

In order to accurately represent the continuous distribution given by (13), the piecewise linear elements must be strategically placed. An advantage of the particle filter is that it places particles in areas of high probability, providing more detailed approximation in areas of higher contribution to the entropy calculation. Taking advantage of this, we choose the particle locations $\{x^i_k\}$ to define the vertices of triangular elements. The entropy calculation will be performed on the piecewise linear function $f(x)$ defined by interpolation between points $\{x^i_k\}$ with values $f(x^i_k)$ given by (13), where the normalization constant $p(z_k|z_1,\ldots,z_{k-1})$ has been removed. The entropy contribution from each region is now an analytical function of the vertex locations and their function values $f(x^i)$.

$$f(x^i_k) = p(z_k|x^i_k)\sum_{j=1}^{N} w^j_{k-1}p(x^i_k|x^j_{k-1}) \tag{14}$$

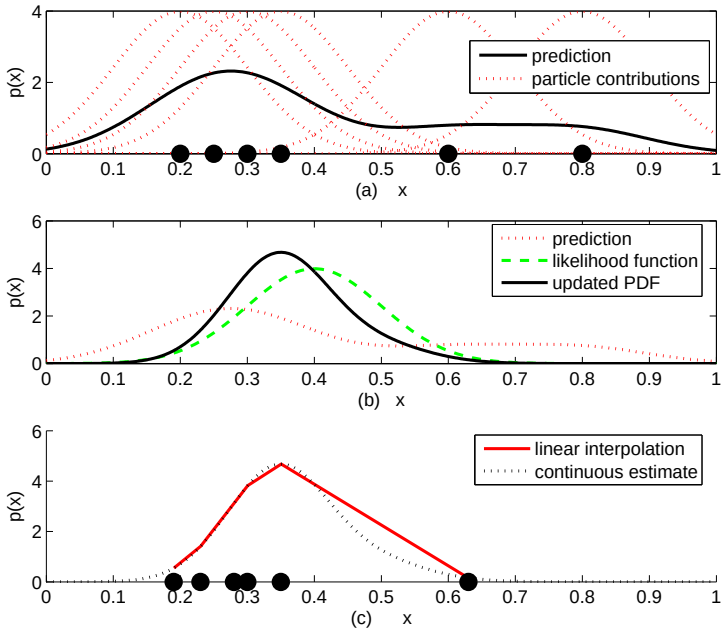

Figure 4. Illustrative example for derivation of a piecewise linear approximation of a PDF represented by a particle set. Unnormalized PDFs are shown.

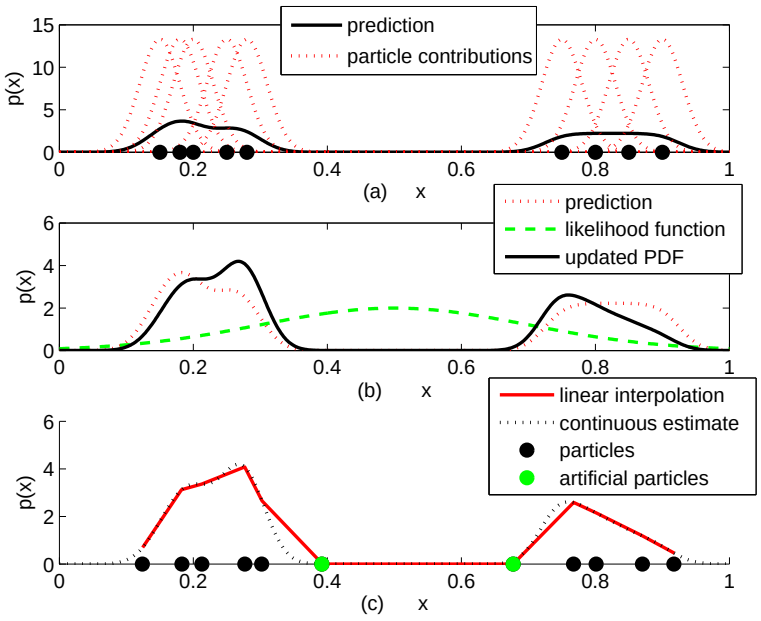

Figure 5. Example for derivation of a piecewise linear approximation of a PDF represented by a particle set, with artificial particles used to force zero-value in areas of low particle density. Unnormalized PDFs are shown.

Figure 4 illustrates an example formulation of a piecewise linear approximation of the filtering density beginning from the particle set representation of $p(X_{k-1}|z_1, \ldots, z_{k-1})$. An unrealistically small number of particles is shown for clarity. In subfigure (a), the motion model centered at each particle is summed to form the prediction $p(X_k|z_1, \ldots, z_{k-1})$. In subfigure (b), the prediction distribution is multiplied by the likelihood function $p(z_k|X_k)$ and normalized to form the updated distribution. In subfigure (c), the updated distribution is evaluated at each of the new particle locations, and the piecewise linear function is formed by interpolation.

A final detail in defining the interpolated function is required to represent areas of low particle density. Linear interpolation between particles which are far apart may not be desired. For example, in a very bimodal distribution, there may be no probability mass in the area between the two modes. Accordingly, the interpolated function should be driven to zero in locations which are far from any particle in a sense defined by the motion model. The predicted distribution consists of a sum over the particles of the target motion model centered at each particle location. The target state transition probability $p(X_k|X^i_{k-1})$ will always tend toward zero with increasing distance $||x_k - x^i_{k-1}||$ for a target with bounded velocity. Therefore, a radius $r(x)$ can be selected such that if $||x_k - x^i_{k-1}|| > r(x^i_{k-1})\ \forall i$, then $p(x_{k|k-1}) < \epsilon$. For a Gaussian motion model, this radius can be chosen to correspond to the 3σ bound. This is implemented in the linear interpolation by placing virtual particles with $f(x^i) = 0$ at distance $r(x^i)$ from the true particles, as shown in figure 5.

IV. Simulations and Experimental Results

Results from a number of simulations and a flight experiment will be presented. In simulation 1, the particle filter and corresponding entropy calculation are applied to duplicate the known solution of a problem with linear models and Gaussian noise. This confirms that the entropy calculated from the piecewise linear interpolated function is a good approximation to the entropy of the optimal Bayes estimate, in a case where such an estimate is available. In simulation 2, non-linear models are introduced which prohibit the application of a Kalman filter. A simple proportional controller based on the particle filter mean is adequate but does not use the available motion models to form an optimal plan. Simulation 3 implements an information-theoretic receding horizon control (RHC) for a simplified version of the scenario in simulation 2. The RHC produces behavior similar to the solution which has been derived in the literature from an extended Kalman filter representation. In simulation 4, information-theoretic RHC is applied for the scenario from simulation 2, resulting in slightly improved performance. Finally, tracking data from a flight experiment is presented to introduce future experimental work.

Simulation 1: Comparison to Kalman filter

The particle filter-based entropy calculation is first verified against a Kalman filter in one dimension. The target motion model is a random walk with covariance $W_m = 3$, and the sensor model is linear with covariance $W_s = 1$.

$$\begin{aligned} x_{k+1} &= x_k + w_m \\ z_k &= z_k + w_s \\ p(x_0) &\sim \mathcal{N}(0, 15) \\ w_m &\sim \mathcal{N}(0, W_m), \quad w_s \sim \mathcal{N}(0, W_s) \end{aligned} \tag{15}$$

The Kalman filter estimate entropy is calculated from the posterior estimate covariance Z by $H = \log\sqrt{2\pi e Z}$. For particle filter implementations with 250, 500, and 1000 particles, entropy is calculated as described in section B. Each trial simulates 100 filter iterations. The resulting root mean square (RMS) error is shown in Table 1.

Simulation 2: Non-linear models, proportional control

A particle filter is implemented to track a target with terrain-dependent motion using a non-linear sensor and a proportional heading rate control. When the target is on a road it has a known nominal velocity V_{nom} parallel to the road in addition to random Gaussian motion. The time discretization is Δ and the random

Table 1. Root mean square error of particle filter entropy compared to Kalman filter, where steady state entropy of Kalman Filter is 1.3

250 particles	500 particles	1000 particles
0.0557	0.0335	0.0124

velocity is V_t. When the target is not on a road $V_{nom} = 0$. This target motion corresponds to the following model.

$$p(x_0) = \text{uniform} \tag{16}$$
$$p(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + V_{nom}\Delta, \Sigma), \quad \Sigma = \begin{bmatrix} V_t\Delta & 0 \\ 0 & V_t\Delta \end{bmatrix}$$

The UAV motion is represented deterministically by a coordinated turn model with constant altitude and bounded turn rate, where the UAV state y includes position (y_x, y_y) (in coordinates aligned with the UAV heading), heading angle ψ and roll angle ϕ. The control input is $u = \dot{\psi}$ and the constant UAV velocity is V_u. This model is commonly used for control design for an autopilot-stabilized UAV.

$$y_{x,k+1} = y_{x,k} + \frac{V_u}{\dot{\psi}}\sin(\Delta\dot{\psi}) \tag{17}$$
$$y_{y,k+1} = y_{y,k} + \frac{V_u}{\dot{\psi}}(1 - \cos(\Delta\dot{\psi}))$$
$$\phi = -\arctan(\frac{V_u\dot{\psi}}{g})$$
$$\psi_{k+1} = \psi_k + \dot{\psi}\Delta, \quad |\dot{\psi}| \leq u_{\max}$$

The bearing-only sensor model (18) approximates monocular vision[18] based on a noisy measurement of the bearing from the UAV to the target. A sensor footprint $\mathcal{F}(y)$ is defined by the UAV state, modeling a fixed downward-looking camera with limited field of view angle β. The probability of detecting a target within the footprint is $1 - P_{miss}$ and the probability of detecting a target outside the footprint is zero.

$$p(z = \emptyset|x) = 1 \quad \forall x \notin \mathcal{F}(y) \tag{18}$$
$$p(z = \emptyset|x) = P_{miss} \quad \forall x \in \mathcal{F}(y)$$
$$p(z = \tilde{z} \in \mathbf{R}|x) = 0 \quad \forall x \notin \mathcal{F}(y)$$
$$p(z = \tilde{z} \in \mathbf{R}|x) = (1 - P_{miss})\mathcal{N}(\tilde{z}; x, \sigma^2) \quad \forall x \in \mathcal{F}(y)$$

This model can easily be extended to include the possibility of false detections. A more accurate model for monocular vision has also been developed and demonstrated in a two-UAV flight experiment[19] and models a resolution-dependent probability of detection. The various model constants are given in Table 2.

Table 2. Simulation parameters

V_t	5 meters per second	V_u	20 meters per second
$u_{\max}$	0.2 radians per second	g	9.8 meters per second^2
Δ	1 second	P_{miss}	0.1
β	24 degrees	σ	2.5 degrees
UAV altitude	150 meters	number of particles	500
V_{nom}	15 meters per second		

The first three plots of figure 6 show the progression of the UAV position, sensor footprint, and particle distribution through the simulation. A proportional heading rate control steers the UAV toward the expected

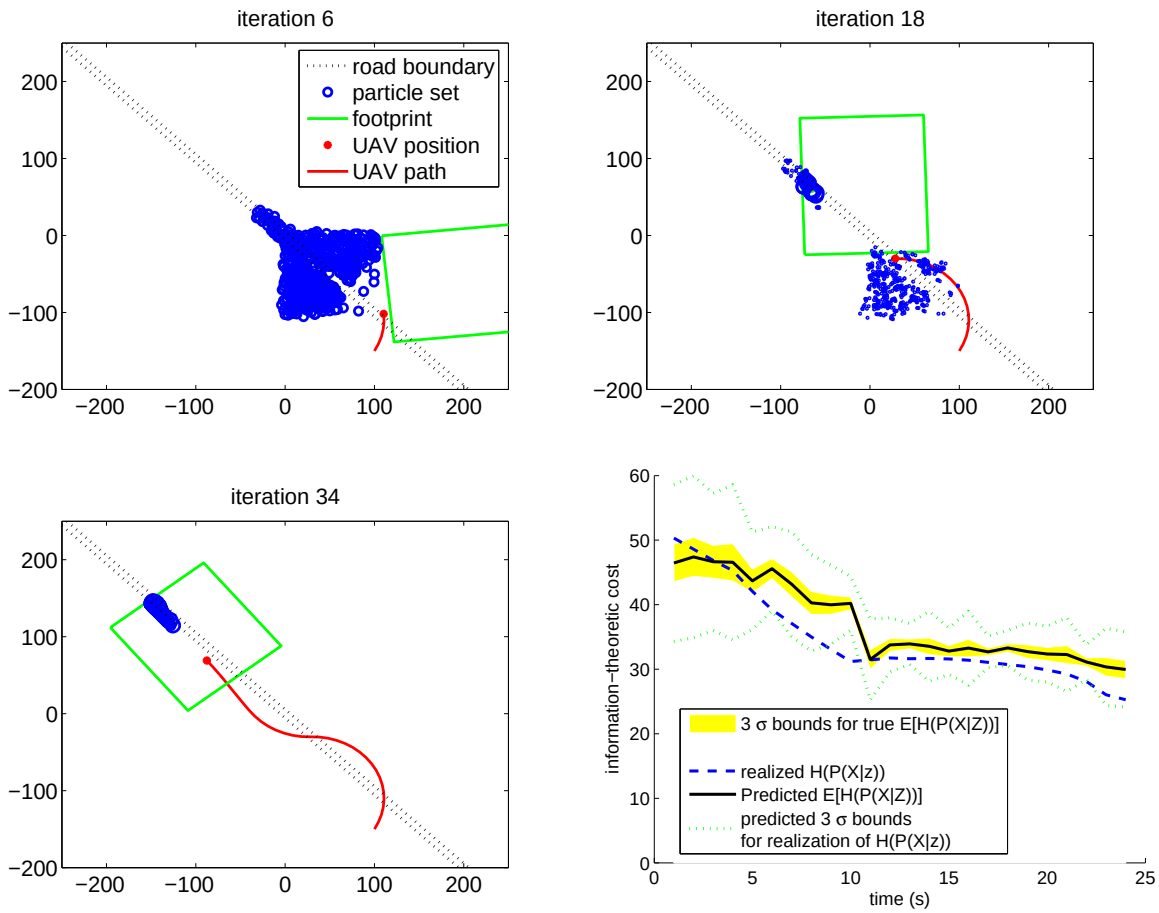

Figure 6. (Simulation 2) A simple proportional heading rate control drives the UAV toward the expected target position based on the particle filtering density. The lower left figure shows predicted and realized information-theoretic costs with associated uncertainty bounds.

target position calculated from the filtering density. The UAV footprint is seen to lie on the outside of the UAV's curved path as determined by the coordinated turn model. Iteration 6 shows how particles travel quickly along the road but randomly otherwise. Iteration 18 shows where some possible target locations have been eliminated before the target is found, resulting in a multi-modal distribution. This would not be possible in a traditional Kalman filter-based tracking formulation. The target is observed in iteration 34, and is known to lie near the edge of the sensor footprint because it was not seen in the preceding iterations.

The final plot of figure 6 shows the performance of this simulation with respect to the information theoretic objective (4), which is predicted using the methods described in section III. The objective function is a sum over the receding horizon of the expected entropy of the filtering density, referred to as predicted $E[H(P(X|Z))]$ in the figure. In this simulation, the prediction horizon is 6 steps of 1 second each. Therefore, the expected entropy plotted at time t is a sum over the expected filtering entropy for times $t+1$ to $t+7$ (seconds). In contrast with the expected entropy prediction, the realized entropy referred to in the figure is calculated after the observations have occurred. The realized entropy plotted at time t is a sum from $t+1$ to $t+7$ of the filtering entropy conditioned on the observations z_{k+1} to z_{k+7}. Calculation of the error bounds from the sample statistics was described in section III-A.

Simulation 3: RHC with random walk target model

This set of two simulations includes the same UAV and sensor model as in simulation 2 and a target model which has been slightly simplified by removing the road. The two simulations differ only by the length of the planning horizon, and show improved performance due to longer planning. The target model is as in simulation 2 (16), except $V_{nom} = 0$. A control is selected from a uniform discretization of the UAV turnrate

range to N_u values at each control step. The information-theoretic cost is calculated for each control sequence candidate, and the control with the lowest predicted cost is applied. The control choice is further discretized with respect to time by allowing each control to be held constant for a number of simulation steps N_c. This allows the planning horizon to be extended to T simulation steps with the search space reduced to N_u^{T/N_c} rather than N_u^T, but sacrifices optimality due to the addition of extra constraints. An exhaustive search method is chosen rather than an iterative method such as sequential quadratic programming because it is difficult to detect stationary points due to the stochastic cost estimate. This may not be appropriate for a multiple-input system because the dimension of the search space would grow more quickly with the horizon length.

Figure 7 shows results of the two simulations which vary only in the control horizon. In the short horizon simulation, the estimate entropy is predicted a single step (0.5 seconds) ahead. In the long horizon simulation, the entropy is predicted 2 seconds ahead to form a 2 second plan in which each control is held constant for two steps. The form of the planned control is $[u_1 u_1 u_2 u_2]$. The longer planning horizon results in a path which initially curves away from the target for better triangulation with the bearing-only sensor, and then approaches it, resulting in less estimate uncertainty than the short horizon controller. This corresponds to the known optimal behavior which can be derived from an extended Kalman filter formulation.[20]

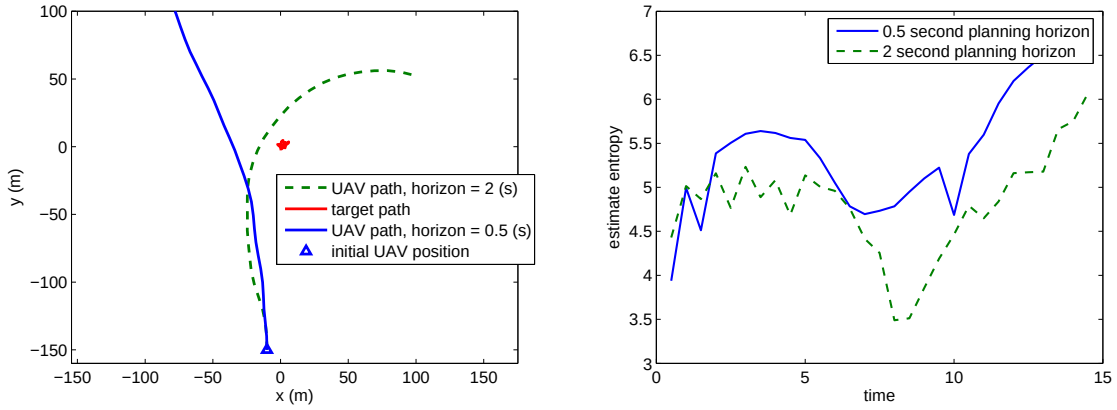

Figure 7. (Simulation 3) UAV paths generated by entropy-minimizing control for 2 second and 0.5 second planning horizons (left). Comparison of resulting estimate entropy showing improved performance with longer horizon (right).

Simulation 4: RHC with terrain-dependent target model

In this simulation, the receding horizon information-theoretic control formulated in section II is applied to track the terrain-dependent target model first presented in simulation 2, using the same sensor model. The first three plots of figure 8 show the UAV path, footprint, and particle filter estimate at various times in the simulation. The planned UAV trajectory resulting from the selected control sequence is also shown.

The lower right plot of figure 8 shows the analysis used to select the optimal control at iteration 4. For each of 25 candidate control sequences, the probability of detection and the expected information-theoretic cost are calculated. (Recall that the information-theoretic cost is expected entropy summed over the planning horizon.) Probability of detection is clearly not a useful metric because 6 of the 25 candidate controls predict certain detection. The information-theoretic cost varies much more over the control space. At iteration four, the target position is still quite uncertain, leading to large uncertainty in the expected entropy calculation. As a result, the 3 σ bounds for the cost estimate are significant compared to the variation between one control and another. In this case, it would have been advantageous to draw more observation samples in order to more reliably choose the optimal control.

Figure 9 shows the predicted and observed information theoretic costs, corresponding to the similar plot in figure 6 for simulation 2. Although these results are quite similar, the receding horizon control shows slightly improved performance. The time-averaged cost under receding horizon control is 32.0 and the time-averaged cost under proportional control is 35.2. The time-averaged uncertainties for each test are 1.7 and 1.5, respectively.

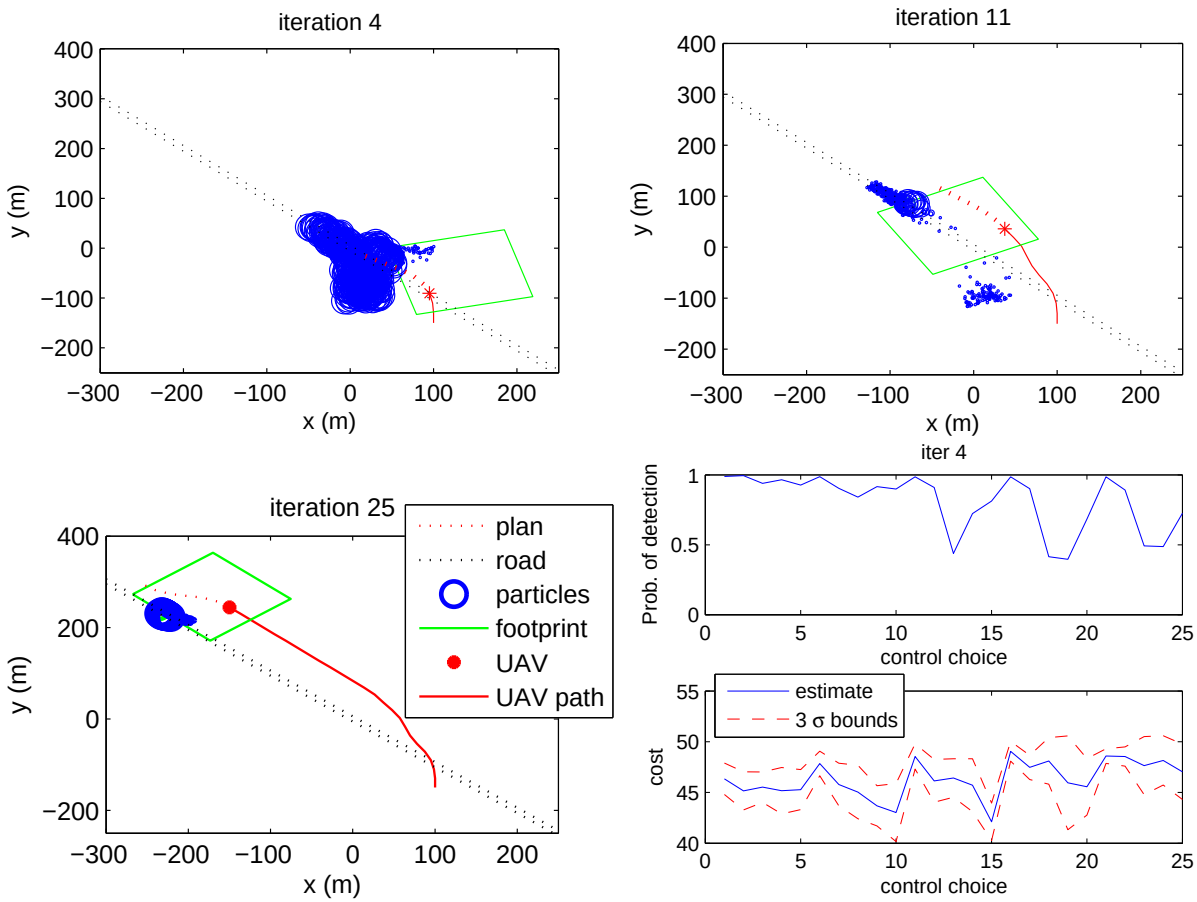

Figure 8. (Simulation 4) UAV position, sensor footprint, and particle filter estimate throughout simulation under receding horizon control. Lower right plot shows predicted cost versus control choice at iteration 4. For each of 25 candidate controls, both information-theoretic cost and probability of detection are compared.

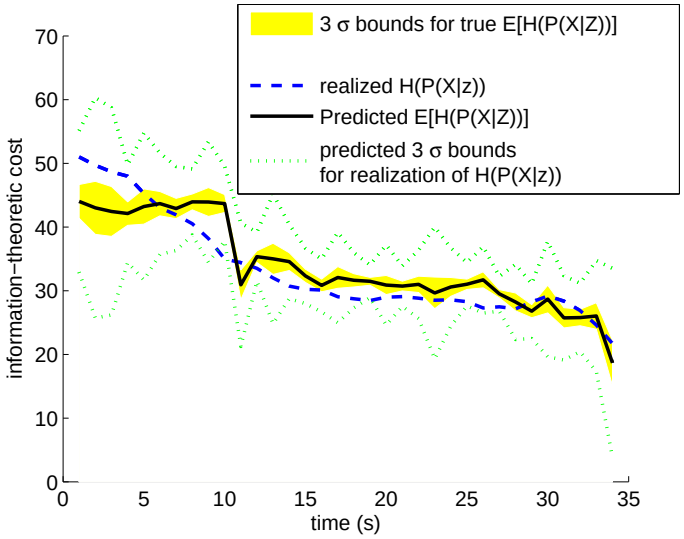

Figure 9. (Simulation 4) Predicted cost compared to conditional entropy, with estimated 3 σ bounds, under receding horizon control

Flight experiment: particle filter estimate from UAV-mounted camera

In initial experimental work, a particle filter is implemented to estimate the position of a moving vehicle using a camera mounted on an unmanned aircraft. A simple classifier is used to detect the truck in each frame, and the sensor likelihood function is based on a pinhole camera model subject to aircraft state uncertainty.[21] The target vehicle motion is terrain dependent and modeled as in simulation 2. The prior distribution for the vehicle position is uniform over a 300 by 200 meter rectangle.

The first plot of figure 10 shows a video frame with the target present. In this initial experiment, video was recorded onboard the UAV and post-processed. The upper right plot shows the particle distribution before the target has been detected, where the center of the initial grid has been searched, and some particles have traveled along the road. The searched area is centered to the right of the UAV's path due to the bank angle effect on the fixed camera. The lower left plot shows the particle distribution after that target has been observed in a single frame. The few particles near the observed target location have high weight, and all others have low weight, although they have not yet been eliminated. In the lower right, the target has been observed in multiple frames and all particles are clustered in an area approximately 30 meters by 5 meters. The random target motion and the poor UAV state estimate prevent the target position from being more tightly localized. Improved UAV state estimation will be critical for future experimental work.

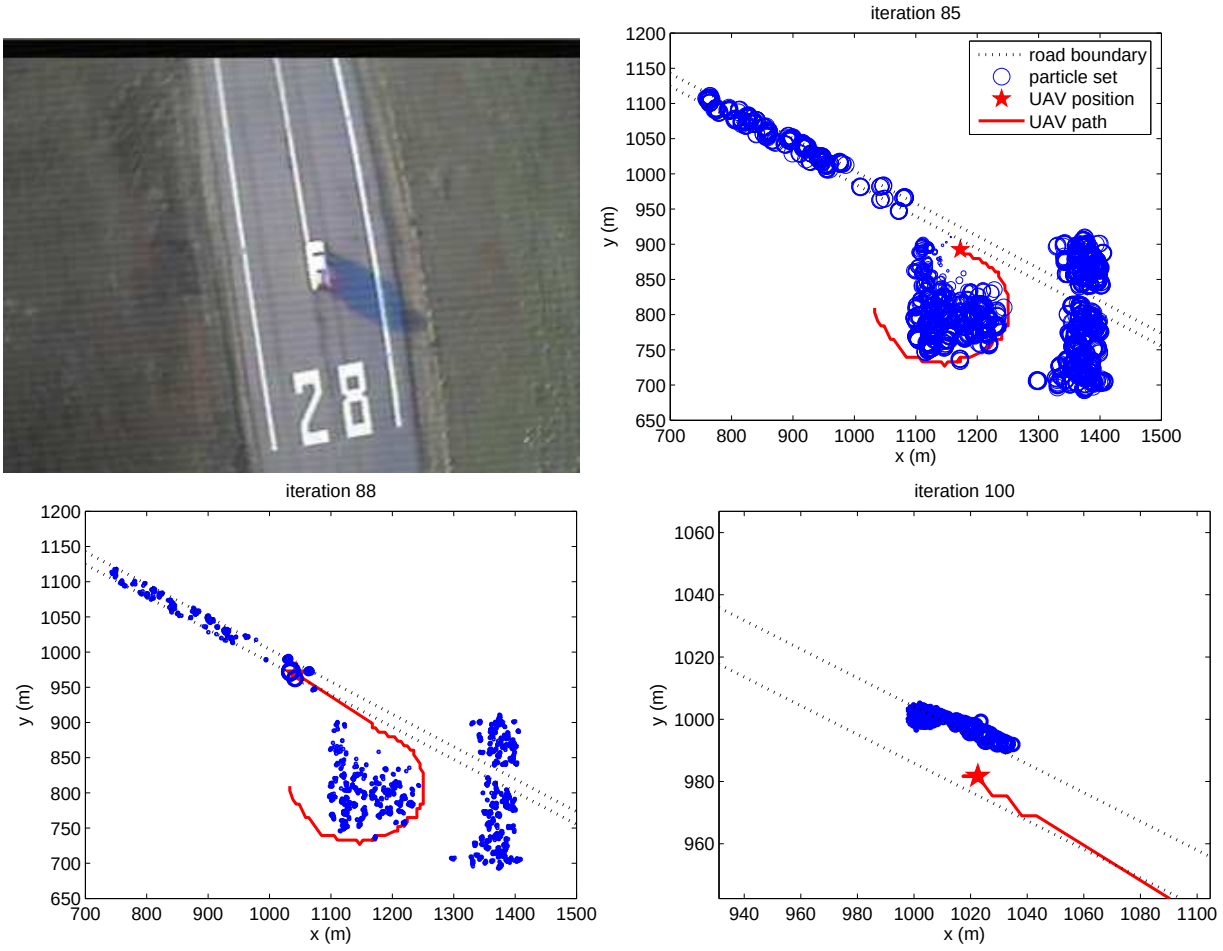

Figure 10. (Flight experiment) Raw video of target vehicle on runway, collected from UAV (upper left). Particle filtering density before target detection (upper right), immediately after truck is detected (lower left) and after many detections (lower right)

V. Conclusions and Future Work

Tracking control for a mobile sensor platform has been formulated as a receding horizon minimization of the entropy of the target state estimate. A particle filter is able to incorporate non-linear motion and sensor models, but requires a novel method to approximate the entropy of the estimate. A novel sampling method is also developed to predict the expected estimate entropy over a planning horizon greater than one. This is the only such control formulation known to the author which allows a moving target, non-Gaussian estimate, and multiple step receding horizon.

Simulation results verify the accuracy of the entropy prediction and show performance gain due to a longer planning horizon. A non-linear sensor and terrain-dependent target motion model are simulated, and also implemented experimentally in a particle filtering application. Further simulation is planned to study the tradeoffs between horizon length and control performance. The formulation and approximation techniques presented here are also being applied to cooperative information-theoretic control,[22] with further development of the multiple-sensor problem as future work.

Acknowledgments

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

References

[1]Spry, S. C., Girard, A. R., and Hedrick, J. K., "Convoy Protection using Multiple Unmanned Air Vehicles: Organization and Coordination," *Proceedings of the American Control Conference*, 2005.

[2]Campbell, M. E. and Wheeler, M., "A Vision Based Geolocation Tracking System for UAVs," *Proceedings of the AIAA Guidance Navigation and Control Conference*, August 2006.

[3]Koller, J. and Ulmke, M., "Data Fusion for Ground Moving Target Tracking," *Aerospace Science and Technology*, Vol. 11, No. 4, May 2007.

[4]Webb, S. and Furukawa, T., "Belief Driven Manipulator Control for Integrated Searching and Tracking," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[5]Lavis, B., Furukawa, T., and Durrant-Whyte, H. F., "Dynamic Space Reconfiguration for Bayesian Search and Tracking with Moving Targets," *Autonomous Robotics, to appear*, 2008.

[6]Rathinam, S., Kim, Z., Soghikian, A., and Sengupta, R., "Vision Based Following of Locally Linear Structures Using an Unmanned Aerial Vehicle," *Proceedings of the IEEE Conference on Decision and Control*, December 2005.

[7]Andrieu, C., Doucet, A., Singh, S., and Tadic, V., "Particle Methods for Change Detection, System Identification, and Control," *Proceedings of the IEEE*, Vol. 92, No. 3, Mar 2004, pp. 423–438.

[8]Shannon, C. and Weaver, W., "The Mathematical Theory of Communication, Bell. Labs. Series," 1950.

[9]Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Athena, 2007.

[10]Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J., "Distributed Cooeprative Search using Information-Theoretic Costs for Particle Filtes, with Quadrotor Applications," *Proceedings of the AIAA Guidance Navigation and Control Conference*, August 2006.

[11]Cole, D. T., Goktogan, A. H., and Sukkarieh, S., "The Demonstration of a Cooperative Control Architecture for UAV Teams," *Proceedings of the 10th International Symposium on Experimental Robotics*, Rio de Janiero, July 2006.

[12]Grocholsky, B., "Information Driven Coordinated Air-Ground Proactive Sensing," *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005.

[13]Bertsekas, D. P., Nedic, A., and Ozdaglar, A. E., *Convex Analysis and Optimization*, Athena Scientific, 2003.

[14]Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, February 2002.

[15]Stachnissy, C., Grisettiyz, G., and Burgard, W., "Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops," *Proceedings of the Internation Conference on Robotics and Automation*, 2005.

[16]Karlsson, R., Schön, T., Törnqvist, D., Conte, G., and Gustafsson, F., "Utilizing Model Structure for Efficient Simultaneous Localization and Mapping for a UAV Application," March 2008.

[17]Jordan, M. I., *An Introduction to Probabilistic Graphical Models, in preparation*.

[18]Huster, A., *Relative Position Sensing by Fusing Monocular Vision and Interial Rate Sensors*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, 2003.

[19]Tisdale, J., Ryan, A., Kim, Z., Tornqvist, D., and Hedrick, J. K., "A Multiple UAV System for Vision-based Search and Localization," *Proceedings of the American Controls Conference*, June 2008.

[20]Hoffmann, G. M. and Tomlin, C. J., "Mobile Sensor Network Control using Mutual Information Methods and Particle Filters," *Submitted to IEEE Transactions on Automatic Control*, 2007.

[21]Kim, Z. and Sengupta, R., "Target Detection and Position Likelihood Using an Aerial Image Sensor," *Proceedings of the International Conference on Robotics and Automation*, Pasadena, May 2008.

[22]Ryan, A., Durrant-Whyte, H., and Hedrick, J. K., "Information-theoretic Sensor Motion Control for Distributed Estimation," *Proceedings of the International Mechanical Engineering Congress and Exposition*, Seattle, November 2007.