On Information Measures based on Particle Mixture for Optimal Bearings-only Tracking

Per Skoglar, Umut Orguner, Fredrik Gustafsson
Automatic Control
Department of Electrical Engineering
Linköping University, Sweden
{skoglar,umut,fredrik}@isy.liu.se

Abstract—In this work we consider a target tracking scenario where a moving observer with a bearings-only sensor is tracking a target. The tracking performance is highly dependent on the trajectory of the sensor platform, and the problem is to determine how it should maneuver for optimal tracking performance. The problem is considered as a stochastic optimal control problem and two sub-optimal control strategies are presented based on the Information filter and the determinant of the information matrix as the optimization objective. Using the determinant of the information matrix as an objective function in the planning problem is equivalent to using differential entropy of the posterior target density when it is Gaussian. For the non-Gaussian case, an approximation of the differential entropy of a density represented by a particle mixture is proposed. Furthermore, a gradient approximation of the differential entropy is derived and used in a stochastic gradient search algorithm applied to the planning problem.

TABLE OF CONTENTS

1. INTRODUCTION

Optimal trajectory for bearings-only tracking is a classical nonlinear estimation problem. The problem is to estimate the state of a target given a number of noisy measurements. The sensor platform is free to maneuver, and the problem is to find the optimal trajectory that maximizes the tracking and estimation performance.

The optimal observer trajectory is in [16] computed by maximizing mutual information. Dynamic programming is used to minimize the determinant of the error covariance of a target with linear dynamics over the entire measurement sequence from a bearing-only sensor. Furthermore an enumeration brute force method with optimal pruning is developed for minimizing the trace of final target error covariance. The Fisher information matrix (FIM) is in [23] used as the objective function. The optimization is done in a dynamic programming framework, where target and observer are modeled as Markov chains. The resulting problem is a Partially Observable Markov Decision Problem (POMDP) for determining the optimal control law of the observer. The computational complexity is very large and only very small problems can be addressed. In [10] and [11] different information-theoretic distributed control architectures for searching and localizing targets are proposed. [10] is using an Information filter framework similar to our approach, and [11] is using the mutual information computed from a particle set representing the target density.

As in our work, [20] uses a Stochastic Approximation (SA) approach to solve an observer trajectory planning problem where the gradient is estimated from a particle mixture. In [18], SA algorithms and particle filters are used in a similar manner to our work for maximum likelihood parameter estimation.

We note that most approaches in this research area are based on some kind of measure from information theory. FIM, CRLB, mutual information, and entropy are all strongly related. Furthermore, similar problem definitions can be found under a broad range of different terms, among others, "sensor scheduling", "target motion analysis", and, "observer trajectory planning".

Outline

In Section 2 we define the problem of planning for optimal estimation (tracking) performance used in this paper. System and observation models are defined and a general optimization problem is formed that must be solved for optimal estimation performance. In Section 3 the general estimation equations are introduced. These equations are fundamental in all target tracking and estimation applications. The general

---

978-1-4244-2622-5/09/$25.00 ©2009 IEEE
IEEEAC Paper #1135, Version 8 Updated 09/01/2009.

planning problem is then discussed from a stochastic optimal control point of view, and two well known sub-optimal approaches are introduced.

The estimation performance measure is the main subject of this paper, in particular we investigate two information measures. In Section 4 a linearized version of the problem is considered and the resulting EKF filter is combined with an information matrix criterion. In Section 6 the more general Particle filter is used as the estimator and this requires another choice of information measure. This measure is an approximation of the differential entropy derived in Section 5 and a stochastic gradient algorithm is proposed. Finally, conclusions are drawn in Section 7.

2. Optimal Trajectory for Bearings-only Localization

In this section we define the problem of planning for optimal estimation performance used in this paper. The task is to localize a nearly stationary feature with a bearings-only sensor. The sensor is attached to a platform moving in the *xy*-plane with constant speed. The proposed planning problem is a simplified target tracking scenario, but it still contains important and interesting aspects of a general target tracking problem.

The target is assumed to be a slowly moving target and modeled as a random walk

$$x_{k+1} = f(x_k, w_k) = x_k + w_k \tag{1}$$

where the state $x_k = (\eta_k,\ \xi_k)^\mathsf{T}$ is the position of the target and the process noise is $w_k \sim \mathcal{N}(0, Q)$.

The state elements of the sensor platform state vector x^s are the position and the heading, $x_k^s = (\eta_k^s,\ \xi_k^s, \psi_k^s)^\mathsf{T}$. The dynamic model is a basic constant speed model with rate of change of heading $u_k = \omega_k$ as the control signal. Thus, the dynamic model is given as

$$x_{k+1}^s = f^s(x_k^s, u_k) = x_k^s + \begin{pmatrix} v\cos(\psi_k^s) \\ v\sin(\psi_k^s) \\ \omega_k T \end{pmatrix} \tag{2}$$

where T is the sampling time and v is the speed. Note that the sensor platform model is deterministic and that we always have perfect state information about x_k^s. This means that we assume that we have neither disturbances nor navigation error.

The observation model is the relative angle between the sensor platform and the target, i.e.,

$$\begin{aligned} y_k &= h(x_k, x_k^s, e_k) \\ &= \arctan_2(\xi_k - \xi_k^s, \eta_k - \eta_k^s) + e_k \end{aligned} \tag{3}$$

where e_k is the measurement noise modeled as $e_k \sim \mathcal{N}(0, R)$.

In the planning problem we search for a control input sequence

$$\pi^{M-1} \triangleq \{u_k\}_{k=1}^{M-1} \tag{4}$$

that minimizes some expected loss. Constraints on the control signal are defined by the set $\mathcal{U}$, in this work we assume that

$$\mathcal{U} = \{u \mid -u_{max} \le u \le u_{max}\}. \tag{5}$$

The loss function $L(x_M)$ is a function of a random variable[1] $x_M \sim p(x_M|I^M, I^0)$ where I^0 represents all information, e.g. measurements, received up to time 0, and I^M is a random variable representing all future information that will be received up to time M. The target density $p(x_k|I^k)$ is computed by an *estimator*, the target tracker, see the estimation theory section below. Thus, the loss function maps a target state to a scalar metric usable in an optimization framework.

To keep notation as simple as possible, we always assume that the planning is performed at time $k = 0$. Thus, in case of replanning the time index is reset and the time for the replanning is 0.

The general planning problem can now be defined as

$$\begin{aligned} \min_{\pi^{M-1}} \quad & E[L(x_M)|I^0] \\ \text{s.t.} \quad & u_k \in \mathcal{U} \\ & x_{k+1} = f(x_k, w_k) \\ & x_0 \sim p(x_0|I^0) \\ & w_k \sim \mathcal{N}(0, Q) \\ & x_{k+1}^s = f^s(x_k^s, u_k) \\ & x_0^s = \bar{x}_0^s \\ & y_k = h(x_k, x_k^s, e_k) \\ & e_k \sim \mathcal{N}(0, R). \end{aligned} \tag{6}$$

3. Estimation Theory and Stochastic Optimal Control

In this section we present some background theory fundamental for our work. First we give some results in estimation theory that are fundamental in target tracking algorithms. Since we consider the planning problem as a stochastic control problem we also give a brief introduction to stochastic control theory. In particular, the terminal stochastic control problem is considered and two sub-optimal approaches, Certainty Equivalence Control (CEC) and open-loop feedback control (OLFC), are introduced. In Section 4 these approaches will be applied to our planning problem.

The discussion in this background section is on a rather general level. Thus, the variables and functions are also general despite similar names as in other parts of this paper.

[1]The lower-case x is here a random variable. The notation may be unclear, the actual meaning of a variable, random or non-random, is given by the context.

General Estimation Theory

Consider a rather general dynamic model defined as

$$x_{k+1} \sim p(x_{k+1}|x_k) \tag{7}$$

where x_k is the state. Furthermore, let the observation model be defined as

$$y_k \sim p(y_k|x_k) \tag{8}$$

and let $Y^k = \{y_1, y_2, ..., y_k\}$ be the set of all observations up to time k. The general state estimator is derived from Bayes rule

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)} \tag{9}$$

and can be expressed as the recursive update formula

$$p(x_k|Y^k) = p(x_k|y_k, Y^{k-1}) = \alpha_k^{-1} p(y_k|x_k)p(x_k|Y^{k-1}) \tag{10}$$

and the one step ahead prediction

$$p(x_k|Y^{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Y^{k-1})dx_{k-1}. \tag{11}$$

The normalizing factor α_k is

$$\alpha_k = p(y_k|Y^{k-1}) = \int p(y_k|x_k)p(x_k|Y^{k-1})dx_k. \tag{12}$$

However, there are only a few cases when it is possible to derive analytic solutions of these equations. One case is the linear Gaussian case, leading to the well known *Kalman filter* and the *Information filter* [13]. In the general case, numeric approximations are necessary and one popular technique is to approximate the density $p(x_k|Y^s)$ by a particle mixture, containing N particles $\{x_{k|s}^{(i)}\}_{i=1}^N$ with associated weights $\{w_{k|s}^{(i)}\}_{i=1}^N$, as

$$p(x_k|Y^s) \approx \sum_{i=1}^{N} w_{k|s}^{i} \delta(x_k - x_{k|s}^{(i)}) \tag{13}$$

where $\delta(.)$ is the Dirac delta function. This approximation leads to the *Particle filter* (PF) [9] [6], and it can be shown that the larger the number of particles is, the better the approximation will be.

Finite Horizon Stochastic Optimal Control

The information gathering problem of a platform with a bearings-only sensor can be viewed as a *terminal stochastic control problem*. See [3] for a more detailed presentation of stochastic optimal control.

Consider a system where the state evolves as the discrete-time stochastic system

$$x_{k+1} = f(x_k, u_k, w_k) \tag{14}$$

where $k = 0, 1, ..., M-1$ is the time, and w_k represents the random disturbances and u_k is a control signal. From the system, only imperfect information of the state is available through the observations

$$y_k = h(x_k, e_k) \tag{15}$$

where e_k represents the random errors in the observations. Now the objective function is naturally described as a function of the final state. In some cases additional loss on the way is necessary to consider, but we ignore that in this work.

The information available for the controller at time k is

$$I^{k-1} = \{x_0, y_1, u_1, ..., y_{k-1}, u_{k-1}\}, \tag{16}$$

i.e., the initial state and the history of all previous control and measurements. An admissible control law can then be defined as a function of available information, i.e.,

$$\pi^{M-1} = \{u_1(I^0), u_2(I^1), ..., u_{M-1}(I^{M-2})\}. \tag{17}$$

The planning problem is now represented by (14), (15) and the expected loss, i.e.,

$$\begin{array}{ll} \min\limits_{\pi^{M-1}} & J(x_0, \pi^{M-1}) = E\left[L_M(x_M)|I^0\right] \\ \text{s.t.} & u_k \in \mathcal{U} \\ & x_{k+1} = f\left(x_k, u_k(I^{k-1}), w_k\right) \\ & y_k = h(x_k, e_k) \end{array} \tag{18}$$

This is a terminal information form version of the finite horizon stochastic optimal control problem [3].

At the core of optimal stochastic control is the "principle of optimality". As stated by Bellman: "Whatever any initial states and decision [or control law] are, all remaining decision must constitute an optimal policy with regard to the state which results from the first decision" [1]. The *dynamic programming* algorithm is based on the principle of optimality. First the optimal problem for the last stage is solved and then the extended problem with the last two stages is solved, and so on until the entire problem is solved. However, in general it is often impossible to find closed form solutions even to the small sub-problem at each stage. A standard solution is to search for approximate numerical solutions by discretizing the problem, but as Bellman observed this method is susceptible to the "curse of dimensionality" where larger problems are prohibitive both computationally and in required memory storage.

An optimal feedback control law will not only steer the system in accordance with the reference signal. In addition, the control law will show *probing* and *caution* behavior. Probing represents actions to enhance estimation precision in order to improve overall performance in the future. Caution is acting so as to minimize the consequences of erroneous assumptions about the state of the environment. Both these components are often in conflict with the error reducing part of the control law and control laws including this compromise are denoted *dual control*. The dual control problem was first discussed by Feldbaum [8].

Partially Observable Markov Decision Problems (POMDP) have received much attention during the recent years. Typically the state space, action space, observation space, and planning horizon all are finite and the solution becomes a piecewise-linear and convex function over the belief space. The first algorithm for an exact solution to POMDP was given by Sondik in [21]. More efficient algorithms have been developed [12] and during the last years, many approximative methods have been proposed to handle the complexity of POMDPs, but still only rather small problems can be handled.

Certainty Equivalence Control

For linear quadratic Gaussian (LQG) problems it is possible to find a closed form solution to the general Dynamic Programming problem. The solution of the LQG problem can be separated into two stages, first an estimation part, and second solving a non-stochastic optimization problem. This separation is very convenient and is called the *certainty equivalence principle*. However, for general problems this principle does not hold.

A popular suboptimal control scheme is to use Assumed Certainty Equivalence (ACE), i.e., to assume that the certainty equivalence principle is holding and consider the estimation and the control independently. The Certainty Equivalent Control (CEC) can be summarized as follows: Given an information vector I^0, an estimator produces a typical value of the state and the disturbance

$$\hat{x}_0 = E[x_0|I^0], \quad \hat{w}_k = E[w_k|\hat{x}_k, u_k], \tag{19}$$

respectively. The problem to solve is then a perfect information problem, i.e., the deterministic version of (18),

$$\begin{array}{ll} \min\limits_{\pi^{M-1}} & L_M(x_M) \\ \text{s.t.} & u_k \in \mathcal{U} \\ & x_0 = \hat{x}_0(I^0) \\ & x_{k+1} = f(x_k, u_k, \hat{w}_k(x_k, u_k)) \end{array} \tag{20}$$

and use the first element in the control sequence as control input, and then repeat. Time 0 is always the time when the planning is performed.

A problem with ACE is that dual control properties such as probing and caution are missing. Thus, ACE is not well suited for sensor planning problems since Certainty Equivalent Control (CEC) will not take the possible future information profit into account.

Open-loop Feedback Control

Another approximation is *open-loop feedback control* (OLFC). Unlike the CEC which computes the estimate $\hat{x}$, OLFC is instead computing the probability distribution $p(x_k|I_k)$ and thus taking the uncertainty about x_k and the disturbances into account. However, OLFC is very "pessimistic" since it selects control input as if no further information will be received. Hence, the name is OLFC since the method is performing feedback from the current measurement, but is assuming open loop control over the remaining steps.

The OLFC method contains the following steps. First compute the conditional probability distribution $p(x_0|I_0)$. Then find a control sequence that solves the problem

$$\begin{array}{ll} \min\limits_{\pi^{M-1}} & E\left[L_M(x_M)\right] \\ \text{s.t.} & u_k \in \mathcal{U} \\ & x_0 \sim p(x_0|I_0) \\ & x_{k+1} = f(x_k, u_k, w_k(x_k, u_k)) \end{array} \tag{21}$$

Use the first element in the control signal as the control input, and then repeat. As before, time 0 is always the time when the planning is performed.

4. An Information Filter Approach

In this section we use the information form of the well known Extended Kalman filter to implement a sub-optimal CEC planner. An advantage of this approach is that the resulting optimization problem is deterministic.

The Information Filter

The Kalman filter is the optimal filter, in the minimum square error sense, for linear systems (14) and (15) with Gaussian noises w and e. The Kalman filter maintains a state vector $\hat{x}_k$ and its covariance matrix P_k. The *Information filter* [13] is equivalent to the Kalman filter, but instead of maintaining a state vector and a covariance matrix, the information filter maintains the information state $\hat{i}_k = P_k^{-1}\hat{x}_k$ and the *information matrix* $\mathcal{Y}_k = P_k^{-1}$.

A popular approach to handle nonlinear models is a linearized version of the Kalman filter called *(Schmidt) Extended Kalman filter* (EKF). The EKF is based on a Taylor series expansion of (14) and (15) as

$$F_k = \left.\frac{\partial f(x,0)}{\partial x}\right|_{x=\hat{x}_{k|k}}, \tag{22}$$

$$G_k = \left.\frac{\partial f(\hat{x}_{k|k},w)}{\partial w}\right|_{w=0}, \tag{23}$$

$$H_k = \left.\frac{\partial h(x,0)}{\partial x}\right|_{x=\hat{x}_{k|k-1}}. \tag{24}$$

The EKF can also be given in an information form called *Extended Information Filter* (EIF). The update and prediction equations of the information matrix in an (Extended) Information filter are

$$\mathcal{Y}_{k|k} = \mathcal{Y}_{k|k-1} + H_k^T R_k^{-1} H_k \tag{25}$$

$$\mathcal{Y}_{k+1|k} = (F_k \mathcal{Y}_{k|k}^{-1} F_k^T + G_k Q_k G_k^T)^{-1} \tag{26}$$

where R_k and Q_k are the covariances of the measurement noise and process noise, respectively. Note that the update step is additive and this is one major reason for the popularity of the information form, especially if information from several sensors must be fused in the filter [17]. We also note that in the linear Gaussian case the information matrix is equivalent to the *Fisher information matrix* that is used for bounding estimation error by the Cramer Rao Lower Bound (CRLB).

CEC Planner based on Information Filter

The Information filter is now used to define a CEC version of the planning problem (6). The "quality" of the state estimate is captured by the information matrix $\mathcal{Y}(t) = P_k^{-1}$, where P_k is the covariance of the position of the target, i.e.,

$$P_k = E[(x_k - Ex_k)(x_k - Ex_k)^{\mathsf{T}}]. \tag{27}$$

In this section we will use a loss function defined as the determinant of the information matrix. The reason for this is, as we will see later in Section 5, that in the Gaussian case, the negative differential entropy is a monotonic function of the determinant of the information matrix. The loss function is then

$$L(X_M) = -\det \mathcal{Y}_M \tag{28}$$

where the state vector X_k is the augmented vector

$$X = \left(x^{\mathsf{T}},\, (x^s)^{\mathsf{T}},\, \mathcal{Y}_{11},\, \mathcal{Y}_{12},\, \mathcal{Y}_{22}\right)^{\mathsf{T}} \tag{29}$$

where $\mathcal{Y}_{ij}$ denotes the element of $\mathcal{Y}$ lying on the intersection of the ith row and the jth column. Note that $\mathcal{Y}_{21}$ is omitted since $\mathcal{Y}$ is symmetric.

The information matrix is updated according to the EIF filter equations in (25). The Jacobian of the observation model (3) is

$$\begin{aligned} H_k &= \nabla_{x_k} h(x_k, x_k^s, 0)\Big|_{x_k = \hat{x}_k} \\ &= \frac{1}{\hat{r}_k^2}\left[-(\hat{\xi}_k - \xi_k^s),\ \hat{\eta}_k - \eta_k^s\right] \end{aligned} \tag{30}$$

where $\hat{r}_k = \sqrt{(\hat{\eta}_k - \eta_k^s)^2 + (\hat{\xi}_k - \xi_k^s)^2}$ is the distance between the sensor and the target.

Thus, the planning problem is

$$\begin{aligned} \min_{\pi^{M-1}} \quad & L(X_M) = -\det \mathcal{Y}_M \\ \text{s.t.} \quad & u_k \in \mathcal{U} \\ & x_0 = \hat{x}_0 \\ & x_{k+1} = f(x_k, 0) \\ & x_{k+1}^s = f^s(x_k^s, u_k) \\ & y_k = h(x_k, x_k^s, 0) \\ & \mathcal{Y}_{k+1} = g(\mathcal{Y}_k) \end{aligned} \tag{31}$$

where g is the EIF equations in (25) and (26). To solve this problem we will use a gradient search algorithm that is introduced next. Simulation results are then given after that.

Gradient Search Algorithm

Consider the problem of minimizing a loss function $L(\theta)$. Most numerical minimization methods are using an iterative procedure

$$\hat{\theta}^{(k+1)} = \hat{\theta}^{(k)} + a_k g(\hat{\theta}^{(k)}) \tag{32}$$

where $a_k > 0$ is the step size and g is the search direction. One example is the *steepest descent algorithm* where the search direction is determined by the negative gradient

$$g(\theta) = -\frac{\partial L(\theta)}{\partial \theta}. \tag{33}$$

There are other methods, e.g. the Newton-method, with faster convergence rate, but they require computation of the Hessian of the loss function and those methods are therefore not considered in this work. Depending on which information of the gradient that is available, the steepest descent methods can be divided into two groups. Either the loss function is known and differentiable and the gradient can be derived analytically or the gradient is not directly available and one has to compute an approximation of the gradient from measurements of the loss function.

Simulation Result of Information Filter Planner with known Target Position

Let us first make the unrealistic assumption that the error of the initial target position is zero and that the measurements are perfect, i.e., the target position estimate is the true position. The problem is not stochastic and a deterministic gradient search algorithm will perform well. However, the problem is both non-linear and non-convex so the starting point is important and there is no guarantee that the solution is the global optimum.

In Figure 1 three different simulations with different planning horizon are shown, see Table 1 for simulation parameters. The planning horizon lengths are 1, 4, and 8 sampling periods respectively. Note that replanning is done after the whole previous plan has been executed. In practice, replanning should be done as new information is received. The shortest planning horizon gives a "greedy" and shortsighted behavior of the sensor platform. The path is shaped as a spiral because of the trade-off between maximizing the base-line relative the target and getting closer to the target. However, if the planning horizon is increased, the vehicle initially travels more directly towards the object.

The resulting path of the greedy planner in Figure 1 can be explained by the contour plot in Figure 2. The contour plot shows the "information surface" after the measurement update at the sensor position marked with a star. In other words, given the target estimation covariance shown as a dashed error ellipse, the plot shows the determinant of the information matrix after one new measurement update in a new position. We can see that taking a measurement from a position more perpendicular, with respect to the major ellipse axis, is much better than from a position along the extension of the error

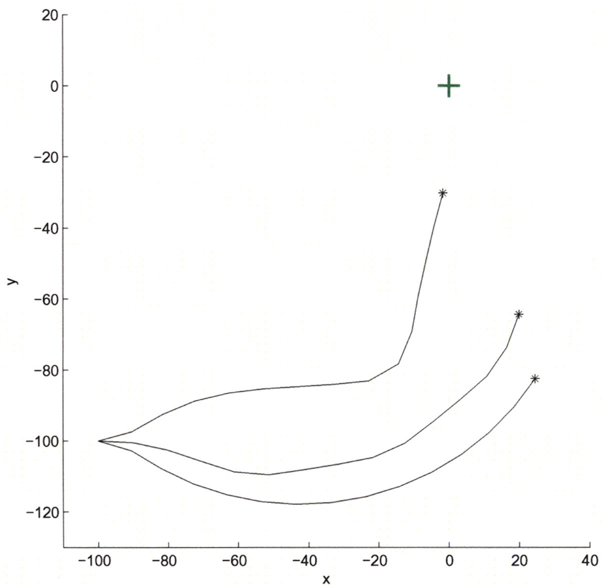

Figure 1. Information filter planning with perfect measurements, i.e., zero estimation error. The trajectory of the sensor platform for three different planning horizon lengths.

Table 1. Parameters of Simulation 1 and 2.

Target position	$x = (0,\ 0)^{\mathsf{T}}$ [m]
Initial target covariance	$P = \mathrm{diag}[50^2,\ 50^2]$
Initial sensor position	$x^s = (-100, -100)^{\mathsf{T}}$ [m]
Sensor platform speed	$v = 10$ [m/s]
Measurement variance	$R = (1\pi/180)^2$ [rad^2]
Process variance	$Q = 0$

ellipse. The greedy planner goes in the direction where the slope of the information surface is the largest.

Simulation Results of CEC Information Filter Planner

In Figure 3, three other simulations, where the position of target is concurrently estimated from the measurements, are shown. The same parameters (Table 1) as in Figure 1 are used. If the estimated position in each planning step is considered as the true position in the planner, then the planning problem is still deterministic. However, even if the global optimum is found, this may not be the best overall solution due to the estimation error.

OLFC Information Filter Planner

The form of the planning problem above is well suited for using with a target tracker maintaining the covariance or information matrix of the target's position, e.g. Extended Kalman Filter. If the target tracker instead is a Particle filter (PF), then the information matrix first has to be computed from the current PF state with obvious degradation of the information in

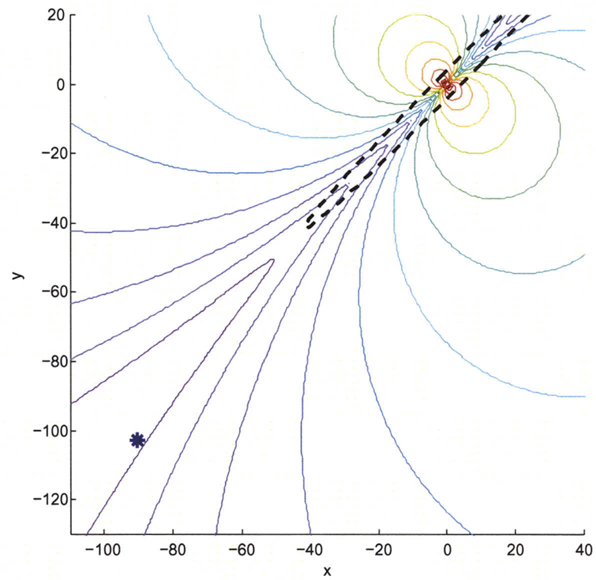

Figure 2. The "information surface" after the first measurement update. Target covariance shown as a dashed ellipse and current sensor position as a star.

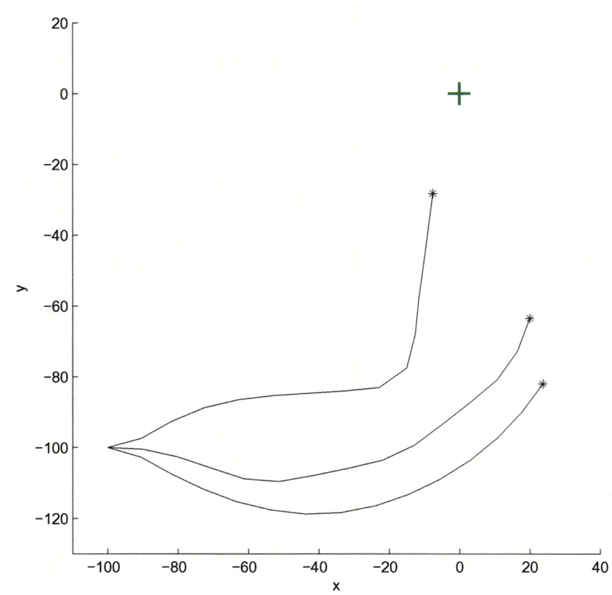

Figure 3. CEC Information Filter planner. The trajectory of the sensor platform for three different planning horizon lengths.

the estimator.

We here propose a heuristic approach that can be used with a PF target tracker, but also with an EKF if samples are drawn based on the target covariance. First, select N particles, which represent different hypotheses about the state of the target, and associate a covariance matrix $P^{(i)}$ to each of them. For all of the particles, do the calculations as in the single estimate case and obtain resulting information matrices $\mathcal{Y}_M^{(i)}$. Let all particle states be augmented into a new large vector $\chi = ((x^{(1)})^\mathsf{T},\ (x^{(2)})^\mathsf{T}, ..., (x^{(N)})^\mathsf{T})^\mathsf{T}$. If all particles are independent, the covariance matrix is a block diagonal matrix

$$\Pi = E[(\chi - E\chi)(\chi - E\chi)^\mathsf{T}] = \mathrm{diag}[\mathrm{P}^{(1)},\ \mathrm{P}^{(2)}, ..., \mathrm{P}^{(\mathrm{N})}] \tag{34}$$

and the information matrix is also block diagonal

$$\Omega = \Pi^{-1} = \mathrm{diag}[\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}, ..., \mathcal{Y}^{(\mathrm{N})}] \tag{35}$$

where $\mathcal{Y}^{(i)} = (P^{(i)})^{-1}$. The loss function is then given as

$$L(X_M) = -\det \Omega_M = -\prod_{i=1}^{N} \det \mathcal{Y}_M^{(i)}. \tag{36}$$

Taking the logarithm of the product, we get

$$\ln L(X_M) = -\sum_{i=1}^{N} \ln \det \mathcal{Y}_M^{(i)} \tag{37}$$

that can be used in an equivalent optimization problem, but with better numerical properties.

One question is how the information matrices $\mathcal{Y}_0^{(i)}$ should be defined. An ad-hoc proposal is to use the information matrix $\mathcal{Y}_0$ from the estimator weighted by $1/N$, since the sum of N normal distributed random variables with mean μ_i and covariance P/N is distributed as $\mathcal{N}(\sum_{i=1}^N \mu_i, \sum_{i=1}^N P/N) = \mathcal{N}(\mu, P)$ where $\mu = \sum_{i=1}^N \mu_i$. However, by inspecting (37) we realize that this weight will not affect the result of the optimization problem.

Thus, we have the following optimization problem

$$\begin{aligned}
\min_{\pi^{M-1}} \quad & -\sum_{i=1}^{N} \ln \det \mathcal{Y}_M^{(i)} \\
\text{s.t.} \quad & u_k \in \mathcal{U} \\
& x_{k+1}^{(i)} = f(x_k^{(i)}, w_k^{(i)}) \\
& x_0^{(i)} \sim p(x_0 | I^0) \\
& w_k^{(i)} \sim \mathcal{N}(0, Q) \\
& x_{k+1}^s = f^s(x_k^s, u_k) \\
& x_0^s = \bar{x}_0^s \\
& y_k^{(i)} = h(x_k^{(i)}, x_k^s, e_k^{(i)}) \\
& e_k^{(i)} \sim \mathcal{N}(0, R) \\
& \mathcal{Y}_{k+1} = g(\mathcal{Y}_k) \\
& \mathcal{Y}_0 = P_0^{-1}/N
\end{aligned} \tag{38}$$

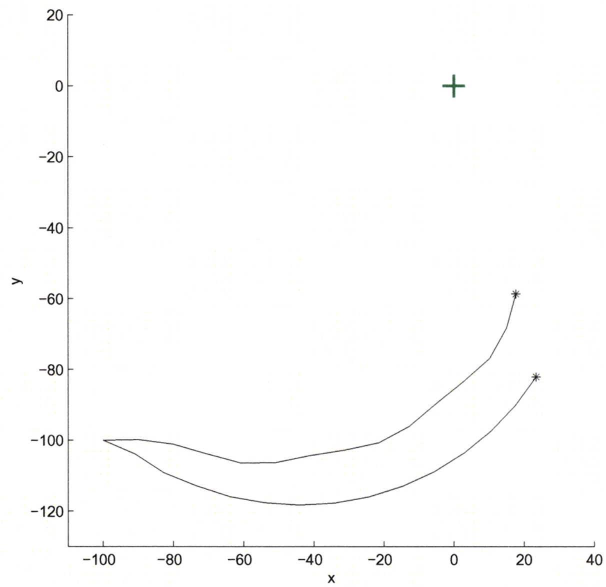

Figure 4. OLFC Information Filter Planner. Trajectory of the sensor platform.

where g is the EIF equations in (25) and (26). Given the samples $x_0^{(i)}$, $w_k^{(i)}$ and $e_k^{(i)}$, this problem is, like the CEC problem, deterministic.

Simulation Results of OLFC Planner

The simulation result is shown in Figure 4 for two cases, planning horizon length 1 and 4. The number of particles is 100. The result is similar to the previous simulations for this basic example, but it is reasonable to believe that this approach will perform better in a more complex scenario. The longer horizon case differs slightly from run to run, depending on the actual realization of the particle set. This is also why the length 8 case is not shown in the figure. This approach also has a "singularity" problem with the determinant criterion when the plan of the sensor platform path is very close to a particle location. This can be overcome in the 3D case when particles and the sensor platform are separated on different altitudes or with a suitable adjustment of the criterion.

5. DIFFERENTIAL ENTROPY OF DENSITY REPRESENTED BY A PARTICLE MIXTURE

The CEC Information filter planning problem (31) assumes that the target probability density can be expressed as a Gaussian probability density. We need a more general approach to be able to handle general target probability densities. In this section we introduce the particle mixture density as a flexible tool of representing general densities. We also present some basic information theory and propose differential entropy as a measure of the "quality" of the target state. Unfortunately, it is not straightforward to compute the differential entropy of a density represented by a particle mixture. We propose a method for the computation of the differential entropy that is then, in the next section, applied to the planning problem.

The Particle Filter Equations

In a Particle filter (PF) the target density $p(x_k|Y^k)$ is approximated by a particle mixture containing N particles $\{x_k^{(i)}\}_{i=1}^N$ and associated weights $\{w_k^{(i)}\}_{i=1}^N$. Thus, at time $k-1$ the target density is

$$p(x_{k-1}|Y^{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \delta(x_{k-1} - x_{k-1}^{(i)}). \quad (39)$$

Substituting this particle representation of $p(x_{k-1}|Y^{k-1})$ into (11), the predicted density $p(x_k|Y^{k-1})$ is obtained as

$$p(x_k|Y^{k-1}) = \sum_{i=1}^{N} w_{k-1}^{(i)} p(x_k|x_{k-1}^{(i)}). \quad (40)$$

One can always get a particle mixture approximation for $p(x_k|Y^{k-1})$ as

$$p(x_k|Y^{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \delta(x_k - x_{k|k-1}^{(i)}) \quad (41)$$

where $x_{k|k-1}^{(i)}$ is sampled from $p(x_k|x_{k-1}^{(i)})$. When the new measurement y_k comes, the new particles and weights are computed by sampling from an importance density $\mu(x_k|x_{k-1}^{(i)}, y_k)$ as

$$x_k^{(i)} \sim \mu(x_k|x_{k-1}^{(i)}, y_k) \quad (42)$$

$$w_k^{(i)} \propto p(y_k|x_k^{(i)}) \frac{p(x_k^{(i)}|x_{k-1}^{(i)})}{\mu(x_k^{(i)}|x_{k-1}^{(i)}, y_k)} \quad (43)$$

where

$$\sum_{i=1}^{N} w_k^{(i)} = 1. \quad (44)$$

Finally, a resampling step is performed to maintain the statistical support. Sampling importance resampling (SIR) is a common standard method, see [2] for details.

Information Theory

Technically, *information* is a measure of the accuracy to which the value of a stochastic variable is known. This section introduces some important definitions and results from information theory, see e.g. [5] for details. The *differential entropy* $H(p(x))$ of a continuous random variable x with density $p(x)$ is defined as

$$\mathcal{H}(p(x)) = -E_x\{\ln p(x)\} = -\int p(x) \ln p(x) dx. \quad (45)$$

It can be shown [5] that the differential entropy of a normal distribution, with mean μ and covariance matrix P, is

$$\begin{aligned} \mathcal{H}(p(x)) &= \frac{1}{2} \ln\left((2\pi e)^n \det P\right) && (46) \\ &= -\frac{1}{2} \ln\left((2\pi e)^{-n} \det \mathcal{Y}\right) && (47) \end{aligned}$$

where n is the size of random variable and $\mathcal{Y} = P^{-1}$ is the information matrix, defined in Section 4. In the normal distribution case the information matrix is equivalent to the Fisher information matrix. Thus, the entropy is a monotonic function of the determinant of the information matrix and, hence, minimizing the entropy is equivalent to maximizing the Fisher information in the Gaussian case. We also note that this is equivalent to *D-optimal* design in the vocabulary of *experiment design* [7]. Other possible suggestions for criterion from experiment design include *A-optimal* design, i.e., minimizing the trace of the covariance, and *E-optimal* design, i.e., minimizing the maximum eigenvalue of the covariance matrix.

In estimation theory we are interested in the differential entropy of the posterior distribution $p(x|Y^k)$. An interesting recursive relation is obtained by taking the logarithm and the expectation of both sides of the update equation (10), namely

$$-\mathcal{H}(p(x|Y^k)) = -\mathcal{H}(p(x|Y^{k-1})) + E\left\{ \ln \frac{p(y_k|x)}{p(y_k|Y^{k-1})} \right\}. \quad (48)$$

The negative differential entropy can be considered as an *entropic information*, and we see that the posterior entropic information after the update is the sum of the prior entropic information and the information about x contained in the observation y_k, or in other words, the *mutual information* of x and y_k. Thus, the entropic information following an observation is increased by an amount equal to the information inherent in the observation. Compare this to the update of the information matrix in the Information filter (25).

Differential Entropy Approximation

The particle mixture approximation is an useful representation of a probability density that can be used in estimation. Unfortunately, it is not straightforward to compute the differential entropy of the underlying density function from the particle mixture since the differential entropy of the particle set is minus infinity. This is indicated by the fact that the differential entropy of a normal distribution goes to minus infinity as the determinant of the covariance goes to zero, see (46). Since the particle mixture is a weighted sum of impulses, which can be considered as normal densities with zero covariance, it has unbounded entropy from below.

One approach to overcome this problem is to represent the density as a sum of Gaussians where the position of each Gaussian is given by the particles. However, it is not clear how the parameters of the Gaussian kernels should be chosen, and the computational cost of the entropy calculation is also large for this approach.

We are instead proposing an alternative approximation of the differential entropy of a density represented by a particle mixture. By using Bayes rule (9) we have

$$p(x_k|Y^k) = \frac{p(y_k|x_k)p(x_k|Y^{k-1})}{p(y_k|Y^{k-1})} \quad (49)$$

and we obtain the following expression of the differential entropy

$$
\begin{aligned}
\mathcal{H}(p(x_k|Y^k)) &= -\int p(x_k|Y^k)\ln p(x_k|Y^k)dx_k \\
&= -\int p(x_k|Y^k)\ln p(y_k|x_k)dx_k \\
&\quad -\int p(x_k|Y^k)\ln p(x_k|Y^{k-1})dx_k \\
&\quad +\underbrace{\int p(x_k|Y^k)dx_k}_{=1}\ln p(y_k|Y^{k-1})
\end{aligned}
\tag{50}
$$

The last term $\ln p(y_k|Y^{k-1})$ is a constant and by marginalization this term can be expressed as

$$
\ln p(y_k|Y^{k-1}) = \int p(y_k|x_k)p(x_k|Y^{k-1})dx_k. \tag{51}
$$

If we now substitute (51) into (50), make use of the particle mixture in (41) and substitute the density (40) into the second term in (50), we can form an approximation of the differential entropy as

$$
\begin{aligned}
\mathcal{H}(p(x_k|Y^k)) &\approx -\sum_j w_k^{(j)}\ln p(y_k|x_k^{(j)}) \\
&\quad -\sum_j w_k^{(j)}\ln\sum_i w_{k-1}^{(i)}p(x_k^{(j)}|x_{k-1}^{(i)}) \\
&\quad +\ln\sum_i w_{k-1}^{(i)}p(y_k|x_{k|k-1}^{(i)}).
\end{aligned}
\tag{52}
$$

This approximation has been implemented and tested with a one-dimensional sum-of-Gaussian density with known differential entropy. The computational cost of the differential entropy (52) is $\mathcal{O}(N^2)$ due to the double sum term.

Differential Entropy Gradient Approximation

In a similar way we can derive the gradient of the differential entropy as

$$
\begin{aligned}
\frac{\partial}{\partial u}\mathcal{H}(p(x_k|Y^k)) &= -\int\frac{\partial}{\partial u}p(x_k|Y^k)\ln p(x_k|Y^k)dx_k \\
&= -\int\frac{\partial p(x_k|Y^k)}{\partial u}\ln p(x_k|Y^k)dx_k \\
&\quad -\underbrace{\int\frac{\partial p(x_k|Y^k)}{\partial u}dx_k}_{=0}.
\end{aligned}
\tag{53}
$$

To realize that the second term is zero, take the derivative of both sides of the requirement

$$
\int p(x_k|Y^k)dx_k = 1 \tag{54}
$$

with respect to u. Using Bayes rule (49) in (53), the gradient expression becomes

$$
\begin{aligned}
&\frac{\partial}{\partial u}\mathcal{H}(p(x_k|Y^k)) \\
&= -\int\frac{\partial p(x_k|Y^k)}{\partial u}\left(\ln p(y_k|x_k)+\ln p(x_k|Y^{k-1})\right)dx_k \\
&+\ln p(y_k|Y^{k-1})\underbrace{\int\frac{\partial}{\partial u}p(x_k|Y^k)dx_k}_{=0}
\end{aligned}
\tag{55}
$$

As in the derivation of the differential entropy approximation we use the mixture (41) for $p(x_k|Y^{k-1})$. When a particle mixture approximation for jth element of the gradient given as

$$
\frac{\partial}{\partial u_j}p(x_k|Y^k) \approx \frac{1}{\epsilon}\sum_{i=1}^{N}\Delta w_k^{j,(i)}\delta(x_k - x_k^{(i)}) \tag{56}
$$

is available, the differential entropy gradient becomes

$$
\begin{aligned}
&\frac{\partial}{\partial u_j}\mathcal{H}(p(x_k|Y^k))|_{u=\hat{u}} \\
&\approx -\frac{1}{\epsilon}\sum_{i=1}^{N}\Delta w_k^{j,(i)}\Big(\ln p(y_k|x_k^{(i)}) \\
&+\ln\Big[\sum_{\ell=1}^{N}w_{k-1}^{(\ell)}p(x_k^{(i)}|x_{k-1}^{(\ell)})\Big]\Big)dx_k.
\end{aligned}
\tag{57}
$$

Equations (52) and (57) constitute some of the important contributions of this paper and they are used in the two algorithms that will be presented in the next section.

6. A Particle Filter Approach

In this section the differential entropy is used as the objective function in the planning problem. We first describe the planning problem and introduce stochastic approximation search. Then the computations are described in detail and simulation results are presented.

The Planning Problem

We define the planning problem as in (6), repeated here for convenience,

$$
\begin{aligned}
\min_{\pi^{M-1}}\quad & E[L(x_M)|I^0] \\
\text{s.t.}\quad & u_k \in \mathcal{U} \\
& x_{k+1} = f(x_k, w_k) \\
& x_0 \sim p(x_0|I^0) \\
& w_k \sim \mathcal{N}(0, Q) \\
& x_{k+1}^s = f^s(x_k^s, u_k) \\
& x_0^s = \bar{x}_0^s \\
& y_k = h(x_k, x_k^s, e_k) \\
& e_k \sim \mathcal{N}(0, R).
\end{aligned}
\tag{58}
$$

Now we use our expressions of the differential entropy to define a new loss function

$$
E[L(x_M)|I^0] = E[\mathcal{H}(p(x_M|Y^M, Y^0))|Y^0] \tag{59}
$$

where Y^0 is the measurement we have received and Y^M is a random variable representing the future measurements. Taking the expectation in (59), which should be done over both

x_M and Y_M, is not possible analytically for the general problem definition. A Monte-Carlo based averaging strategy like

$$L(x_M) = \sum_{j=1}^{N_y} \hat{\mathcal{H}}\left(p(x_M|Y^{M\,(j)}, Y^0)\right) \tag{60}$$

may be applicable, where the independent sequences $\{Y^{M\,(j)}\}$ are generated from the density $p(Y^M|Y^0)$. However, the computational load is $\mathcal{O}(N_y N^2)$ and in practice N_y needs to be very large. We, here, propose a stochastic gradient based optimization algorithm using the differential entropy expression $\mathcal{H}(p(x_M|Y^M, Y^0))$, which is a stochastic function due to the random process and the measurement noise, as follows

$$\hat{\pi}^{m+1} = \hat{\pi}^m - a_m \frac{\partial}{\partial \pi} \mathcal{H}(p(x_M|Y^M, Y^0))|_{\pi=\hat{\pi}^m}. \tag{61}$$

However, even for such an algorithm the noise will cause serious problems. Therefore, common noise realizations [15] will be used for entropy and entropy gradient evaluations.

Before detailed descriptions of our planning algorithms are presented, stochastic gradient search is introduced in the next sub-section.

Stochastic Approximation

In Section 4 we assume that the loss function is deterministic, but we now have to consider the stochastic case, since only noisy information is available. Robbins and Monro are often referred to as the people who introduced modern stochastic search algorithms. They introduced a *stochastic approximation* method for root-finding when only noisy measurements of the objective function are available [19]. For a good introduction to stochastic optimization see [22].

Let the loss function be given as

$$L(\theta) = E[L(\theta, W)] \tag{62}$$

where W is a random variable causing the stochastic effects of the measurement $L(\theta, W)$ of the loss function. The stochastic gradient is defined as

$$g(\theta) = \frac{\partial L(\theta, W)}{\partial \theta} \tag{63}$$

and the stochastic gradient algorithm is

$$\hat{\theta}^{k+1} = \hat{\theta}^{k+1} - a_k \left. \frac{\partial L_k(\theta, W_k)}{\partial \theta} \right|_{\theta=\hat{\theta}}. \tag{64}$$

For batch processing the gradient above can be replaced by its sample mean. If the gradient can not be calculated explicitly, there are *gradient free* methods for gradient approximation based on values of the loss function. One of the well-known algorithms is the *finite-difference stochastic approximation* (FDSA). The gradient is formed from noisy loss function measurements where each element of θ is perturbed [4]. A drawback with this algorithm is that the computational complexity increases with the dimension of the optimization variable θ. An alternative algorithm is the *simultaneous perturbation stochastic approximation* (SPSA) [22] which uses two, regardless of the dimension of θ, loss function measurements. Nevertheless, the SPSA achieves the same level of statistical accuracy as the FDSA under rather general conditions [22].

The proposed idea in this paper is to use the same noise realization for every estimate of the gradient. Using common random numbers gives better results in gradient evaluations [15].

A Gradient based Algorithm

The algorithm is based on approximative gradient expression (57). For each iteration in the stochastic gradient search algorithm (61) the estimate of the control signal sequence is updated and the gradient approximation is computed as in Algorithm 1. Note that the control signal sequence is defined slightly different here than in previous sections. Here we assume that the $u_k = u_k(I^0)$ and, furthermore, we let sequences like $\{u_0, u_1, ..., u_M\}$ be denoted as $u_{0:M}$.

Algorithm 1 (Gradient Calculation) Suppose we are given the particle representation of $p(x_0|I^0)$ as

$$p(x_0|I^0) = \sum_{i=1}^{N} w_0^{(i)} x_0^{(i)} \tag{65}$$

and a current control sequence estimate $\hat{u}_{0:M-1}$. Then the gradient of the posterior density $p(x_M|Y^M)$ with respect to the input sequence $u_{0:M-1}$ evaluated at the current control input sequence $u_{0:M-1} = \hat{u}_{0:M-1}$ i.e.,

$$\left. \frac{\partial p(x_M|Y^M)}{\partial u_{0:M-1}} \right|_{u_{0:M-1}=\hat{u}_{0:M-1}} \tag{66}$$

is calculated using the following steps.

1. Target State Sequence Generation

(a) Select a single state realization $\bar{x}_0^1$ as

$$P(\bar{x}_0^1 = x_0^{(j)}) = w_0^{(j)} \tag{67}$$

for $j = 1, \ldots, N$.

(b) Generate a single realization of the process noise sequence $v_{0:M-2}^1$

(c) Obtain the single state sequence realization $x_{1:M}^1$ using the single realization of the process noise sequence $v_{0:M-2}^1$ and the single state realization $\bar{x}_0^1$ as

$$x_{i+1}^1 = f(x_i^1, v_i^1), \tag{68}$$
$$x_0^1 = \bar{x}_0^1 \tag{69}$$

for $i = 0, \ldots, M-1$.

2. Measurement Generation

(a) Generate $M+1$ control signal sequences $\{u^j_{0:M-1}\}_{j=0}^M$ where $u^0_{0:M-1} = \hat{u}_{0:M-1}$ is current estimated input sequence. Each $u^j_{0:M-1}$ for $1 \le j \le M$ correspond to the perturbation of the current estimated input sequence $\hat{u}_{0:M-1}$ defined as

$$u^j_i = \begin{cases} \hat{u}_i + \epsilon, & i = j-1 \\ \hat{u}_i, & \text{otherwise} \end{cases} \tag{70}$$

for $i = 0, \dots, M-1$ and $j = 1, \dots, M$.

(b) Obtain the $M+1$ sensor trajectories $\{x^{s,j}_{1:M}\}_{j=0}^M$ corresponding to each of the control signal sequences $\{u^j_{0:M-1}\}_{j=0}^M$.

(c) Generate a single realization of the measurement noise sequence $e^1_{1:M}$.

(d) Calculate $M+1$ measurement sequences $\{y^j_{1:M}\}_{j=0}^M$ corresponding to control signal sequences, sensor trajectories $\{x^{s,j}_{1:M}\}_{j=0}^M$ using the single measurement noise sequence realization $e^1_{1:M}$ and the single state sequence realization $x^1_{1:M}$ as

$$y^j_i = h(x^1_i, x^{s,j}_i, e^1_i) \tag{71}$$

for $i = 1, \dots, M$ and $j = 0, \dots, M$.

3. Calculate $\frac{\partial p(x_M|y_{0:M})}{\partial u_{0:M-1}}$ using marginal particle filter [14]. Particles and weights in this filter are called as $\{\mathsf{x}^{(i)}_k, \mathsf{w}^{(i)}_k\}_{i=1}^{\mathsf{N}}$ to avoid confusion with the similar quantities in the main particle filter. In general, the number of particles N used in the gradient calculating particle filter might be different than the number of particles N used in the main particle filter. Hence, the particles $\{\mathsf{x}^{(i)}_k\}_{i=1}^{\mathsf{N}}$ and weights $\{\mathsf{w}^{m,(i)}_k\}_{i=1}^{\mathsf{N}}$ for $m = 0, \dots, M$ are initialized by sampling them from the main particle filter's particle distribution as follows.

$$P(\mathsf{x}^{(i)}_0 = x^{(j)}_0) = w^{(j)}_0 \tag{72}$$

$$\mathsf{w}^{m,(i)}_0 = \frac{1}{\mathsf{N}} \tag{73}$$

for $j = 1, \dots, N$, $i = 1, \dots, \mathsf{N}$ and $m = 0, \dots, M$. Setting $k = 1$, a single step of the algorithm is given below.

(a) Generate pre-likelihoods $\lambda^{(i)}_k$ as

$$\lambda^{(i)}_k \propto \mathsf{w}^{0,(i)}_{k-1} p(y^0_k | \bar{\mathsf{x}}^{(i)}_k, x^{s,0}_k) \tag{74}$$

with $\sum_{i=1}^{\mathsf{N}} \lambda^{(i)}_k = 1$. Here, state vector $\bar{\mathsf{x}}^{(i)}_k$ is obtained from the corresponding state vector $\mathsf{x}^{(i)}_{k-1}$ by a deterministic relation. Most of the times

$$\bar{\mathsf{x}}^{(i)}_k = f(\mathsf{x}^{(i)}_{k-1}, 0). \tag{75}$$

(b) Sample indices $\{i_\ell\}_{\ell=1}^{\mathsf{N}}$ as follows

$$P(i_\ell = j) = \lambda^{(j)}_k \tag{76}$$

for $1 \le \ell, j \le \mathsf{N}$.

(c) Prediction Update: Sample $\mathsf{x}^{(j)}_k$ as

$$\mathsf{x}^{(j)}_k \sim p(\mathsf{x}_k | \mathsf{x}^{(i_j)}_{k-1}) \tag{77}$$

for $j = 1, \dots, \mathsf{N}$.

(d) Measurement Update: Calculate the weights $\{\mathsf{w}^{m,(i)}_k\}_{i=1}^{\mathsf{N}}$ for $m = 0, \dots, M$ as

$$\mathsf{w}^{m,(j)}_k \propto p(y^m_k | \mathsf{x}^{(j)}_k, x^{s,m}_k) \frac{\sum_{i=1}^{\mathsf{N}} \mathsf{w}^{m,(i)}_{k-1} p(\mathsf{x}^{(j)}_k | \mathsf{x}^{(i)}_{k-1})}{\sum_{i=1}^{\mathsf{N}} \lambda^{(i)}_k p(\mathsf{x}^{(j)}_k | \mathsf{x}^{(i)}_{k-1})} \tag{78}$$

with $\sum_{j=1}^{\mathsf{N}} \mathsf{w}^{m,(j)}_k = 1$.

(e) If $k = M$, stop. The approximation of $\frac{\partial p(x_M|Y^M)}{\partial u_j}$ is given as

$$\frac{\partial p(x_M|Y^M)}{\partial u_j}(\mathsf{x}_M) \approx \frac{1}{\epsilon} \sum_{i=1}^{\mathsf{N}} (\mathsf{w}^{j,(i)}_M - \mathsf{w}^{0,(i)}_M) \delta_{\mathsf{x}^{(i)}_M}(\mathsf{x}_M) \tag{79}$$

Note that this result gives us the approximation (56) with $\Delta w^{j,(i)}_M = \mathsf{w}^{j,(i)}_M - \mathsf{w}^{0,(i)}_M$. Otherwise (if $k \ne M$), set $k = k+1$, go to step 3a.

4. Finally, the gradient approximation of the differential entropy is computed as in (57) which gives

$$\begin{aligned} &\frac{\partial}{\partial u_j} \mathcal{H}(p(x_M|Y^M))|_{u=u^0} \\ &\approx -\frac{1}{\epsilon} \sum_{i=1}^{\mathsf{N}} (\mathsf{w}^{j,(i)}_M - \mathsf{w}^{0,(i)}_M) \Big(\ln p(y_M | \mathsf{x}^{(i)}_M) \\ &\quad + \ln(\sum_{\ell=1}^{\mathsf{N}} \mathsf{w}^{(\ell)}_{M-1} p(\mathsf{x}^{(i)}_M | \mathsf{x}^{(\ell)}_{M-1})) \Big) \end{aligned} \tag{80}$$

The Marginal Particle Filter has $\mathcal{O}(\mathsf{N}^2)$ complexity and the gradient calculation in the last step is also $\mathcal{O}(\mathsf{N}^2)$. Thus, the overall computational complexity of the algorithm is $\mathcal{O}((M+1)\mathsf{N}^2)$, where M is the number of planning steps. This is significant large complexity and future work is to see how this can be reduced. There are techniques for reducing cost of the Marginal Particle Filter to $\mathcal{O}(\mathsf{N} \ln \mathsf{N})$, see [14]. Using such a technique yields $\mathcal{O}(M \mathsf{N} \ln \mathsf{N} + \mathsf{N}^2)$ complexity of the algorithm.

A Gradient free Algorithm

It is also possible to apply a gradient-free approach, based on the loss function approximation in (52), to the planning problem. For each iteration in the stochastic search algorithm (e.g. FDSA or SPSA), the gradient is estimated based on "measurements" of the differential entropy computed as in Algorithm 2.

Algorithm 2 (Differential entropy) Suppose we are given the particle representation of $p(x_0|I^0)$ as

$$p(x_0|I^0) = \sum_{i=1}^{N} w_0^{(i)} x_0^{(i)}. \tag{81}$$

Then the differential entropy of the posterior density $p(x_M|Y^M)$ for a control signal sequence $u_{0:M-1}$ is computed using the following steps.

1. Target State Sequence Generation; the same as in Algorithm 1.

2. Measurement Generation; basically the same as in Algorithm 1, but in 2(a) the control signal sequences are generated according to the SA algorithm.

3. For each control signal sequence $u^j_{0:M-1}$

(a) Initialize a temporary particle filter by sampling from the initial posterior density $p(x_0|Y^0)$ resulting in a particle set $\{\mathsf{x}_0^{(i)}, \mathsf{w}_0^{(i)}\}_{i=1}^{\mathsf{N}}$.

(b) Update the particle filter with the observation sequence $y^j_{1:M}$ according to the steps in (42)-(43).

(c) Compute differential entropy approximation according to (52).

Notice that the particle filters obtaining the differential entropy values for $u^j_{0:M-1}$ must use the same noise realizations for $j = 0, \ldots, M$.

As noted before, the computational cost of the differential entropy (52) is $\mathcal{O}(\mathsf{N}^2)$. If a SIR particle filter with a $\mathcal{O}(\mathsf{N})$ complexity is used, then the overall computational complexity of the algorithm is $\mathcal{O}(M\mathsf{N} + \mathsf{N}^2)$, where M is the number of planning steps. One thing worth remembering is that a gradient free stochastic approximation algorithm in general requires more iterations than an algorithm where the gradient is given.

Simulation Results

Figure 5 shows the result from a simulation with the same conditions as before, see Table 1. The information criterion is the differential entropy and Algorithm 1 is used.

As in most problems where SA is applied, there are problems with determining good parameter values, e.g. the step size of the SA algorithm. Suitable values of the step size vary much depending on the planning conditions, even in the simplified planning problem considered here. Some type of adaptive methods is required to tune the values.

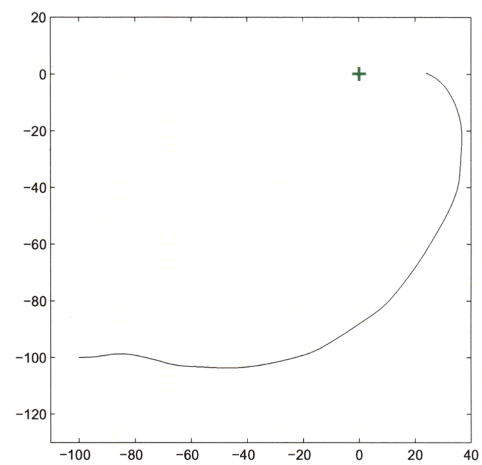

Figure 5. Simulation result from a differential entropy based planning.

7. Conclusions

In this work we consider a target tracking scenario where a moving observer with a bearings-only sensor is tracking a target. The tracking performance is highly dependent on the trajectory of the sensor platform, and the problem is how it should maneuver for optimal estimation performance.

The planning problem can be considered as a stochastic optimal control problem and if a sub-optimal control scheme, for instance certainty equivalence control, is used with a Gaussian target uncertainty assumption, the resulting problem is deterministic. However, the problem is non-linear and nonconvex which still makes it a challenge to solve optimally.

Particle mixture is a popular approach to handle more general target densities in the estimation field. To compute good information metrics based on particle mixtures is not straightforward. In this work we propose a differential entropy calculation method for particle mixtures and derive a stochastic gradient search algorithm which is applied to the planning problem. However, the cost of being able to handle non-Gaussian target densities are a much higher computational load. Furthermore, there are more parameters to tune in the optimization routine.

Acknowledgments

This work is part of the ARCUS project, a project in the FMV founded TAIS-programme, Sweden.

References

[1] R. Bellman. *Dynamic Programming*. Princeton Univ. Press, 1957.

[2] Niclas Bergman. *Recursive Bayesian Estimation:*

Navigation and Tracking Applications. PhD thesis, Linköping Studies in Science and Technology, May 1999.

[3] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 2nd edition, 2001.

[4] Julius R. Blum. Multidimensional stochastic approximation methods. *The Annals of Mathematical Statistics*, 25(4):737–744, 1954.

[5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[6] A. Doucet, N. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[7] V. Fedorov. *Theory of optimal experiments*. Academic press, New York ed., 1972.

[8] A. A. Feldbaum. Dual control theory. I-IV. *Automation Remote Control*, 21,22:874–880, 1033–1039, 1–12, 109–121, 1960-61.

[9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107–113, April 1993.

[10] B. Grocholsky, R. Swaminathan, J. Keller, V. Kumar, and G. Pappas. Information driven coordinated air-ground proactive sensing. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2211–2216, April 2005.

[11] Gabriel M. Hoffmann and Claire J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *Unpublished, available at* $hoffmann.stanford.edu/publications\_.html$, 2007.

[12] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[13] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice Hall, 2000.

[14] Mike Klaas. Toward practical N^2 Monte Carlo: The marginal particle filter. In *in Uncertainty in Artificial Intelligence*, 2005.

[15] N. L. Kleinman, J. C. Spall, and D. Q. Naiman. Simulation-based optimisation with stochastic approximation using common random numbers. *Management Science*, 45(11):1571–1578, 1999.

[16] A. Logothetis, A. Isaksson, and R. J. Evans. An information theoretic approach to observer path design for bearings-only tracking. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 4, pages 3132–3137, San Diego, CA, USA, December 1997.

[17] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Prentice Hall, 1994.

[18] G. Poyiadjis, S. S. Singh, and A. Doucet. Gradient-free maximum likelihood parameter estimation with particle filters. pages 3062–3067, June 2006.

[19] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[20] S. Singh, N. Kantas, B. Vo, A. Doucet, and Robin J. Evans. Simulation-based optimal sensor scheduling with application to observer trajectory planning. *Automatica (Journal of IFAC)*, 43:817–830, May 2007.

[21] Edward Sondik. *The optimal control of partially observable markov processes*. PhD thesis, Stanford University, 1971.

[22] J. C Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Hoboken, NJ, 2003.

[23] O. Tremois and J. P. Le Cadre. Optimal observer trajectory in bearings-only tracking for manoeuvring sources. In *Radar, Sonar and Navigation, IEE Proceedings -*, volume 146, pages 31–39, February 1999.

BIOGRAPHY

Per Skoglar received M.S. in electrical engineering from Linköping University, Linköping, Sweden in 2002. Between 2003 and 2007, he was working at the Department of Sensor Systems, FOI, Swedish Defense Research Agency. Since 2007 he has been a Ph.D. student in Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping.

Umut Orguner received B.S., M.S. and Ph.D. degrees all in electrical engineering from Middle East Technical University, Ankara, Turkey in 1999, 2002 and 2006 respectively. Between 1999 and 2007, he was with the Department of Electrical and Electronics Engineering of the same university as a teaching and research assistant. Since January 2007 he has been working as a postdoctoral associate in Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden. His research interests include estimation theory, multiple-model estimation, target tracking and information fusion.

Fredrik Gustafsson is professor in Sensor Informatics at Department of Electrical Engineering, Linköping University, since 2005. He received the M.Sc. degree in electrical engineering 1988 and the Ph.D. degree in Automatic Control, 1992, both from Linköping University. During 1992-1999 he held various positions in automatic control, and in 1999 he got a professorship in Communication Systems. In 2004, he was awarded the Arnberg prize by the Royal Swedish Academy of Science (KVA) and in 2007 he was elected member of the Royal Academy of Engineering Sciences (IVA). His research interests are in stochastic signal processing, adaptive filtering and change detection, with applications to communication, vehicular, airborne and audio systems, where the current focus is on sensor fusion algorithms for navigation and tracking problems. He was an associate editor for IEEE Transactions of Signal Processing 2000-2006 and is currently associate editor for EURASIP Journal on Applied Signal Processing and International Journal of Navigation and Observation.