OPTIMAL STRATEGIES FOR MOBILE ROBOTS BASED ON THE CROSS-ENTROPY ALGORITHM

F. Celeste and F. Dambreville

CEP/Dept. of Geomatics Imagery Perception
94114 Arcueil France
{francis.celeste, frederic.dambreville}@etca.fr

J.-P. Le Cadre

IRISA/CNRS
Campus de Beaulieu
35042 Rennes
lecadre@irisa.fr

ABSTRACT

This paper deals with the problem of optimizing the navigation of an intelligent mobile with respect to the maximization of the performance of the localization algorithm used during execution. It is assumed that a known map composed of features describing natural landmarks in the environment is given. The vehicle is also equipped with a range and bearing sensor to interact with its environment. The measurements are associated with the map to estimate its position. The main goal is to design an optimal path which guarantees the control of a measure of the performance of the map-based localization filter. Thus, a functional of the approximate Posterior Cramer-Rao Bound is used. However, due to the functional properties, classical techniques such as Dynamic Programming is generally not usable. To face that, we investigate a learning approach based on the Cross-Entropy method to stress globally the optimization problem.

1. INTRODUCTION

We are concerned with the task of finding a plan for a vehicle moving around in its environment. That is to say, reaching a given goal position from an initial position. In many application, it is crucial to be able to estimate accurately the state of the mobile during the execution of the plan. So the planning and the execution stages must be drawn conjointly. One way to achieve that is to define trajectories which imply a high performance of the localization algorithm. This problem have been well studied and in most approaches the environment is discretized and described as a graph whose nodes correspond to particular area and edges are actions to move from one place to another. Some of these previous contributions address the problem within a Markov Decision Process (MDP).
In the present paper, we will also use the basis of the constrained MDP framework, as in [1]. Our optimality criterion is based on the Posterior Cramer-Rao bound. However, the nature of the objective function for path planning makes it impossible to perform complete optimization of MDP with classical means. Indeed, we will show that the reward in one stage of our MDP depends on the complete history of the trajectory. To solve the problem, the Cross-Entropy originally used for *rare-events* simulation seemed a valuable tool.

The paper is organized in five parts. In the second section we introduce the problem in details. Section three deals with the Posterior Cramér-Rao bound and its properties. We also derive its formulation for our particular case and the optimization criterion. In section four, we make a short introduction of the Cross-entropy and show how to apply it to our needs. Finally, in section five the results of a first example are discussed.

2. PROBLEM STATEMENT

Let X_k, A_k and Z_k respectively denote the state of the vehicle, the action and the vector of observations at time k. We consider here that the state vector X_k is the position and the orientation of the vehicle in a given reference basis $\mathcal{R}_0 \subset \mathbb{R}^2$, so that $X_k \triangleq [x_k, y_k, \theta_k]'$. The action vector $A_k \triangleq (a_x, a_y)$ is restricted to the set $\mathcal{A}(X_k) \subset \mathbb{R}^2$. The state $\{X_k\}$ motion is governed by a dynamic model and an initial uncertainty, given by :

$$X_0 \sim \pi_0 \triangleq \mathcal{N}(\overline{X_0}, P0)$$
$$X_{k+1} = f(X_k, A_k) + w_k \quad w_k \sim \mathcal{N}(0, Q_k)$$

where f is linear, $\{w_k\}$ a white noise sequence and the symbol "$\sim$" means distributed according to [1].

The map $\mathcal{M}$ is composed of N_f features $(m_i)_{1 \leq i \leq N_f}$ with position $(x_i, y_i) \in \mathcal{R}_0$. At time k, due to the sensor capabilities, the observer received signals from only a few landmarks (see figure 1). Moreover, we do not consider data association and non detection problem for the moment.

[1] The symbol $\mathcal{N}$ means normal

1-4244-0657-9/06/$20.00 ©2006 IEEE

So each measurement is made from one landmark represented in the map. If we denote $I_v(k)$ the indexes of visible landmarks at time k, the measurement vector at time k is $Z_k = \{z^k(j)\}_{j \in I_v(k)}$ with $\forall j \in I_v(k)$:

$$z^k(j) \triangleq \begin{cases} z_r^k(j) = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} + \gamma_r^k(j) \\ z_\beta^k(j) = \arctan_2(\frac{y_j - y_k}{x_j - x_k}) - \theta_k + \gamma_\beta^k(j) \end{cases}$$

where $\{\gamma_r^k(j)\}$ and $\{\gamma_\beta^k(j)\}$ are considered as white Gaussian noise and mutually independent. Nevertheless, a more complex modeling must be considered if we want to take into account correlated errors in the map [2] or observations[3].

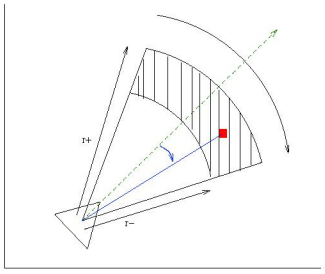

Fig. 1. Sensor model with its visibility area.

2.1. A discrete approach

As in [1], we first formulate the problem within a discrete sequential decision planning framework similar with a Markov Decision Process. Generally, a MDP is defined by its state and action sets, a state transition probabilities and a reward functions. In our case, the state and action spaces are discrete and finite. Indeed the map is discretized in $N_s = N_x \times N_y$ locations and one action is defined as a move between two neighbor points (figure 2). At each point, the mobile can choose one action among N_a to change state. So we can model the process with:

- $S = \{1, ..., N_s\}$ the state space and $A = \{1, ..., N_a\}$ the action space.
- $T_{ss'}(a) \triangleq Pr(s_{k+1} = s' | s_k = s, a_k = a)$ the transition function.
- $R_{ss'}(a)$ the cost function, associated with transition from state s to state s', for an action a.

In our case, we consider that the neighborhood of each state is composed of less than eight states (see figure 2), depending whether it is near obstacles or the border of the map. Moreover each choice of action in a state is equivalent to choose one orientation for the mobile and the displacement between two states is made with a constant velocity. Finding the optimal path between two states s_i and s_f is equivalent to determine the *best* sequence of actions/decisions $V_a^* = (a_0^*, ..., a_K^*)$ allowing to simultaneously connecting them and optimizing a (global) criterion. Moreover, we take into account operational constraints on

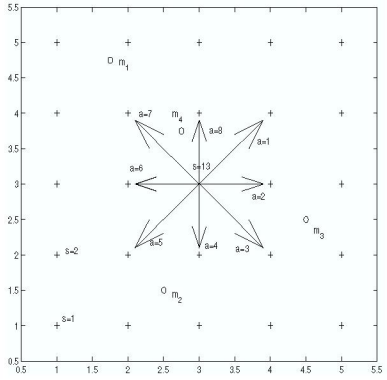

Fig. 2. grid example and features(o). States (crosses) and actions (arrows).

the mobile dynamic between two following epochs as in [1]. It is possible to do that by defining one *authorized transition matrix* $\delta(a_k, a_{k-1})$ which indicates actions that can be chosen at time k according to the choice at time $k-1$. For example, if only $[-\frac{\pi}{4}, \frac{\pi}{4}]$ headings controls are allowed, if $a_{k-1} = 1$ then $a_k \in \{1, 7, 2\}$.

In the MDP context when the reward is completely known, optimal policy can be computed using Dynamic Programming technique or similar algorithms such as Value Iteration and Policy Iteration [8]. However, the basic assumptions of MDP are no longer valid in our context. More precisely, any Posterior Cramér-Rao Bound (PCRB) based functional is not separable (with respect to actions) and even not monotonic. Unable to apply a general monotonic "comparison" principle (see [2]), the MDP principle is irrelevant.

3. POSTERIOR CRAMÉR-RAO BOUND

3.1. Definition

In this section, we briefly remind the properties of the Cramér-Rao bound for estimation problem. Let $\hat{X}(Z)$ be an estimator of an unknown random vector $X \in \mathbb{R}^d$ based on random observations Z. When $\hat{X}(Z)$ is also an unbiased estimate the Posterior Cramér-Rao Bound is given by the inverse of the Fisher Information Matrix (FIM) F [9] and

[2] e.g. Markov Random Fields
[3] e.g. colored noise or biases

"measures " [4] the minimum mean square errors of $\hat{X}(Z)$:

$$E\{(\hat{X}(Z)-X)(\hat{X}(Z)-X)^T\} \succeq F^{-1}(X,Z)\ , \quad (1)$$

where "$A \succeq B$" means $(A-B)$ is positive semi-definite. Let Δ_X^X be the second-order partial derivatives operator. If X is a n-dimensional random vector and $p_{x,z}(X,Z)$ the joint probability density of the pair (X,Z), F is a $n \times n$ matrix which can be derived from the following formula :

$$F = E\left[-\Delta_X^X \log p_{x,z}(X,Z)\right]$$

For the filtering case, we estimate at time k the state X_k based on the collection of observations $Z_{1:k} = (Z_1, ..., Z_k)$ received since the beginning until k. If we note $\hat{X}_k = \hat{X}_k(Z_{1:k})$ the estimate at time k and $V_a^k = \{a_1, ..., a_k\}$ the sequence of decisions chosen by the observer, it can be shown [4] that :

$$E\{\left(\hat{X}_k - X_k\right)\left(\hat{X}_k - X_k\right)^T | V_a^k\} \succeq J_k^{-1}(V_a^k)$$

where $J_k^{-1}(V_a^k)$ is the lower-right block matrix of the FIM of the vector of state history $X_{0:k} = (X_0, ..., X_k)$ based on $Z_{1:k}$.

3.2. Tichavsky PCRB recursion

A remarkable contribution of Tichasky et al. [4] was to introduce a Ricatti like recursion for computing J_k :

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1}D_k^{12} \quad (2)$$

where

$$\begin{aligned} D_k^{11} &= E\{-\triangle_{X_k}^{X_k} log(p(X_{k+1}|X_k)\} \ , & (3)\\ D_k^{12} &= E\{-\triangle_{X_k}^{X_{k+1}} log(p(X_{k+1}|X_k)\} \ , & (4)\\ D_k^{21} &= [D_k^{12}]^T \ , & (5)\\ D_k^{22} &= E\{-\triangle_{X_{k+1}}^{X_{k+1}} log(p(X_{k+1}|X_k)\} + & (6)\\ & \quad E\{-\triangle_{X_{k+1}}^{X_{k+1}} log(p(Z_{k+1}|X_{k+1})\ .\} & (7) \end{aligned}$$

The initial information matrix J_0 can be calculated from π_0. The dynamics being linear we have :

$$\begin{aligned} D_k^{11} = Q_k^{-1}, \quad D_k^{12} = -Q_k^{-1}\ , \\ D_k^{12} = -Q_k^{-1}, \quad D_k^{22} = Q_k^{-1} + J_{k+1}(Z)\ . \end{aligned} \quad (8)$$

where $J_{k+1}(Z)$ is given by:

$$J_k(Z) = E_{X_k}\{\sum_{j \in I_v(X_k)} H(X_k,j)^T R_k^{-1} H(X_k,j)\}\ ,$$

[4] Actually, this is a lower bound which may be reasonably accurate

and R_k is the covariance matrix of the combined visible observations which only depends on the current position. For our observation model , the matrix $H(X_k, j)$ is as follows :

$$H(X_k,j) = \begin{pmatrix} \frac{(x_k-x_j)}{\sqrt{d_k^j}} & \frac{(y_k-y_j)}{\sqrt{d_k^j}} \\ \frac{(y_k-y_j)}{d_k^j} & \frac{-(x_k-x_j)}{d_k^j} \end{pmatrix}\ , \quad (9)$$

$d_k^j = (x_k-x_j)^2+(y_k-y_j)^2$. Obviously there is no explicit expression for $J_k(Z)$ as the observation model is nonlinear, so that it is necessary to resort to Monte Carlo simulation to estimate it.

3.3. A criterion for path planning

We are interested in finding one or more paths connecting two points which maximizes a functional ϕ of the PCRB along the trajectory [1]. We consider a functional which depends on the determinant of the history of PCRB $\{J_1, ..., J_K\}$. The determinant is linked to the volume of the ellipsoid of error of the position estimate :

$$\phi(J_{1:K}) = -\sum_{k=0}^{K} w_k\ \det(J_k)$$

As classical method such as Dynamic Programming is irrelevant [2], we investigate a learning approach based on the Cross Entropy (CE) methods developed by [5].

4. THE CROSS ENTROPY ALGORITHM

4.1. A short introduction

The Cross Entropy method was first used to estimate the probability of *rare events*[5, 6]. A rare event is an event with very small probabilities. It was then adapted for optimization assuming that sampling around the optimum of a function is a rare event.

4.1.1. Simulation of rare event

Let X a random variable on a space $\mathcal{X}$, p_x its probability density function (pdf) and ϕ a function on $\mathcal{X}$. Suppose, we are concerned with estimating $l(\gamma)$ the probability of the event $F_\gamma = \{x \in \mathcal{X} | \phi(x) \geq \gamma\}$ with $\gamma \in \mathbb{R}^+$. F_γ is a rare event if $l(\gamma)$ is very small. An unbiased estimate can be obtained via Crude Monte Carlo simulation. Given a random sample $(x_1, \cdots, x_N)$ drawn from p_x, this estimate is :

$$\hat{l}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma\right]$$

For *rare event*, the variance of $\hat{l}(\gamma)$ is very high and it is necessary to increase N to improve the estimation. The es-

timate properties can be improved with *variance minimization* technique such as *importance sampling* where the random sample is drawn from a more appropriate probability density function q. It can be easily shown that the optimal q^* pdf is given by $I\left[f(x) > \gamma\right] p_x(x)/l(\gamma)$. Nevertheless, q^* depends on $l(\gamma)$ which needs to be estimated. To get round this difficulty, it can be convenient to choose q in a family of pdfs $\{\pi(.,\lambda)|\lambda \in \Lambda\}$. The idea is to find the optimal parameter λ^* such as $\mathcal{D}(q^*, \pi(.,\lambda))$ is minimized where $\mathcal{D}$ is the *Kullback-Leibler* "pseudo-distance" :

$$\mathcal{D}(p,q) = \mathbb{E}_p \ln \frac{p}{q} = \int_X p(x) \ln p(x) dx - \int_X q(x) \ln p(x) dx$$

Minimizing $\mathcal{D}(q^*, \pi(.,\lambda))$ is equivalent to maximize $\int_X q^*(x) \ln \pi(.,\lambda) dx$ which implies :

$$\lambda^* \in \arg\max_{\lambda \in \Lambda} \mathbf{E}_p \left(I\left[\phi(x) \geq \gamma\right] \ln \pi(x,\lambda) \right) \tag{10}$$

The computation of the expectation in 10 must also be done using importance sampling. So we need a change of measure q, drawing one sample $(x_1, \cdots, x_N)$ from q and estimate λ^* as follows :

$$\hat{\lambda^*} \in \arg\max_{\lambda \in \Lambda} \sum_{i=1}^{N} \left(I\left[\phi(x_i) \geq \gamma\right] \frac{p_x(x_i)}{q(x_i)} \ln \pi(x_i,\lambda) \right) \tag{11}$$

However, q is still not known in equation 11. The C.E algorithm tries to overcome this difficulty by constructing adaptively a sequence of parameters $(\gamma_t | t \geq 1)$ and $(\lambda_t | t \geq 1)$ such as :

- F_{γ_1} is not a rare event.
- $F_{\gamma_{t+1}}$ is not a rare event for $\pi(.,\lambda_t)$.
- $\lim_{t\to\infty} \gamma_t = \gamma$.

More precisely, given $\rho \in]0,1[$:

- choose λ_0 such as $\pi(.,\lambda_0) = p_x$.
- draw $(x_1, \cdots, x_N)$ from $\pi(.,\lambda_{t-1})$.
- sort in increasing order $(\phi(x_1), \cdots, \phi(x_N))$ and evaluate the $(1-\rho))$ quantile γ_t
- compute λ_t as

$$\lambda_t \in \arg\max_{\lambda \in \Lambda} \sum_{i=1}^{N} \left(I\left[\phi(x_i) \geq \gamma_t\right] \frac{p_x(x_i)}{\pi(x_i,\lambda_{t-1})} \ln \pi(x_i,\lambda) \right)$$

- if $\gamma_t < \gamma$, set $t = t+1$ and go to step 2. Else estimate the probability of F_γ with :

$$\hat{l}(\gamma) = \frac{1}{N} \sum_{i=1}^{N} I\left[\phi(x_i) \geq \gamma\right] \frac{p_x(x_i)}{\pi(x_i,\lambda_t)}$$

This is the main C.E algorithm but other versions can be found in [5].

4.1.2. application to optimization

The C.E. was adapted to solve optimization problem. Consider the optimization problem :

$$\phi(x^*) = \gamma^* = \max_{x \in \mathcal{X}} \phi(x) \tag{12}$$

The principle of C.E for optimization is to translate the problem 12 into an *associated stochastic problem* and then solved it adaptively as the simulation of a rare event. If $\gamma*$ is the optimum of ϕ, F_{γ^*} is generally a rare event. The main idea is to define a family $\pi(.,\lambda)|\lambda \in \Lambda$ and iterate enough the C.E algorithm such as $\gamma_t \overset{\infty}{\to} \gamma^*$ to draw samples around the optimum. Unlike the other local random search algorithm such as simulated annealing which used the assumption of local neighborhood hypothesis, the CE method tries to solve globally the problem.
Given a selection rate ρ, a well-suited family of pdf $\pi(.,\lambda)|\lambda \in \Lambda$, the algorithm for the optimization proceeds as follows :

1. Initialize $\lambda_t = \lambda_0$
2. Generate a sample of size N $(x_i^t)_{1 \leq i \leq N}$ from $\pi(.,\lambda_t)$, compute $(\phi(x_i^t))_{1 \leq i \leq N}$ and order them from smallest to biggest. Estimate γ_t as the $(1-\rho)$ sample percentile.
3. update λ_t with :

$$\lambda_{t+1} = \arg\max_{\lambda} \frac{1}{N} \sum_{i=1}^{N} I\left[\phi(x_i^t) \geq \gamma_t\right] \ln \pi(x_i^t,\lambda)$$

4. repeat from step 2 until convergence.
5. assume convergence is reached at $t = t^*$, an optimal value for ϕ can be done by drawing from $\pi(.,\lambda_{t^*})$.

4.2. Application to the path planning task

In this part we deal with the application of the CE methods to our task. First of all, it is necessary to define the random mechanism $(\pi(.,\lambda)|\lambda \in \Lambda)$ to generate path examples. More precisely we want to generate sample which :

- start from s_i and end at s_f.
- respect the authorized matrix $\delta(a_k, a_{k+1}),\ \forall k$.
- have whole length less than T_{max}.

One way to achieve that is to use a probability matrix $\mathbf{P_{sa}} = (p_{sa})$ with $s \in \{1, ..., N_s\}$ and $a \in \{1, ..., N_a\}$(in our case $N_a = 8$).

$$\mathbf{P_{sa}} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{17} & p_{18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{N_s 1} & p_{N_s 2} & \cdots & p_{N_s 7} & p_{N_s 8} \end{pmatrix} \tag{13}$$

with $\forall s$, $P_s(.)$ is a discrete probability law such as :

$$P_s(a=i) = p_{si}, i = 1, \cdots, 8 \text{ with } \sum_{i=1}^{8} p_{si} = 1$$

So, to solve our problem we are concerned with the optimization of the $N_s \times N_a$ parameters (p_{sa}) using the C.E. algorithm. Let introduce :

- $A(\tilde{s}, \tilde{a})_{k-1}^{k} = \{a | s_k = \tilde{s},\ \delta(a_k = a, a_{k-1} = \tilde{a}) = 1\}$
- $\tilde{P}_s(.)$ such as,

$$\forall a \in A(\tilde{s}, \tilde{a})_{k-1}^{k}, \quad \tilde{p}_{\tilde{s}a} = \frac{p_{\tilde{s}a}}{\sum_{a \in A(\tilde{s}, \tilde{a})_{k-1}^{k}} p_{\tilde{s}a}}$$
$$else \qquad \tilde{p}_{\tilde{s}a} = 0;$$

$A(\tilde{s}, \tilde{a})_{k-1}^{k}$ is the admissible actions at time k in state $s_k = \tilde{s}$ knowing that $\tilde{a}$ was chosen at time $k-1$ and $\tilde{P}_{\tilde{s}}(.)$ is the normalized restriction of $P_{\tilde{s}}(.)$ to $A(\tilde{s}, \tilde{a})_{k-1}^{k}$. Paths can be generated as described in table 1.

Table 1. Path generation principle

$j = 0$
while $(j < N)$
  $k = 0, \quad set \quad s_0 = s_i$
  generate one action a_0^j according to
  $P_{s_0.}$ and apply it.
  set $k = k+1$ and $T = 1$
  until $s_k = s_f$ do
    - compute $A(\tilde{s}, \tilde{a})_{k-1}^{k}$
    if $A(\tilde{s}, \tilde{a})_{k-1}^{k} \neq \varnothing$
    - generate one action $a_k \in A(s_k, a_k)_{k-1}^{k}$
    according to $\tilde{\mathbf{P}}_{s_k.}$ and apply it.
    else stop and set $j = j$
    - set $k = k+1$ and $T = T+1$.
    if $T > T_{max}$ stop and set $j = j$
    else return
  $x(j) = (s_i, a_0^j, s_1^j, a_1^j, ..., s_{k-1}^j, a_{k-1}^j, s_f)$
  $j = j+1$

4.2.1. *Updating the* $\mathbf{P_{sa}}$ *matrix*

At step t of the C.E algorithm, given N admissible paths $(x(j))_{1 \leq j \leq N}$, we can evaluate each one by calculating the PCRB sequence and applying ϕ and then estimate the parameter γ_t. One can update $\mathbf{P}_{sa}$ by solving 3.
Let $x(j) = (s_i, a_0^j, s_1^j, a_1^j, ..., s_{k-1}^j, a_{k-1}^j, s_f)$ be a path, we have :

$$\ln \pi(x(j), \mathbf{P_{sa}}) = \sum_{i=0}^{k-1} I\left[\{x(j) \in \chi_{sa}\}\right] \ln p_{sa} \qquad (14)$$

where I is the indicator function and $\{x(j) \in \chi_{sa}\}$ means that the trajectory contains a visit to state s in which action a is taken. Since for each s, the row $P_s(.)$ is a discrete probability, 3 must be solved under the condition that the rows of $\mathbf{P_{sa}}$ sum up to 1. This yields to solve the following problem using Lagrange multipliers $(\mu_s)_{\{1 \leq s \leq N_s\}}$:

$$\max_{\mathbf{P_{sa}}} \min_{\mu_1, \cdots, \mu_{N_s}} \frac{1}{N} \sum_{j=1}^{N} I\left[\phi_k(x(j)) \geq \gamma_t\right] \ln \pi(x(j), \mathbf{P_{sa}}) \qquad (15)$$
$$+ \sum_{s=1}^{N_s} \left(\mu_s \sum_{a=1}^{a=8} (p_{sa} - 1) \right)$$

After differentiation with respect to each p_{sa} and applying $\sum_{a=1}^{8} p_{sa} = 1$, we can obtain μ_s and the final updating formula:

$$p_{sa} = \frac{\sum_{j=1}^{N} I\left[\{\phi_k(x(j)) \geq \gamma_t\}\right] \cdot I\left[\{x(j) \in \chi_{sa}\}\right]}{\sum_{j=1}^{N} I\left[\{\phi_k(x(j)) \geq \gamma_t\}\right] \cdot I\left[\{x(j) \in \chi_{s}\}\right]} \qquad (16)$$

where $\{x(j) \in \chi_s\}$ means that the trajectory contains a visit to state s. For the first iteration, $\forall s,\ P_s(.)$ is a uniform probability density function.

5. RESULTS

The algorithm has not been widely tested, and only one simple scenario is introduced in this paper. In this example, the map is defined on $[-2, 12] \times [-2, 12]$ with 6 m_j point features (figure 3). The state space is discretized in $N_s = 15 \times 15$ states. The initial and terminal states are respectively in positions $(0, 0)$ and $(10, 10)$. For the dynamic model noise process,we consider the same error on both axis $(\sigma^2 = 0.05)$.

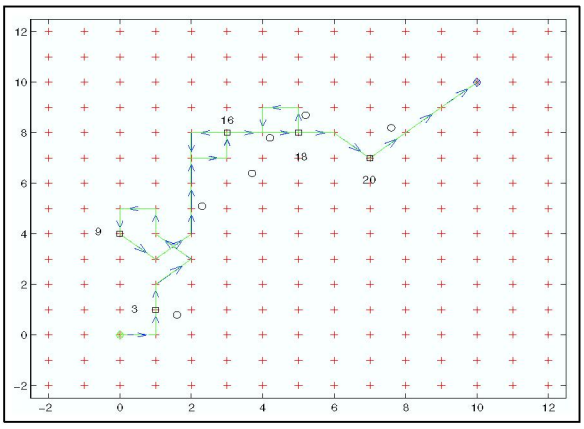

Fig. 3. The best solution after convergence of the CE. s_i and s_f states ($\diamond$) and map features ($\circ$).

The mobile can only apply $[-\frac{\pi}{2}; \frac{\pi}{2}]$ headings controls at each time and $T_{max} \leq 30$.
For the observation model, a landmark m_j is visible at time k provided that $z_r^k(j) \leq 2$ and $|z_\beta^k(j)| \leq 40$ deg. with the noise characteristics $\sigma_r = 1.5.10^{-3}$ (range) and $\sigma_\beta = 0.5$ deg. (bearing) for all features.

The computation of the PCRB matrices was performed with $N_c = 1000$ to estimate $J_k(Z)$. For the optimization step, the Cross Entropy algorithm was implemented with 1000 iterations, $N = 5000$ admissible paths and $\rho = 0.1$. That is to say, the 500 best samples are used for updating the p_{sa} probabilities. Figure 3 shows the optimal trajectory after convergence.

5.1. Analysis

As expected the mobile is guided toward the area with landmarks in order to improve its performance of localization. Moreover, it operates to keep the landmarks visible while the maneuvers (δ matrix) and the time constraints (T_{max}) allow it. We can also notice that the algorithm converges rapidly toward a solution. To illustrate that we present in the next figure, the evolution of parameters γ and the maximum value of ϕ at each iteration of the CE algorithm (figure 4).

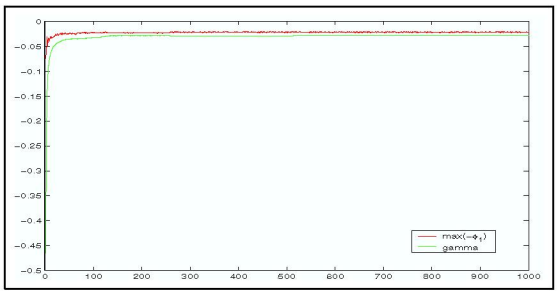

Fig. 4. Evolution of γ (solid line) and the minimum value of the functional (dashed line).

When we look at precisely after convergence the densities $(P_s(.))$ for all s in the optimal trajectory we can notice that some of them are not a dirac probability law. For instance, in position point 16 (see figure 3) the probability density function is bimodal indicating that the mobile hesitates between "go directly on the right side" and "make a cycle". Such behaviors are concentrated on states where maneuvers can be done to keep the landmarks visible as far as possible.

6. CONCLUSIONS AND PERSPECTIVES

In this paper, we presented a framework to solve a path planning task for a mobile. The problem was discretized and considered as a sequential decision process. Our main goal was to find the optimal trajectory according to a measure of capability of estimating accurately the state of the mobile during the execution. A functional of the PCRB was introduced as the criterion of performance. The main contribution of the paper is the use of the Cross Entropy algorithm to solve the optimization step as Dynamic Programming could not be applied. This approach was tested on a simple first example and seems to be relevant.
Future work will first concentrate on the complete implementation of the algorithm and application to more examples. We will also investigate a continuous approach. The tuning of the Cross-Entropy to our specific task was not studied, some experiments have to be carried out based on device given in [5].

7. REFERENCES

[1] S. Paris, J-P. Le Cadre *Planning for Terrain-Aided Navigation*, Fusion 2002, Annapolis (USA), pp 1007–1014, 7-11 Jul. 2002.

[2] J.-P. Le Cadre and O. Tremois , *The Matrix Dynamic Programming Property and its Implications.*. SIAM Journal on Matrix Analysis, 18 (2): pp 818-826, April 1997.

[3] R. Enns and D. Morrell, "Terrain-Aided Navigation Using the Viterbi Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 8, pp. 1444–1449, November-December 1995.

[4] P. Tichavsky, C.H. Muravchik, A. Nehorai Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering,IEEE Transactions on Signal Processing, Vol. 46, no. 5, pp. 1386–1396, May 1998.

[5] R. Rubinstein,D. P. Kroese *The Cross-Entropy method. An unified approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*, Information Science & Statistics, Springer 2004.

[6] de-Boer,P. Kroese, D. Mannor and R.Rubinstein *A tutorial on the cross-entropy method.*, 2003. http://www.cs.utwente.nl/ ptdeboer/ce/

[7] S. Mannor, R. Rubinstein, Y.Gat *The Cross Entropy method for Fast Policy Search*, Proc.of the 20^{th} I.C on Machine Learning, 2003 .

[8] R. S. Sutton and A.G.Barto, *Reinforcement Learning. An Introduction*. A Bradford book, 2000.

[9] H.L. Van Trees, *D*etection, Estimation and Modulation Theory, New York Wiley, 1968.