Evaluation of a robot learning and planning via Extreme Value Theory

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

J-P. LE CADRE
IRISA/CNRS
Campus de Beaulieu
34042 Rennes, France
Email: lecadre@irisa.fr

***Abstract*—This paper presents a methodology for the evaluation of a path planning algorithm based on a learning approach. Here this evaluation procedure is applied for the problem of optimizing the navigation of a mobile robot in a known environment. A metric map composed of landmarks representing natural elements is given to define the best trajectory which permits to guarantee a localization performance during its execution. The vehicle is equipped with a sensor which enables it to obtain range and bearing measurements from landmarks. These measurements are matched with the map to estimate its position. As the mobile state and the measurements are stochastic, the optimal planning scheme considered in this paper deals with Posterior Cramér-Rao Bound as a performance measure. Because of the nature of the cost function, classical optimization algorithms like Dynamic Programming are irrelevant. Therefore, we propose to achieve the optimization step with the Cross Entropy algorithm for optimization to generate trajectories from a suitable parameterized probability density functions family. Nevertheless, although the convergence of this algorithm can be assessed with the analysis of the stationarity of its intrinsic parameters, we are not able to quantify the level of convergence around the optimal value. As a consequence, an external investigation can be applied from an alternative stochastic procedure followed by an analysis via extreme value theory.**

Keywords: planning, Cross Entropy, estimation, extreme value analysis.

I. INTRODUCTION

This paper is in the continuation of the work introduced in [1] dealing with the optimal path planning problem for an autonomous agent. The considered agent is a mobile robot system navigating in an environment with a provided metric map. During navigation, the accurate determination of its position and orientation with respect to a fixed reference frame are of great importance. This global localization is very often performed with noisy measurements obtained from exteroceptive sensors. Several robot localization approaches were proposed in the literature, depending on the nature of the sensors (vision, laser, GPS...) and also on the noise models. Most of them treated the uncertainties as random phenomena and employed *Bayesian estimation* [10]. Others considered errors as unknown but bounded and derived the localization task as a *set-membership estimation* issue [12]. In our case, only Bayesian localization techniques are considered, but the different results introduced below can be applied to set-membership localization. The main aim is to compute the posterior density of the current robot pose conditioned on the accumulated measurements. Depending on the assumptions on the robot dynamic and observation models, different algorithms can be used. Very often, when the systems are linear or nearly linear with errors described by Gaussian probability density function, *Kalman filtering* (KF) or *Extended Kalman filtering* (EKF) [10] are valuable tools. For situations with severe nonlinearity and multi-modal density, more suitable methods such as *particle filtering* [10] can be used. For our application, the robot collects measurements of its surround environment and compares them with its known map to increase its predicted state.

In Bayesian filtering, the main criterion for analyzing the performance of a given filter is the Posterior Cramér-Rao bound (PCRB). In [1], we used a function of this criterion to find optimal trajectories based on the Cross-Entropy algorithm for combinatorial optimization. The developed framework seemed to give good results of convergence on first scenarios. A difficulty with the Cross Entropy approach is to determine whether the algorithm really converged toward an optimal solution. In this paper, we try to study the convergence of the planning design using an auxiliary random procedure based on Markov Decision Process and extreme value theory [15]. This external procedure enables us to obtain an approximation of the true density of the objective functions over the admissible paths (or configuration) space. The goal of extreme value theory is to approximate the tail of this density in order to estimate the endpoint which is the relevant statistic in the context of stochastic optimization.

The paper is organized as follows. In section II we introduce the robot dynamic model. The next section deals with the localization algorithm and the Posterior Cramér-Rao Bound. We make also its derivation for our problem and define the objective function for the planning task. The admissible path space for the planning problem is detailed in section IV. Then we present the path algorithm based on the Cross-entropy algorithm before an introduction of the extreme value theory for the evaluation. Finally, we illustrate the reasoning with an example and conclude.

II. THE ROBOT DYNAMIC MODEL

In this section the dynamic model used for the robot localization is introduced.

A. Motion model

The state of the mobile robot is composed of its 2D Cartesian position (r^x, r^y) and its orientation θ with respect to a fixed reference frame $\mathcal{R}_0$ (see figure 1). At time t_k, the state vector is

$$p_k = [r_k^x \, r_k^y \, \theta_k]^T . \quad (1)$$

The state evolution is governed by the following nonlinear controlled dynamic system

$$p_{k+1} = f(p_k, u_k, e_k) = \begin{pmatrix} r_k^x + v_k d_k \cos(\theta_k + \delta_k) \\ r_k^y + v_k d_k \sin(\theta_k + \delta_k) \\ \theta_k + \delta_k \end{pmatrix} + e_k \quad (2)$$

where $u_k = [v_k \, \delta_k]^T$ is the control input composed of the velocity and the angular variation of the mobile at time t_k. $d_k = t_{k+1} - t_k$ is the laps time between two consecutive times. The process vector noise e_k is supposed independent and describes the uncertainty in the dynamic model. Here, e_k is a Gaussian process with zero mean and Q_k covariance matrix ($e_k \sim \mathcal{N}(0, Q_k)$). Moreover, we suppose that the initial position error is also Gaussian with covariance matrix Q_0.

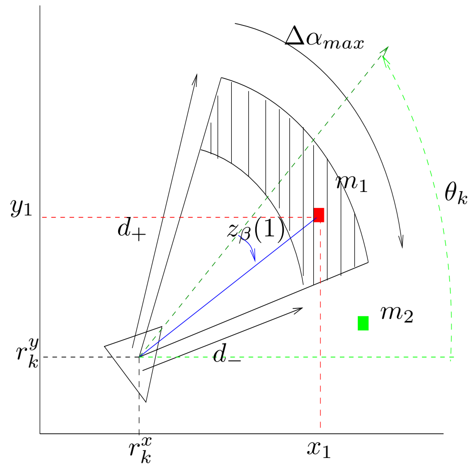

Figure 1. sensor model. A visible (red) and non visible (green) landmark.

B. Map Description and observation model

The a priori description of the environment provided to the robot is a metric map. It is composed of a set of spatially l located landmarks $(m_i)_{1 \leq i \leq l}$ with respective position $(x_i, y_i) \in \mathbf{R}^2$. This map $\mathcal{M}$ represents particular features in the environment extracted during a previous mapping stage. Therefore, this map may suffer from some uncertainties. In order to operate with such a map, the robot needs to use feature extraction algorithms, supposing that structures corresponding to the landmarks are present in the environment. Obviously, the extraction procedure varies according to the sensor used to collect measurements. For example, when vision systems are employed Harris or SIFT image operators can be used. At time t_k, due to the sensor local field of view, a few landmarks can be observed. We do not consider data association and non detection problem for the moment. Let n_k be the number of visible landmarks with indexes $I_v(p_k)$, the measurement is a stacked vector y_k :

$$y_k = [y_{i_1}^k \; y_{i_2}^k \; \cdots \; y_{i_{n_k}}^k]^T \quad (3)$$

The observation model is given by

$$y_k = \mathbf{h_k}(p_k, \mathcal{M}) + \mathbf{W_k} \quad (4)$$

where the $2 \times j$-th $2 \times j + 1$-th elements of $\mathbf{h_k}(p_k, \mathcal{M})$ are respectively the components

$$\sqrt{(r_k^x - x_{i_j})^2 + (r_k^y - y_{i_j})^2} \quad (5)$$

$$\arctan(\frac{y_{i_j} - r_k^y}{x_{i_j} - r_k^x}) - \theta_k \quad (6)$$

corresponding to the range and relative bearings of the landmarks from the robot pose. $\mathbf{W_k}$ represents the stacked vector of range and bearing observed noise components. Its covariance matrix $\mathbf{R_k}$ of size $(2n_k \times 2n_k)$ is block diagonal with the j^{th} block equal to the covariance matrix $C_k = diag[\sigma_{r,i_j}^2, \sigma_{\beta,i_j}^2]$ of each individual measurement.

III. Bayesian estimation and The Posterior Cramér-Rao Bound

Bayesian estimation and in particular Bayes filters address the problem of estimating the state of a dynamic system from sensor measurements. The aim is to estimate the posterior distribution over the state-space conditioned on these observations. The main assumption is the Markovian properties, that is the current measurement are conditionally independent of past ones knowing the state.

A. recursive robot Localization

For the mobile robot with dynamic system specified by equations 2 and 4, Bayes filters try to construct the probability density function (pdf.)

$$p(p_{1:k} | y_{1:k}, u_{1:k-1}). \quad (7)$$

This can be done recursively using Bayes rules considering the two following expressions :

$$p(p_{k+1} | y_{1:k+1}, u_{1:k}) = \int p(p_{k+1} | p_k, u_k) p(p_k | y_{1:k}, u_{1:k-1}) dp_k \quad (8)$$

$$p(p_k | y_{1:k}, u_{1:k-1}) = \frac{p(y_k | p_k) p(p_k | y_{1:k-1}, u_{1:k-2})}{p(y_k | y_{1:k-1})} \quad (9)$$

Depending on the properties of the system dynamic the appropriate Bayes filter can be used to approximate these density. For instance, the robust Monte Carlo Localization algorithm [11] is a particle filter formulation adapted for mobile robots.

B. The Posterior Cramér-Rao Bound

The main objective of the planning task introduced in the next section is to design paths with good localization performance. So, fundamental limits of estimation performance depending on the system configuration must be defined. The Posterior Cramér-Rao Bound is such a measure.

1) Definition: The Posterior Cramér-Rao Bound is a lower bound for the second order moment for unbiased estimator $\hat{x}$ of a random vector x from random measurements y. It is based on the derivatives of the logarithm of the joint probability density function of both random processes. For the filtering case, if $\hat{p}_k$ is an estimate of p_k derived from the measurements $y_{1:k}$ given the control input sequence $u_{1:k-1}$, the PCRB can be calculated recursively, [5]

$$P_{k+1}^{-1} = D_k^{22} - D_k^{21}(P_k^{-1} + D_k^{11})^{-1}D_k^{12} \tag{10}$$

where

$$\begin{aligned}
D_k^{11} &= E\{-\Delta_{p_k}^{p_k} \log\left(p(p_{k+1}|p_k)\right)\} \ , & (11)\\
D_k^{12} &= E\{-\Delta_{p_k}^{p_{k+1}} \log\left(p(p_{k+1}|p_k)\right)\} \ , & (12)\\
D_k^{21} &= [D_k^{12}]^T \ , & (13)\\
D_k^{22} &= E\{-\Delta_{p_{k+1}}^{p_{k+1}} \log\left(p(p_{k+1}|p_k)\right\} + & (14)\\
& \quad E\{-\triangle_{p_{k+1}}^{p_{k+1}} \log\left(p(y_{k+1}|p_{k+1}) \ \right\}. & (15)
\end{aligned}$$

where P_k verified [1] :

$$Cov(p_k - \hat{p}_k) = \mathbf{E}\left[(p_k - \hat{p}_k)(p_k - \hat{p}_k)^T\right] \succeq P_k \tag{16}$$

The densities $p\left(p_{k+1}|p_k\right)$ and $p(y_{k+1}|p_{k+1})$ are respectively computed from the dynamic and observation models. Those relations are valid under the assumptions that the mentioned second-order derivatives, expectations and matrix inverses exist.

2) Application to our system: Given the trajectory for state vector the PCRB history can be calculated as below. For the robot system defined by equations 2 and 4, we need to perform simulations to approximate the PCRB because some expected expressions in 11 are analytically unsolvable due to the nonlinearities of the dynamic systems equations. More precisely, we have

$$\begin{aligned}
D_k^{11} &= E_{p_k}\left\{\left[\nabla_{p_k} f^T(p_k)\right] Q_k^{-1} \left[\nabla_{p_k} f^T(p_k)\right]^T\right\}, & (17)\\
D_k^{12} &= -E_{p_k}\left\{\left[\nabla_{p_k} f^T(p_k)\right]\right\} Q_k^{-1}, & (18)\\
D_k^{22} &= Q_k^{-1} + J_{k+1}^y, & (19)\\
J_k^y &= E_{p_k}\{\sum_{j \in I_v(p_k)} H(p_k, j)^T C_k^{-1} H(p_k, j)\} \ . & (20)
\end{aligned}$$

where

$$\nabla_{p_k} f^T(p_k) = \begin{pmatrix} 1 & 0 & -v_k \sin(\theta_k + \delta_k) d_k \\ 0 & 1 & v_k \cos(\theta_k + \delta_k) d_k \\ 0 & 0 & 1 \end{pmatrix} \tag{21}$$

and

$$H(p_k, j) = \begin{pmatrix} \frac{(r_k^x - x_j)}{\sqrt{d_k^j}} & \frac{(r_k^y - y_j)}{\sqrt{d_k^j}} & 0 \\ \frac{(r_k^y - y_j)}{d_k^j} & \frac{-(r_k^x - x_j)}{d_k^j} & -1 \end{pmatrix} \tag{22}$$

where $d_k^j = (r_k^x - x_j)^2 + (r_k^y - y_j)^2$. Given the input sequence $u_{1:K-1}$ $K > 1$, The approximations can then be obtained from crude Monte Carlo simulation. Let $\left\{p_{1:K}^i\right\}_{1 \leq i \leq N}$ be N noisy realizations of the same trajectory, we get $\forall k \in \{1; \cdots ; K\}$:

$$\begin{aligned}
D_k^{11} &\approx \frac{1}{N} \sum_i \left[\nabla_{p_k} f^T(p_k^i)\right] Q_k^{-1} \left[\nabla_{p_k} f^T(p_k^i)\right]^T & (23)\\
D_k^{12} &\approx \frac{1}{N} \sum_i \left[\nabla_{p_k} f^T(p_k^i)\right] Q_k^{-1} & (24)\\
J_k^y &\approx \frac{1}{N} \sum_i \sum_{I_v(p_k^i)} H(p_k^i, j)^T C_k^{-1} H(p_k^i, j) & (25)
\end{aligned}$$

The recursion 10 can be initialized with $P_0 = Q_0$.

3) Criterion for path planning: Once the PCRB history $P_{0:K} = \{P_0, \cdots, P_K\}$ computed, we can deduce a criterion for the robot localization performance along the predefined path. Different functionals can be used. In this work, we considered the determinant of the PCRB matrix which is linked with the volume of the "smaller" error ellipse that can be reached. We denote $\phi(u_{1:K-1})$ this measure :

$$\phi(u_{1:K-1}) = -\sum_{k=1}^{K} w_k \ \det(\mathcal{B}^T P_k(u_{1:K-1}) \mathcal{B}) \tag{26}$$

where w_k are weighting factors and $\mathcal{B}$ the projection matrix of the state on the position subspace.
It is important to note that the computation of this criterion depends on all the past of the trajectories. Nevertheless, we can cite the works in [7] where the authors try to get rid of the simulation stage by making local approximations.

IV. The path planning formulation

In this section, we introduced the framework used to tackle the path planning problem. The admissible paths or configuration space is introduced, followed by the implementation of the Cross Entropy algorithm for the optimization.

A. Configuration space

We formalize the problem as a sequence of decisions problem [4]. We suppose that the path is a sequence of displacements with constant velocity and constant heading. First of all, a grid with N_s points is defined over the map $\mathcal{M}$. For each point $(x(i,j), y(i,j))$ on that grid, we can associate one index $s = g(i,j) \in \mathcal{S} = \{1, \cdots, N_s\}$, $\forall (i,j) \in \{1, \cdots, N_x\} \times \{1, \cdots, N_y\}$ where g is an obvious mapping. We can then introduce an auxiliary state s which represents the position index in $\mathcal{S} = \{1, \cdots, N_s\}$ of the associated point on the grid. An admissible path is composed of sequence of points of that grid. The mobile orientation is also supposed to be restricted to eight directions which correspond to eight actions or decisions $a \in \mathcal{A} = \{1, \cdots, N_a\}$ (with $N_a = 8$). Therefore from each point of the grid, there are at most 8 reachable neighbors. The actions are clockwise numbered from 1 (”go up and right”) to 8 (”go up”). So a given trajectory defined by $p_{1:k}$ is equivalent to the associated sequence $s_{1:k} = \{s_1, \cdots, s_k\}$ and $a_{1:k-1} = \{a_1, \cdots, a_{k-1}\}$. In that way, a graph structure $\mathcal{G}(\mathcal{S}, \mathcal{A})$ can be used to describe our problem, where the vertices and edges are respectively the states s and the actions

[1] $A \succeq B$ means A-B is positive semi-definite

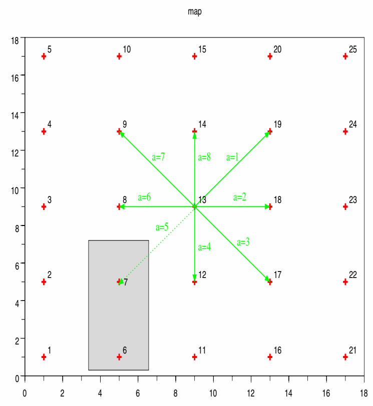

Figure 2. Grid over the map and actions in state 13.

a. let q_i and q_f (or more generally the set X_f) be the starting and goal positions (the goal set). They are supposed to be on the grid otherwise the nearest positions on the grid are used. Figure 2 shows one example with 25 states. We can notice that in state 13 only 7 actions guarantee to stay in the free space, action 5 is not admissible. Moreover constraints on the heading control are also taken into account to prevent from particular motion behavior such that "bang-bang" effect. We defined an *authorized transition matrix* $\delta(.,.)$ [2] which indicates the admissible actions at time k with respect to those selected at time $k-1$. For example, for $\frac{\pi}{4}$ bounded heading controls, such a matrix is given by the following expression :

$$\delta(a_k, a_{k-1}) = \left(\begin{array}{c|cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right) \quad (27)$$

The admissible paths space is now detailed. It consists in multileg paths starting in q_i and ending in q_f with orientation constraints between two consecutive times. Given such a path, the PCRB sequence and the performance criterion can be computed as explained in section III. Over this configuration space we are looking for the path with the best $\phi()$ value. To make the optimization, classical techniques like Dynamic Programming based on the Markov Decision Process framework cannot be directly applied as our functional does not satisfy the Matrix Dynamic Programming properties as derived in [3]. We investigate a learning based approach based on the Cross-Entropy method [1] to solve the problem.

B. The cross-entropy algorithm

The Cross Entropy method (CE) initially used for estimating the probability of *rare events*, was further adapted to solve combinatorial optimization problem such that TSP [6]. We quickly remind its main principle and its application to our task but for more details the reader may refer to [1], [6].

1) Principles: Consider the following optimization problem:

$$\phi(x^*) = \gamma^* = \max_{x \in \mathcal{X}} \phi(x) \quad (28)$$

The principle of the CE for optimization is to translate the problem 28 into an *associated stochastic problem* and then solved it adaptively as the simulation of a rare event. If $\gamma*$ is the optimum of ϕ and x random, $F_{\gamma^*} = \{x \in \mathcal{X} | \phi(x) \geq \gamma^*\}$ is generally a rare event. The main idea is to learn iteratively a probability density function in a suitable parameterized family $\pi(., \lambda) | \lambda \in \Lambda$ in order to draw samples around the optimum. The learning stage consists in solving an optimization problem which goal is to minimize the *Kullback-Leibler* "pseudo-distance" to improve the performance simulation in the tail of the underlying density. 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 the problem globally.
Given a selection rate $\rho \in [0, 1[$, 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 :

$$\lambda_{t+1} = \arg\max_{\lambda} \frac{1}{N} \sum_{i=1}^{N} I\left[\phi(x_i^t) \geq \gamma_t\right] \ln \pi(x_i^t, \lambda) \quad (29)$$

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 $\pi(., \lambda_{t^*})$.

This is the main version of the CE algorithm, but in practice the update stage (3) includes a smoothing procedure. If $\tilde{\lambda}_{t+1}$ is the solution of the optimization problem 3 and $0 \leq \nu \leq 1$,

$$\lambda_{t+1} = \nu \tilde{\lambda}_{t+1} + (1 - \nu)\lambda_t \quad (30)$$

2) Application to our task: To apply the CE to our planning problem, we need to define a family of pdfs to generate admissible trajectories. in [1], we considered a family of probability matrix $\mathbf{P_{sa}} = (p_{sa})$ with $s \in \{1, ..., N_s\}$ and $a \in \{1, ..., N_a\}$(in our case $N_a = 8$).

$$\mathbf{P_{sa}} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{17} & p_{18} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{N_s1} & p_{N_s2} & \cdots & p_{N_s7} & p_{N_s8} \end{pmatrix} \quad (31)$$

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

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

Some elements of this matrix can be always equal to zero to take into account the map configuration (border, forbidden area, obstacles...). We showed [1] that it can be used to generate admissible paths using an acception-rejection algorithm. To limit the rejection rate due to the final point constraint, we since modified the algorithm by adapted the objective function considering a final cost. Indeed, Let τ an admissible trajectory with length K a cost associated with the final state can easily be introduced to penalize it when the final state s_K is far from the planning goal q_f :

$$\phi_f(\tau, s_K, q_f) = \begin{cases} -\mu_f \; f(dist(q_K, q_f)) & \text{if } s_K \in E(q_f) \\ -\infty & \text{else} \end{cases} \tag{32}$$

where $E(q_f)$ is a subset around q_f, a constant $\mu_f > 0$ and f is a function which increases with the distance between the final state s_K of τ and q_f. For instance, a quadratic function may be used. From now on, the notation ϕ will be used instead of $\phi + \phi_f$.

Given N $\{\tau_j\}_{j=1}^N$ samples of path with associated cost $\{\phi(\tau_j)\}_{j=1}^N$ at iteration t, the update stage for the elements of $\mathbf{P_{sa}}$ is given by the following expression [1]:

$$p_{sa} = \frac{\sum_{j=1}^N I\left[\{\phi(\tau_j) \leq \gamma_t\}\right] \cdot I\left[\{\tau_j \in \chi_{sa}\}\right]}{\sum_{j=1}^N I\left[\{\phi(\tau_j) \leq \gamma_t\}\right] \cdot I\left[\{\tau_j \in \chi_s\}\right]} \tag{33}$$

where $\{\tau_j \in \chi_s\}$ means that the trajectory contains a visit to state s and $\{\tau_j \in \chi_{sa}\}$ means that the trajectory contains a visit to state s where action a is chosen. For the first iteration, $\forall s,\ P_s(.)$ is taken as a uniform probability density function.

V. Extreme Value Theory

The convergence of the CE algorithm only relies on the stationarity of the observed quantile γ. In order to measure the performance of the result found by the CE, we consider an auxiliary random mechanism to analyze the distribution of the cost function ϕ over the space of admissible paths. For optimization, it is the tail of this distribution which is relevant. We exploited extreme value theory (EVT) results to perform such an analysis.

A. A short introduction

Let $x_1, x_2, \cdots, x_N$ be a sample of a random variable X drawn from an unknown continuous Cumulative Density Function (CDF) F having finite and unknown right endpoint $\omega(F)$. The EVT tries to model the behavior of the random process mainly outside the range of these available observations in the (right) tail of F [15]. In EVT there are two main ways of making the extreme values analysis. The first one deals with the modeling of the law of maxima of subsamples ("blocks"), while the second one is concerned with the distribution of data over of a specified threshold ("exceedances"). We make a short introduction of the main materials of the last one.

Let u be a certain threshold and F_u the distribution of exceedances, i.e. the values of x above u:

$$\begin{aligned} F_u(y) &\triangleq Pr(X - u \leq y | X > u), \\ &= \frac{F(u+y) - F(u)}{1 - F(u)} \, 0 \leq y \leq \omega(F) - u \end{aligned} \tag{34}$$

We are mainly interested in the estimation of F_u when u become higher and higher. A great part of the main results of the EVT is based on the following fundamental theorem from *Pickands*

Theorem 1: For a large class of underlying distributions F the exceeding distribution F_u, for u enough large, is well approximated by

$$F_u(y) \approx G_{\xi,\sigma}(y),\ u \rightarrow \omega(F)$$

where $G_{\xi,\sigma}$ is the so-called Generalized Pareto Distribution (GPD) defined as:

$$G_{\xi,\sigma}(y) = \begin{cases} 1 - (1 + \frac{\xi}{\sigma} y)^{-\frac{1}{\xi}} & \text{if } \xi \neq 0 \\ 1 - e^{-y/\sigma} & \text{if } \xi = 0 \end{cases} \tag{35}$$

At this point, this theorem indicates that, under some particular conditions on F, the distribution of exceeding observations can be approximated by a parametric model with two parameters. Inference can then be made from this model to make an analysis of the behavior of F around its maximum endpoint. Hence the p-th quantile x_p, more often called the Value-at-Risk (VaR_p) can be evaluated. It corresponds to the value of x which is exceeded with probability p:

$$1 - p = F(x_p) \tag{36}$$

The smaller p is, the higher is x_p. This result can be used in stochastic optimization where we are interested in knowing how far we are from the optimum value of our objective function. Indeed, the stochastic optimization process can be seen as a process to generate outcomes of the objective function. By this way, it is an estimator of the optimal value. The main problem is to assess the performance of this random mechanism by describing empirical distribution of the outcomes. Moreover, if the optimum cost is approached as the number of simulations increases, the endpoint of this distribution will be the optimum value. The main aim of the extreme values analysis is to try to estimate this endpoint as x_p. The computation of this estimate, or better its confidence interval is then of huge interest.

An analytical expression of x_p [14] can be derived from equations 34 and 35

$$x_p = u + \frac{\sigma}{\xi}\left[\left(\frac{N}{N_u} p\right)^{-\xi} - 1\right] \tag{37}$$

where N_u is the number of observations above the threshold u. The GPD parameters $\zeta \triangleq (\xi, \sigma)$ can be estimated from Maximum Likelihood method. Let $(x_l)_{l \in L}$ be the observations above the threshold, we have to maximize the log-likelihood function assuming independent observations:

$$L(\xi,\sigma)=\begin{cases}-N_u\log\sigma-(\frac{1}{\xi}+1)\sum_l\log\left[1+\frac{\xi}{\sigma}\left(x_l-u\right)\right]\\ -N_u\log\sigma-\frac{1}{\sigma}\sum_l\left(x_l-u\right)\end{cases} \tag{38}$$

This nonlinear maximization can be made using classical numerical optimization technique such as the Nelder-Mead search algorithm. The result of the estimation can be analyzed with graphical data exploration tools. For example, with a QQ-plot, we can visually check whether the samples satisfy the GPD distribution assumptions. Instead of making a simple point estimation for parameter ζ, it may be better to get an interval estimator. If we suppose that the estimate $\hat{\zeta}$ follows (in limit) a normal distribution, the variance-covariance matrix is given by the inverse of the Information matrix

$$V(\hat{\zeta})\triangleq(I(\hat{\zeta}))^{-1}=(-H(\hat{\zeta}))^{-1}$$

where H is the Hessian matrix of the log-likelihood functional in $L(\zeta)$ (see equation (38)). Then, a $1-\alpha$ asymptotic confidence interval for each component $\hat{\zeta}_i$ $(i\in 1,2)$ is given by

$$CI(\hat{\zeta}_i)=\left[\hat{\zeta}_i-z_{1-\alpha/2}\sqrt{V(\hat{\zeta}_i)},\hat{\zeta}_i+z_{1-\alpha/2}\sqrt{V(\hat{\zeta}_i)}\right] \tag{39}$$

where $z_{1-\alpha/2}$ is the $1-\alpha$ quantile of the $\mathcal{N}(0,1)$ normal distribution.

One estimate of the $p-th$ quantile can be deduced from $(\hat{\xi},\hat{\sigma})$ and equation (37)

$$\hat{x}_p\triangleq g(\zeta)=u+\frac{\hat{\sigma}}{\hat{\xi}}\left[\left(\frac{N}{N_u}p\right)^{-\hat{\xi}}-1\right]$$

A confidence interval can also be calculated for $\hat{x}_p$ using the "delta" method [8] x_p is a function of ζ

$$CI(\hat{x}_p)=\left[\hat{x}_p-z_{1-\alpha/2}\sqrt{dg},\hat{x}_p+z_{1-\alpha/2}\sqrt{dg}\right] \tag{40}$$

where $dg=\nabla_\zeta g(\hat{\zeta})^T V(\hat{\zeta})\nabla_\zeta g(\hat{\zeta})$. Nevertheless, in practice the log-likelihood functional is highly asymmetric and the delta method is inappropriate, therefore it is worth to use *"profile likelihood"* method [13], [14] to provide more accurate confidence intervals. For parameters ξ and σ they can be directly constructed from $L(\xi,\sigma)$. For example, the $1-\alpha$ confidence interval for ξ is given by values of ξ satisfying:

$$CI(\xi)=\left\{\xi|-2\left[\max_{\sigma}L(\xi,\sigma)-L(\hat{\xi},\hat{\sigma})\right]\leq\chi^2_{1,\alpha}\right\}$$

where $\chi^2_{1,\alpha}$ is the $1-\alpha$ quantile of the χ^2 distribution with one degree of freedom. For $\hat{x}_p$ the GPD density (equation (35)) must be reparameterized as a function of ξ and x_p. To do that equation (37) is used to express σ as a function of ξ and x_p, then the σ parameter is replaced in the GPD definition (see equation (41))

$$G_{\xi,x_p}(y)=\begin{cases}1-\left(1+\frac{\left[\left(\frac{N}{N_u}p\right)^{-\xi}-1\right]}{x_p-u}y\right)^{-\frac{1}{\xi}} & \text{if } \xi\neq 0\\ 1-e^{-\frac{y}{x_p-u}} & \text{if } \xi=0\end{cases} \tag{41}$$

This new expression of the log-likelihood function can be deduced and exploited to compute a $1-\alpha$ confidence interval $CI(\hat{x}_p)\triangleq[\hat{x}_p-\delta x_p^-,\hat{x}_p+\delta x_p^+]$ based on profile likelihood for $\hat{x}_p$.

We can notice that the estimation of the parameters of the extreme values distribution and of course of the estimate of the high quantile depends on the choice of the threshold u. There is no main rules to choose that parameter, but obviously the number of data above u decreases when u grows up which means that the estimation procedure becomes more and more noisy. Graphical tools can also be used to determine suitable thresholds values. In practice we determine the threshold value by taking 2% of the sample size of observated data.

B. Application to optimization

Consider a random mechanism from which we can generate N samples of trajectories $\{\tau_j\}_{j=1}^N$ with associated cost $\{\phi(\tau_j)\}_{j=1}^N$. From this observed data, we can make an extreme value analysis as explained above. We have to note that the true density of $\phi(\tau)$ is very complex and depends on many parameters (the robot dynamic systems, the paths characteristics, the sensor ability and the map properties (noise, correlation...)). We remind that the planning problem cannot be solved with classical Dynamic Programming techniques. Nevertheless, such an approach can be useful to make the generation of sample paths within a constrained Markov Decision process framework [2]. More precisely, one admissible path τ_j can be obtained as follows:

1. allocate an auxiliary random cost series $\Upsilon_j=\left(c^j_{s's}(a)\right)$ where $c^j_{s's}(a)\sim\mathcal{U}_{[0,1]}$, $\forall(s,s',a)\in\mathcal{S}\times\mathcal{S}\times\mathcal{A}$ where $\mathcal{U}_{[0,1]}$ is the uniform density on $[0,1]$.
2. then compute the optimal path τ_j according to the constrained dynamic programming algorithm [2].
3. and finally computing the approximated PCRB sequence along τ_j and get $\phi(\tau_j)$.

With those samples the right tail of the distribution of $\phi(\tau)$ can be approximated with a GPD distribution. Then, high quantile estimator x_p with $1-\alpha$ related accurate confidence intervals can be calculated. From these best trajectories, we can determine:

- how far we are from the optimal solution and what would be the difference of information gain between the best observed trajectory and the optimal trajectory (for a given risk level p)?
- How many simulations would be necessary to achieve this given level of performance?

Indeed, we can approximate the average number of observed $\hat{n}_p^\alpha$ trajectories which can be associated with the performance $\hat{x}_p$ are those which lie in the confidence interval:

$$\begin{aligned}\hat{n}_p^\alpha &\triangleq Pr\left(x\in CI^\alpha(\hat{x}_p)\right)N_u \\ &= \left(F_u\left(\hat{x}_p+\delta x_p^+\right)-F_u\left(\hat{x}_p-\delta x_p^-\right)\right)N_u\end{aligned} \tag{42}$$

At that point, we can now compare the result of the convergence of the optimal solution given by the planning algorithm based on the CE with the given level of performance $\hat{x}_p$ for risk level p. The solution of the planning will be considered as satisfactory if it is better than $\hat{x}_p + \delta x_p^+$.

VI. Results

In this section we consider one scenario to illustrate our procedure of evaluation. The map is defined on the set $[-4, 54] \times [-4, 54]$ and the grid resolution is $d_x = d_y = 4$, so the dimension of the discretized state space $\mathcal{S}$ is $N_s = 225$. There are only four landmarks whose positions are given in table I. Three of them are located in the left and up side of the map and the last one is lower and on the right. For the mobile dynamic system and the observation model, the covariance matrix of the noises model are

	m_0	m_1	m_2	m_3
x	7.2	7.8	22.8	43.1
y	20.4	35.2	42.8	25.8

Table I
THE LANDMARKS OF THE MAP

$$P_0 = Q_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}, \quad C_k = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix} \forall k \text{ and } \forall m_j.$$

The variances on the robot orientation and on the bearings measurements are expressed in degree here for clarity but have to be converted in radian. The constant velocity at each step is $4\,m.s^{-1}$ and the robot can only apply $\{-\frac{\pi}{4}, 0, \frac{\pi}{4}\}$ controls between two consecutive times, then the δ matrix is the same as in section IV. For the observation model, a landmark m_j is visible at time k if its relative distance and bearing verify:

$$\begin{cases} r_- \leq y_r^k(j) \leq r_+ \\ |y_\beta^k(j)| \leq \theta_{max} \end{cases}$$

where $r_- = 0.01$ m., $r_+ = 12$ m. (3 times the grid resolution) and $\theta_{max} = 90$ deg..

For the planning, we consider trajectories with at most $K = 30$ elementary moves and the initial and final positions are respectively $q_i = (6; 2; 45)$ and $q_f = (46; 46)$. The PCRB estimate is approximated from Monte Carlo simulation with $N_{mc} = 800$ noisy realizations of the trajectories. For the extreme values analysis, we generate 40000 paths and evaluate the cost function ϕ for each one. We considered 2% of the sample to make inference and estimate the parameters $(\hat{\xi}, \hat{\sigma})$ of the GPD distribution, that is to say the 800 best plans are considered. Let $\{\phi^j\}, j \in \{1, \cdots, 40000\}$ be the associated costs of the paths. We first normalize to study the new random variable

$$c^j = \frac{\phi^j - m[\phi]}{\sigma[\phi]}$$

Figure 3 shows the best path drawn with the procedure based on Dynamic Programming. The associated cost is $c^{max} \approx 0.8815$. The threshold corresponding to 2% is $u = 0.8479$. The estimation of the quantile x_p and its related confidence interval can be calculated. For $p = 2.10^{-5}$, we found $\hat{x}_p = 0.8804$ and the 95% confidence interval given by the profile likelihood and delta methods are respectively $[0.879; 0.8829]$ and $[0.8758; 0.8849]$. The values of $1 - F_u(x)$ are represented on figure 4 with $\hat{x}_p$ and the both confidence intervals. We can notice that the confidence interval based on the "delta" method is larger and is less accurate than the result based on the profile likelihood. We then derived $\hat{n}_p^\alpha \approx 2.0325$ which is coherent with the number 2 of observed trajectories with costs in the confidence interval (see figure 4).

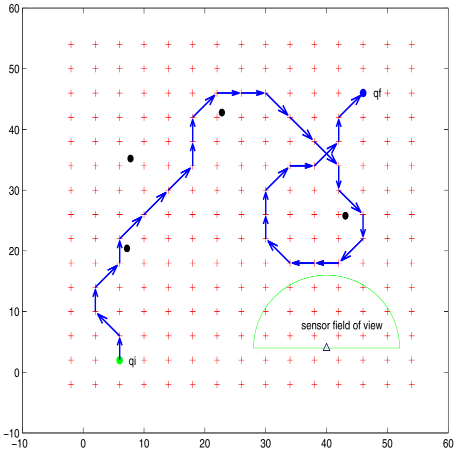

Figure 3. Best drawn trajectory from Dynamic Programming (40000 samples).

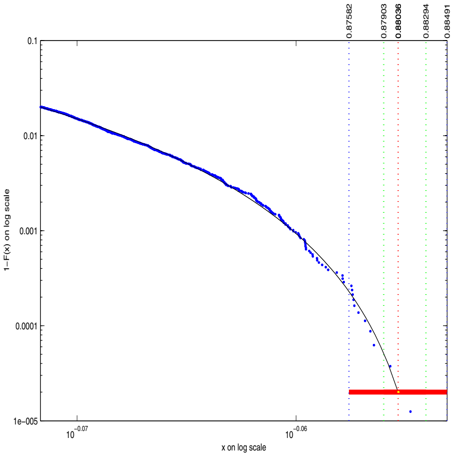

Figure 4. Continuous curve of $1 - G_{\hat{\xi},\hat{\sigma}}$ fitting the 800 extreme values (dots). $\hat{x}_p$ (red) with its 95% intervals of confidence based on Profile Likelihood (green) and the "delta" method (blue).

The optimal planning using the CE algorithm is made on the same experiment with 4000 trajectories samples at each iteration of the algorithm. The selection rate was $\rho = 0.1$, therefore the 400 best paths contributed to the update stage of the $\mathbf{P_{sa}}$ matrix. The smoothing parameter was $\nu = 0.4$. Figure 5 shows the evolution of estimated quantile γ and c^{max} the best samples at each iteration for the performance function c. The best cost obtained with the MDP approach is also displayed. We can notice that the algorithm converges rapidly

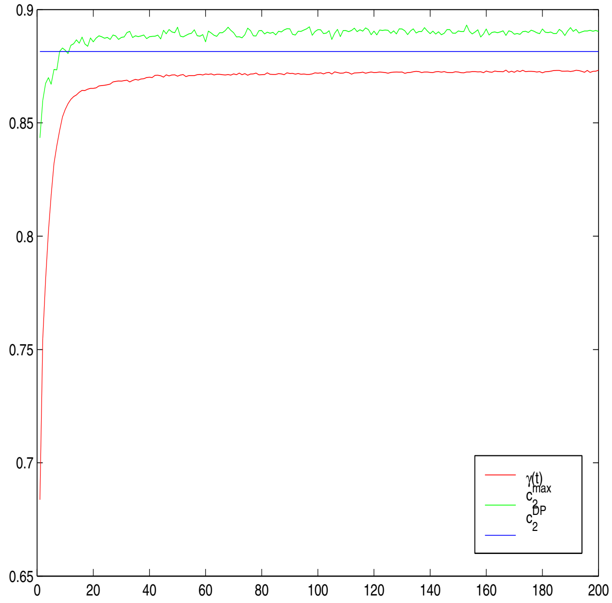

Figure 5. γ (red) and the maximum cost (green) curves during the optimization. Best cost from first approach (blue).

to a solution. Indeed the $\gamma(t)$ and $c^{\max}(t)$ become stable around 20 steps of the CE Moreover, The CE seems make possible the sampling of trajectories with better performance than the best path found by the MDP approach and the performance $\hat{x}_p$. We select as the optimal path, the best one drawn during the last iteration of the CE It is represented in figure (6). Its associated cost is 0.8903 while the upper bound of the high estimated quantile for $p = 2.10^{-5}$ is 0.8829.

VII. Conclusion and Perspectives

In this paper, we investigated a framework to evaluate the optimal path planning for a mobile robot. The goal of the planning task is to find the path which provides good performance of localization using a given landmarks map. This localization performance is directly linked with a functional of the Posterior Cramér-Rao bound. Classical optimization like Dynamic programming was irrelevant due to the functional properties. As a consequence, a learning based approach implying the Cross-Entropy algorithm was used to solve the problem [1]. Some improvements of the first version were introduced in the paper, in particular the adapted cost function to reduce the acceptation-rejection rate. But the main contribution of the current work is the evaluation procedure of this planning algorithm using extreme value theory. This theory gives tools to determine the performance of stochastic optimization via the computation of estimated high quantile. One example was presented to illustrate the reasoning. It confirms that extreme values analysis can be a valuable auxiliary tool to make a decision on the convergence of the path planning algorithm. It is important to know that this framework can also be used to perform comparison of several different path planning algorithms. Future work will concentrate on the generalization

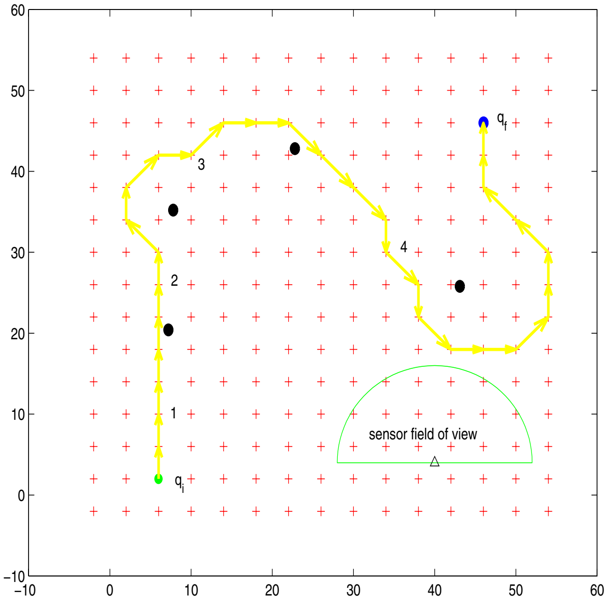

Figure 6. Best path found by the cross entropy algorithm

of this approach for different localization techniques based on unknown but bounded noise models. Indeed, the path planning and the evaluation procedure are relatively independent from the underlying localization methods.

References

[1] F. Celeste, F. Dambreville and J-P. Le Cadre, "Optimal path planning using the Cross-Entropy method" in 9^{th} *International Conference on Information fusion 2006*, Florence 10-13 July, 2006.

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

[3] 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.

[4] LaValle, S.M., *Planning Algorithms*, Cambridge University Press, 2006

[5] 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.

[6] 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.

[7] R. Karlsson, F. Gustafsson Bayesian Surface and Underwater Navigation,IEEE Transactions on Signal Processing, Vol. 54, no. 11, pp. 4204-4213, Nov. 2006.

[8] G. Saporta *Probabilités, analyse des données et statistique*,technip, 1992.

[9] R. Karlsson, F. Gustafsson Bayesian Surface and Underwater Navigation,IEEE Transactions on Signal Processing, Vol. 54, no. 11, pp. 4204-4213, Nov. 2006.

[10] S. Thrun and W. Burgard and D. Fox *Probabilistic Robotics*,MIT press, sept. 2005 vol. 1, no. 1, pp. xxx–yyy, 2006.

[11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert "Robust Monte Carlo Localization for Mobile Robots", *Artificial Intelligence*, Vol. 28, no. 1-2, pp. 99-141, 2000.

[12] C. Durieu, M-J Aldon, D. Meizel *Multisensor Data Fusion for localization in Mobile Robotics*. Traitement du signal, vol. 13, no. 2, 1996 Publisher, Location, Year, 2007.

[13] Venzon D.J., and Moolgavkar,S.H., *A method for Computing Profile-Likelihood-Based Confidence Intervals*, Applied Statistics,1988

[14] E. Këllezi, and M. Gilli, *Extreme Value Theory for Tail-Related Risk Measures*, Computational finance,2000

[15] S. Coles, *Extreme value theory and applications*, technical report,1999