

# Optimized trajectories for mobile robot with map uncertainty

F. Celeste\* F. Dambreville\*\* J.-P. Le Cadre\*\*\*

\* DGA/CEP, 94114, Arcueil Cedex France (e-mail:  
francis.celeste@etca.fr).

\*\* DGA/CEP, 94114, Arcueil Cedex France (e-mail:  
frederic.dambreville@general.com)

\*\*\* IRISA/CNRS, Campus de Beaulieu, 35042, Rennes, France,  
(e-mail: lecadre@irisa.fr)

---

**Abstract:** In this paper we propose a methodology to generate trajectories for a mobile robot which uses an uncertain geometric map to estimate its position and orientation during its displacement. This methodology is based on the definition of particular paths and on defining a cost function to weigh the trade-off between expected localization gain and the risk due to the map uncertainty.

Keywords: Estimation, path planning, stochastic optimization, risk.

---

## 1. INTRODUCTION

In mobile robotics, we often need to estimate a robot state based on a known map of the environment. The map based localization is made from data collected correlated with the prior map. The choice of the appropriate localization algorithm depends on the system dynamics and observation models (Kalman, particle filters...). Before path execution the mission planner has to design trajectories with expected “good localization” performance. In this paper, we introduce a framework to select paths for a mobile robot state (position and heading) estimation from range and bearing data of features in a given 2D uncertain map. One of the main challenges is to choose a relevant metric for the optimization. The first interesting result of this work is the derivation of a closed-form expression for the proposed metric based on the Fisher Information Matrix (FIM) (Van tree [1968]). The impact of the robot constraints (dynamics, limited Field Of View (FOV)...) and the map uncertainty are also dealt with to derive a risk utility function. Then, we formulate the problem as an optimal control problem with integral cost and solve it using an meta-heuristic approach (Rubinstein et al. [2004]). Finally, the paper ends with one illustrative example.

## 2. PROBLEM FORMULATION

### 2.1 The system models

We consider a mobile system which evolves in a 2D environment. Its state  $\mathbf{X}_t$  is a three dimensional vector composed of its 2D position  $(x_t, y_t) \in \mathcal{D} \subset \mathbb{R}^2$  and its orientation  $\varphi_t \in (-\pi, \pi]$  in a global frame  $\mathcal{R}_g := (O, \vec{u}, \vec{v})$  (see Fig. 1). To navigate and get its location in the

environment, an embedded map of salient features are available to “correlate” range and bearing measurements provided by its sensor (e.g. camera, laser rangefinders...). Its motion is governed by a continuous dynamic system

$$\dot{x}_t = v_t \cos \varphi_t, \quad \dot{y}_t = v_t \sin \varphi_t, \quad \dot{\varphi}_t = \omega_t. \quad (1)$$

where  $\mathbf{u}_t := [v_t, \omega_t]$  accounts for the control input at time  $t$  with  $\mathbf{u}_t \in \mathcal{U}$

$$\mathcal{U} := \{(v, \omega), \|v\| \leq \underline{v} \quad \& \quad \|\omega\| \leq \underline{\omega}\}.$$

During execution, the motion is not perfect and a filtering method based on the map and the acquired measurements has to be used to estimate  $\mathbf{X}_t$ . The map  $\mathcal{C}$  is composed of  $N_C$  point features  $f_i, 1 \leq i \leq N_C$  with known but uncertain global 2D position  $(x^i, y^i)$  in  $\mathcal{R}_g$ . More often, they are obtained from a previous mapping stage with variable accuracy (for instance, *Simultaneously Localization And Mapping* SLAM in robotics). We assume that a mean position  $(\underline{x}^i, \underline{y}^i)$  and a  $2 \times 2$  covariance matrix  $\mathcal{E}(f_i)$  which characterizes the uncertainty are given for  $f_i$ . Due to the sensor limitations (maximum and minimum range and aperture see Fig.1) and some processing steps of the localisation procedure, only a few features are visible and detected at each epoch. We define

$$\delta_t^i = \begin{cases} 1 & \text{if } f_i \text{ are used for state estimation at time } t, \\ 0 & \text{else.} \end{cases}$$

Therefore at time  $t$ , we have the associated vector of noisy measurements  $\mathbf{Z}_t = [z_{i_1}^t, \dots, z_{i_{m_t}}^t]$  where  $z_{i_l}^t, \forall l \in \{1, \dots, m_t\}, i_l \in \{1, \dots, N_C\}$  is the range and bearing of  $f_{i_l}$  from the robot state. Then, the observation model stands as follows

$$\mathbf{Z}_t = \mathbf{H}_t(\mathbf{X}_t, \mathcal{C}) + \mathbf{W}_t.$$

$\mathbf{Z}_t$  is a  $2 \times m_t$  vector where the  $2l^{\text{th}}$  and  $2l + 1^{\text{th}}$  elements of  $\mathbf{H}_t(\mathbf{X}_t, \mathcal{C})$  are the components of  $h(\mathbf{X}_t, f_{i_l})$

\* This work was supported in part by the National Technological Agency. (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase.

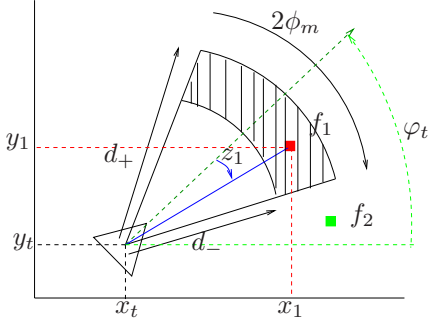


Fig. 1. sensor model ( $d_+$ ,  $d_-$  max and min detection range,  $\phi_m$  half aperture). A visible (red) and non visible (green) landmark.

$$h(\mathbf{X}_t, f_{i_l}) := \begin{cases} \sqrt{(x_t - x^{i_l})^2 + (y_t - y^{i_l})^2}, \\ \text{atan}_2\left(\frac{y^{i_l} - y_t}{x^{i_l} - x_t}\right) - \varphi_t. \end{cases} \quad (2)$$

The noise vector  $\mathbf{W}_t$  is obtained by stacking the noise of each individual measurements  $w_t^{i_l}$  which are supposed to be i.i.d. Gaussian noises with zero mean and  $\Sigma_t^{i_l}$  covariance matrix. Moreover,  $w_t^j$  and  $w_t^l$  are independent for  $l \neq j$  and

$$\forall t, \forall i_l \in \{1, \dots, m_t\}, \Sigma_t^{i_l} = \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_\alpha^2 \end{pmatrix}. \quad (3)$$

In light of the previous assumptions, the observation probability density is

$$p(\mathbf{Z}_t | \mathbf{X}_t) \propto \exp\left(-\frac{1}{2} \sum_{l=1}^{m_t} \|z_{i_l} - h(\mathbf{X}_t, f_{i_l})\|_{\Sigma_t^{i_l}}^2\right). \quad (4)$$

where  $\|B\|_{\Sigma}^2 = B^* \Sigma^{-1} B$ ,  $B \in \mathbb{R}^2$ .

## 2.2 Localization performance

If  $\hat{\mathbf{X}}_t$  is one estimate based on the measurement  $\mathbf{Z}_t$ , for example the maximum likelihood estimate, the covariance of the error of estimation is lower bounded by the inverse on the well-known *Fisher Information Matrix* (FIM) (Van tree [1968]).

$$(\mathbf{X}_t - \hat{\mathbf{X}}_t)(\mathbf{X}_t - \hat{\mathbf{X}}_t)^* \succ F^{-1}.$$

with  $F$  in our case given by the relation,

$$F = \sum_{i=1}^{N_c} \delta_t^i \left( \frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t} \right) \Sigma^{-1} \left( \frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t} \right)^*. \quad (5)$$

Let us introduce  $\delta p_i(t) := [x^i - x_t, y^i - y_t]^*$  the ‘‘mobile-feature  $f_i$ ’’ vector and  $\rho_i := \|\delta p_i(t)\|$ ,  $\alpha_i(t) := \angle \vec{u} \delta p_i(t)$ ,  $c_i := \cos \alpha_i$  and  $s_i := \sin \alpha_i$ . We can easily show that

$$\frac{\partial h(\mathbf{X}_t, f_i)}{\partial \mathbf{X}_t} = \begin{pmatrix} c_i & s_i & 0 \\ -s_i & c_i & -1 \\ \rho_i & \rho_i & -1 \end{pmatrix},$$

and using matrix product properties, we can rewrite  $F$  as  $F(t) = G(t)G(t)^*$  where

$$G(t) = (G_1^d \cdots G_{N_c}^d \ G_1^\alpha \cdots G_{N_c}^\alpha). \quad (6)$$

$\forall 1 \leq i \leq N_c$  vectors ( $G_i^d$ ) and ( $G_i^\alpha$ ) are given by

$$G_i^d = \delta_t^i \sigma_d^{-1} (c_i \ s_i \ 0)^*, \quad G_i^\alpha = \delta_t^i \sigma_\alpha^{-1} \left( \frac{s_i}{\rho_i} \ -\frac{c_i}{\rho_i} \ 1 \right)^*. \quad (7)$$

The FIM is a valuable quantity to derive a relevant *measure of information* to design paths with good expected

estimate accuracy performance. Indeed, several matrix operators can be used. In particular the determinant which can be linked to the volume of the ellipsoid uncertainty is of high interest. In the next section, we quickly remind the closed form expression for the determinant introduced in (Celeste et al. [2008]). It is interesting to note that this expression directly implies the norms and arguments of vectors  $\delta p_i(t)$ .

## 2.3 A closed form performance criteria expression

We are now concerned with the derivation of the function  $\mathcal{L}(t, \mathcal{C})$  defined by

$$\mathcal{L}(t, \mathcal{C}) = \det(G(t)G(t)^*). \quad (8)$$

The closed form expression for  $\mathcal{L}(t, \mathcal{C})$  can be obtained by applying the algebra property *Binet-Cauchy formula*<sup>1</sup> with  $B = A = G$  and  $m = 3$ ,  $n = 2N_c$ . Indeed,

$$\mathcal{L}(t, \mathcal{C}) = \sum_{1 \leq i < j < r \leq 2N_c} \{\det(G_i(t), G_j(t), G_r(t))\}^2. \quad (9)$$

Therefore, we have to compute determinants of three dimensional matrices by enumerating the different configurations made from groups of three columns given by (eq. 7). By applying determinant computation properties and relations between trigonometric functions Celeste et al. [2008]

$$\mathcal{L}(t, \mathcal{C}) = a_1 \mathcal{L}_1(t, \mathcal{C}) + a_2 \mathcal{L}_2(t, \mathcal{C}) + a_3 \mathcal{L}_3(t, \mathcal{C})$$

with  $a_1 = \sigma_d^{-4} \sigma_\alpha^{-2}$ ,  $a_2 = \sigma_d^{-2} \sigma_\alpha^{-4}$ ,  $a_3 = \sigma_\alpha^{-6}$  and

$$\mathcal{L}_1(t, \mathcal{C}) = \sum_{i=1}^{N_c} \sum_{j>i}^{N_c} \sum_{r=1}^{N_c} \delta_t^i \delta_t^j \delta_t^r g_{ijr}^{(1)}(t), \quad (10)$$

$$\mathcal{L}_2(t, \mathcal{C}) = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} \sum_{r>j}^{N_c} \delta_t^i \delta_t^j \delta_t^r g_{ijr}^{(2)}(t), \quad (11)$$

$$\mathcal{L}_3(t, \mathcal{C}) = \sum_{i=1}^{N_c} \sum_{j>i}^{N_c} \sum_{r>j}^{N_c} \delta_t^i \delta_t^j \delta_t^r g_{ijr}^{(3)}(t), \quad (12)$$

and  $\forall i, j, k$ ,

$$g_{ijr}^{(1)}(t) = \sin^2(\alpha_i - \alpha_j) \quad (13)$$

$$g_{ijr}^{(2)}(t) = \left( \frac{\cos(\alpha_i - \alpha_r)}{\rho_r} - \frac{\cos(\alpha_i - \alpha_j)}{\rho_j} \right)^2$$

$$g_{ijr}^{(3)}(t) = \left( \frac{\sin(\alpha_i - \alpha_r)}{\rho_i \rho_r} + \frac{\sin(\alpha_i - \alpha_j)}{\rho_i \rho_j} + \frac{\sin(\alpha_j - \alpha_r)}{\rho_j \rho_r} \right)^2.$$

We can remark that the first term only depends on information provided by range measures, the third one only on bearing measurements, whereas the one in the middle takes into account combinations of both kind of measurements. In (Celeste et al. [2008]), we make a geometric analysis to determine the optimal placement of the features for a given mobile position in order to maximize the information linked with  $\mathcal{L}_1(t, \mathcal{C})$ , in particular for sensors with limited FOV. Based on this performance criteria,

<sup>1</sup>  $\det(AB) = \sum_S \det(A_S) \det(B_S)$ ,  $S \subset \{1, \dots, n\}$  is a subset of size  $m$ ,  $A$  is a  $m \times n$  matrix and  $B$  a  $n \times m$  matrix,  $A_S$  ( $B_S$ ) is a  $m \times m$  matrix whose columns (rows) are those of  $A$  ( $B$ ) with indices in  $S$

derive an integral cost for our path planning problem. From now on and for the sake of clarity, we only consider the component  $\mathcal{L}_1$  but the reasoning can be generalized to  $\mathcal{L}_2$  and  $\mathcal{L}_3$ .

### 3. THE PATH PLANNING PROBLEM

#### 3.1 An optimal control problem

We suppose that motion is made between  $t_0$  and  $t_f$  with  $t_0 < t_f$  from the start position  $q_s \in \mathcal{D}$  to the target set  $\mathcal{B}(q_f, R_f)^2 \subset \mathcal{D}$ . The main goal is to design the path  $(\mathbf{X}_t)_{t \in [t_0, t_f]}$  (or equivalently sequence of  $\mathbf{u}_t$ ) which maximizes

$$\Psi([t_0, t_f]) = \int_{t_0}^{t_f} \mathcal{L}(t) dt.$$

The problem can be formalized in the optimal control framework with two boundary constraints. Thus, the Hamiltonian and also the Euler-Lagrange conditions can be derived. Unfortunately, due to the cost expression, it is rarely possible to get an analytic formulation of the optimal path. Moreover, we have to deal with the limited sensing capabilities (FOV). Consequently, the proposed more tractable approach is restricted to the set of paths of sequences of motions with constant velocity and constant heading.

#### 3.2 Description of the path space

First we consider a graph  $\mathcal{G}(V, E)$  composed of  $|V|$  nodes  $(s_q, 1 \leq q \leq |V|)$  and  $|E|$  edges  $\{e_{pq} = (s_p, s_q) \in V^2, q \in \mathcal{V}(p)\}$ , where  $\mathcal{V}(p)$  is the set of reachable nodes from  $p$ . Each node  $s_p$  corresponds to a 2D position  $(x_{s_p}, y_{s_p}) \in \mathcal{D}$ . The robot moves on the edges with constant velocity  $v_e$  and  $\varphi_e$  heading the argument of  $e$  in  $\mathcal{R}_g$ . We also denote  $\Delta_{s_p}^{s_q}$  the travel time between two nodes  $(s_p, s_q)$ . As the velocity is bounded,  $\Delta_{s_p}^{s_q}$  is also bounded and depends on the length  $l_{pq}$  of  $e_{pq}$ . An admissible path  $\tau$  is composed of  $n_\tau + 1$  nodes  $(s_{i_k})_{k=0, \dots, n_\tau}$  with  $s_{i_0} = q_s$  and  $s_{i_{n_\tau+1}} \in \mathcal{B}(q_f, R_f)$ . The mobile reaches node  $s_{i_k}$  at time  $t_k$  and the total duration of the travel is  $\Delta_\tau = \sum_{k=0}^{n_\tau} \Delta_{s_{i_k}}^{s_{i_{k+1}}} = \sum_{k=0}^{n_\tau} \frac{l_{i_k i_{k+1}}}{v_k}$ ,  $v_k$  is the velocity on the edge  $e_{i_k i_{k+1}}$ .

#### 3.3 The cost associated to a path

The cost of a path  $\tau$  is calculated straightforwardly :

$$\Psi(\tau, \mathcal{C}) = a_1 \sum_{k=0}^{n_\tau} \overbrace{\int_{t_k}^{t_{k+1}} \mathcal{L}_1(t, \mathcal{C}) dt}^{\Gamma^1(k)}. \quad (14)$$

Now, we need to compute each elementary term  $\Gamma^1(k)$  depending on the visible part of the map during the motion on  $e_{i_k i_{k+1}}$ . Using (10), we have

$$\Gamma^1(k) = \sum_{i=1}^{N_c} \sum_{j>i}^{N_c} \sum_{r=1}^{N_c} \overbrace{\int_{t_k}^{t_{k+1}} \delta_t^i \delta_t^j \delta_t^r g_{ijr}^{(1)}(t) dt}^{\Gamma_{ijr}^1(k)}. \quad (15)$$

<sup>2</sup>  $\mathcal{B}(a, r)$  is the ball with center  $a$  and radius  $r$

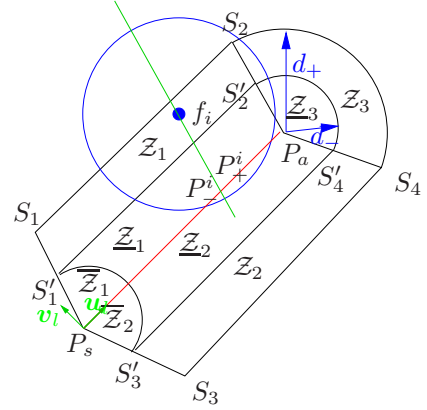


Fig. 2. Sensor coverage for one elementary motion.

#### 3.4 Sensing coverage along an edge

We suppose that the sensing area of the sensor is characterized by three parameters  $d_-$ ,  $d_+$  and  $\phi_m \leq \frac{\pi}{2}$  which are the minimum and maximum range detection and the half aperture angle. So, during a displacement on one edge from  $P_s$  to  $P_a$  the area visible by the sensor is the union of the regions  $Z_1$ ,  $Z_2$ ,  $Z_3$ ,  $Z_1/Z_2$ ,  $Z_2/Z_3$  and  $Z_3$  made from point  $(S_i)_{1 \leq i \leq 4}$ ,  $(S'_i)_{1 \leq i \leq 4}$ ,  $P_s$  and  $P_a$  (see Fig. 2). For each  $f_i$ ,  $1 \leq i \leq N_c$ , it is possible from geometric reasoning to deduce the positions  $P_-^i$  and  $P_+^i$  where it is seen for the first and last time. Finally, it is possible to get for all groups of three features  $(f_i, f_j, f_r)$  the points  $P_-^{ijr}$  and  $P_+^{ijr}$  which define the part of the edges where they are simultaneously visible. We are going to show that  $\Gamma_{ijr}^1(k)$  can be explicitly deduced from these points.

#### 3.5 $\Gamma_{ijr}^1(k)$ derivation

As the motion is made with constant  $v_k$  and  $\varphi_k$ , we have the relation between infinitesimal displacement and the associated infinitesimal duration

$$dt = \begin{cases} \frac{dx}{v_k \cos(\varphi_k)} & \text{if } \varphi_k \neq \frac{\pi}{2}[\pi] \text{ (non vertical motion),} \\ \frac{\pm dy}{v_k} & \text{else (vertical motion).} \end{cases}$$

Moreover, we have the parameterization along the edge

$$y = \beta + \gamma x, \forall x \in [x_{s_{i_k}}, x_{s_{i_{k+1}}}] \text{ if } \varphi_k \neq \frac{\pi}{2}[\pi] \quad (16)$$

$$x = x_{s_{i_k}}, \quad y_{s_{i_k}} \leq y \leq y_{s_{i_{k+1}}} \text{ else.} \quad (17)$$

We can also express  $\rho_l$  and  $\sin(\alpha_m - \alpha_n)$  as functions of the position coordinates. Indeed, whatever the value of  $\varphi_k$ , we have  $\rho_l = p_l(u) := a_l u^2 + b_l u + c_l$  and

$$\sin^2(\alpha_m - \alpha_n) = \frac{p_{mn}^2(u)}{p_m(u)p_n(u)} \quad (18)$$

with  $p_{mn}(u) := a_{mn}u + b_{mn}u$ . The coefficients of the polynomials depend on the orientation of the edge and are computed from relation (16). The cost  $\Gamma_{ijr}^1(k)$  can then be computed using relevant change of variables and are in fact obtained by integration of rational functions. The integration is done with the  $x$  and  $y$  variables respectively for non vertical motion and vertical motion. More precisely<sup>3</sup>,

<sup>3</sup> we can note that  $\Gamma_e^{(1)} \propto \frac{1}{v_e}$

$$\Gamma_{ijr}^{(1)}(k) = \begin{cases} \frac{1}{v_k \cos(\varphi_k)} \int_{x_-^{ijr}}^{x_+^{ijr}} \frac{p_{ij}^2(x)}{p_i(x)p_j(x)} dx & \text{if } \varphi_k \neq \frac{\pi}{2} [\pi] \\ \pm \frac{1}{v_k} \int_{y_-^{ijr}}^{y_+^{ijr}} \frac{p_{ij}^2(y)}{p_i(y)p_j(y)} dy & \text{else.} \end{cases} \quad (19)$$

To complete the computation, we have to proceed to partial expansion of the rational functions. It is not the same if  $p_i(y) = p_j(y)$  or not, so we have to pay attention to whether the edge support corresponds to the perpendicular bisector of  $[f_i f_j]$ .

(1) If  $e_{s_{i_k} s_{i_{k+1}}}$  is on the perpendicular bisector

$$\frac{(a_{ij}u + b_{ij})^2}{p_i(u)p_j(u)} = \frac{r_1u + s_1}{p_i(u)} + \frac{r_2u + s_2}{p_i^2(u)}. \quad (20)$$

(2) else

$$\frac{(a_{ij}u + b_{ij})^2}{p_i(u)p_j(u)} = \frac{r_1u + s_1}{p_i(u)} + \frac{r_2u + s_2}{p_j(u)}. \quad (21)$$

Identification of both side numerators yields the linear systems to deduce the vector  $\chi = [r_1 \ s_1 \ r_2 \ s_2]^*$ ,

$$M_{ij}^{(c)} \chi = \mathcal{B}_{ij}, \text{ for } c = 1, 2 \quad (22)$$

with

$$M_{ij}^{(1)} = \begin{pmatrix} a_i & 0 & 0 & 0 \\ b_i & 0 & a_i & 0 \\ c_i & 1 & b_i & 0 \\ 0 & 0 & c_i & 1 \end{pmatrix}, M_{ij}^{(2)} = \begin{pmatrix} a_j & 0 & a_i & 0 \\ b_j & a_j & b_i & a_i \\ c_j & b_j & c_i & b_i \\ 0 & c_j & 0 & c_i \end{pmatrix} \quad (23)$$

and

$$\mathcal{B}_{ij} = (0 \ a_{ij}^2 \ 2a_{ij}b_{ij} \ b_{ij}^2)^*. \quad (24)$$

Hence,  $\Gamma_{ijr}^{(1)}(k)$  is derived from integrals like ( $n \in \{1, 2\}$ ,  $l \in \{i, j\}$ ):

$$H^{(n)}(l, r, s, x_-, x_+) = \int_{x_-}^{x_+} \frac{rx + s}{p_l^n(x)} dx \quad (25)$$

Such integrals can be computed using classic primitives and integration by parts and we have the final result

$$H^{(1)}(\cdot) = \nu_l^{(1)} \ln \left( \frac{p_l(x_+)}{p_l(x_-)} \right) \quad (26)$$

$$+ \mu_l^{(1)} \arctan \left( \frac{2a_l \sqrt{D_l} (x_+ - x_-)}{D_l + (2a_l x_+ + b_l)(2a_l x_- + b_l)} \right),$$

$$H^{(2)}(\cdot) = \frac{1}{D_l} \left( \frac{\nu_l^{(2,1)} x_+ + \nu_l^{(2,2)}}{p_l(x_+)} - \frac{\nu_l^{(2,1)} x_- + \nu_l^{(2,2)}}{p_l(x_-)} \right)$$

$$+ \mu_l^{(2)} \arctan \frac{2\sqrt{D_l} a_l (x_+ - x_-)}{D_l + (2a_l x_+ + b_l)(2a_l x_- + b_l)} \quad (27)$$

with  $D_l = 4a_l c_l - b_l^2$ ,  $\nu_l^{(1)} = \frac{r}{2a_l}$ ,  $\mu_l^{(1)} = \frac{2a_l s - b_l r}{a_l \sqrt{D_l}}$ ,  $\nu_l^{(2,1)} = (2a_l s - b_l r)$ ,  $\nu_l^{(2,2)} = (b_l s - 2r c_l)$  and  $\mu_l^{(2)} = \frac{2\nu_l^{(2,1)}}{D_l}$ . More

precisely, we have finally

$$\Gamma_{ijr}^{(1)}(k) = \gamma_k (H^{(1)}(i, r_1, s_1, u_-^{ijr}, u_+^{ijr}) + H^{(m)}(j, r_2, s_2, u_-^{ijr}, u_+^{ijr})). \quad (28)$$

with  $m = 1$  for a displacement on the perpendicular bisector and  $m = 2$  else,  $(r_1, s_1, r_2, s_2)$  are obtained via the linear systems introduced above, and  $u = x$  and  $\gamma_k = 1/v_k \cos \varphi_k$  for non vertical motion and  $u = y$  and  $\gamma_k = \pm \frac{1}{v_k}$  for vertical motion. To sum up, we can notice

that in order to compute  $\Gamma^1(k)$ , it is necessary to compute the quantities  $\Gamma_{ijr}^1(k)$ ,  $(i, j, r) \in \{1, \dots, N_C\}^3$ ,  $j > i$  for the edge traversed at stage  $k$ . So we have to consider a cube matrix with  $N_C^3$  elements equal to  $\Gamma_{ijr}^1(k)$ . It is important to understand that a great part of those elements are equal to zero as we must have  $j > i$ .

## 4. UNCERTAINTY MANAGEMENT

### 4.1 A risk utility function for the optimization

All the materials introduced in the previous section consider that the map  $\mathcal{C}$  is perfect. Nevertheless the features' coordinates positions are uncertain and the cost  $\Psi(\tau, \mathcal{C})$  is therefore a random variable. So, we are facing a stochastic optimization problem that we propose to solve by modifying the original cost function. Indeed, we consider the following *risk function* (Ruszczyński et al. [2006])

$$r(\Psi(\tau, \mathcal{C})) = \mathbb{E}_C \{ \Psi(\tau, \mathcal{C}) \} - \lambda \mathbb{D}_C \{ \Psi(\tau, \mathcal{C}) \}. \quad (29)$$

with  $\lambda > 0$  and  $\mathbb{D}_C \{ \Psi(\tau, \mathcal{C}) \}$  a measure of dispersion. Some hints on the choice of the  $\mathbb{D}_C \{ \Psi(\tau, \mathcal{C}) \}$  and their associated properties on the optimization problem can be found in (Ruszczyński et al. [2006] and Artzner et al. [1999]). For our problem, we investigate the standard deviation as the measure of dispersion (*mean-deviation risk function*):

$$\mathbb{D}_C \{ \Psi(\tau, \mathcal{C}) \} = \left( \mathbb{E}_C \left\{ (\Psi(\tau, \mathcal{C}) - \mathbb{E}_C \{ \Psi(\tau, \mathcal{C}) \})^2 \right\} \right)^{\frac{1}{2}} \quad (30)$$

It seems difficult to get a closed-form expression of the mean and standard deviation of the cost, so we proceed by Monte Carlo or an equivalent method such as the Unscented Transformation (UT) (Julier et al. [1996]) to get estimates of those quantities.

### 4.2 Approximation using the UT

The UT approach consists in defining a minimal set of deterministically chosen points to represent the uncertainty associated to a random vector. These particular points called *sigma vectors* completely capture the two first moments (mean and covariance). Based on these points the two first moments of the random vector obtained via a nonlinear transformation can be estimated. In our particular case, we consider the stacked vector (also denote  $\mathcal{C}$ ) of size  $2N_C$  composed of coordinates of the map features, i.e.

$$\mathcal{C} = (x^1, y^1, \dots, x^{N_C}, y^{N_C}).$$

The mean and the covariance of  $\mathcal{C}$  are respectively

$$\underline{\mathcal{C}} = (\underline{x}^1, \underline{y}^1, \dots, \underline{x}^{N_C}, \underline{y}^{N_C}) \quad (31)$$

and  $\mathcal{E}(\mathcal{C}) = \text{diag}(\mathcal{E}(f_1), \dots, \mathcal{E}(f_{N_C}))$ . The "sigma vectors"  $\mathcal{C}_l$ ,  $0 \leq l \leq 2(2N_C)$  and their respective weights  $W_m^{(l)}$  and  $W_c^{(l)}$  (Julier et al. [1996]) are

$$\mathcal{C}_0 = \underline{\mathcal{C}}, \quad (32)$$

$$\mathcal{C}_l = \underline{\mathcal{C}} + \sqrt{2N_C + \kappa} \sqrt{\mathcal{E}(\mathcal{C})}_l, \quad l = 1, \dots, 2N_C$$

$$\mathcal{C}_l = \underline{\mathcal{C}} - \sqrt{2N_C + \kappa} \sqrt{\mathcal{E}(\mathcal{C})}_{l-2N_C}, \quad l = 2N_C + 1, \dots, 4N_C$$

with  $W_m^{(0)} = \kappa / \sqrt{(2N_C + \kappa)}$ ,  $W_c^{(0)} = W_m^{(0)} + (1 - c_1^2 + c_2)$  and  $W_c^{(l)} = W_m^{(l)} = 1/2(2N_C + 1)$ ,  $l > 0$ .  $\kappa$ ,  $c_1$  and  $c_2$  are three real parameters.  $\sqrt{\mathcal{E}(\mathcal{C})}_j$ ,  $1 \leq j \leq 2N_C$  is the  $j^{\text{th}}$

column of the square root of the matrix  $\mathcal{E}(\mathcal{C})$  obtained from Cholesky factorization. For a given edge  $e_{s_{i_k} s_{i_{k+1}}}$ , we can compute the elementary cost  $\Gamma_l^{(1)}(k)$  for each new map  $\mathcal{C}_l$  using the algorithm introduced in section 3 and estimates of the first two moments of  $\Gamma^{(1)}(k)$

$$\hat{\Gamma}^{(1)}(k) = \sum_{l=0}^{2N_c} W_m^{(l)} \Gamma_l^{(1)}(k),$$

$$\hat{\sigma}_{\Gamma^{(1)}(k)}^2 = \sum_{l=0}^{2N_c} W_c^{(l)} \left( \Gamma_l^{(1)}(k) - \hat{\Gamma}^{(1)}(k) \right)^2.$$

We can then deduce the estimate of the mean and the variance of the cost for a path  $\tau$  with edges  $e$  traversed with velocity  $v_e$ <sup>4</sup>

$$\hat{\Psi}(\tau, \mathcal{C}) = \sum_{k=0}^{n_\tau} \hat{\Gamma}^{(1)}(k) := \sum_{e \in \tau} \frac{1}{v_e} c(e),$$

$$\hat{\sigma}_{\Psi(\tau, \mathcal{C})}^2 = \sum_{k=0}^{n_\tau} \hat{\sigma}_{\Gamma^{(1)}(k)}^2 := \sum_{e \in \tau} \frac{1}{v_e^2} \sigma_{c(e)}^2.$$

## 5. A SOLUTION BASED ON THE CROSS ENTROPY METHOD

Given  $\mathcal{G}(V, E)$  with the mean cost  $c(e)$  and the variance  $\sigma_{c(e)}^2$  for each edge  $e$  traversed with velocity  $v_e$ , the source node  $s_{i_0}$ , the set of final nodes  $B_f$ , the maximum delay  $\Delta_m = t_f - t_0$ , and a given  $\lambda$  decision parameter. The constrained optimization problem is

$$\max_{\tau \in P(s_0, B_f), (v_i)_{0 \leq i \leq n_\tau - 1}} \sum_{e \in \tau} \frac{1}{v_e} c(e) - \lambda \sqrt{\sum_{e \in \tau} \frac{1}{v_e^2} \sigma_{c(e)}^2} \quad (33)$$

where  $P(s_0, B_f)$  is the set of paths starting in  $s_0$  and ending in  $B_f$  which satisfy

$$\sum_{e \in \tau} \frac{l(e)}{v_e} \leq \Delta_m \quad (34)$$

$$\forall e, f(e, \text{succ}(e)) = 1 \quad (35)$$

$$\forall e, \underline{v}_- \leq v_e \leq \underline{v}_+ \quad (36)$$

So we need to solve a hierarchical optimization problem, ones the path is chosen it is necessary to optimize the sequence of velocities. This second level constrained non-linear optimization problem is very hard to solve. In this paper, we suppose that the travel time  $\frac{l(e)}{v_e}$  is constant and equal to  $\Delta T$  for all arcs. That is to say the nodes and the velocities are automatically adjusted to satisfy those conditions. We continue to investigate the original problem in future works. Due to the nature of the cost function, it is not possible to use a classic optimization approach such as dynamic programming, therefore we propose to apply the Cross Entropy method (Rubinstein et al. [2004]).

### 5.1 The Cross Entropy method

Consider the following optimization problem:

$$\phi(x^*) = \gamma^* = \max_{x \in \mathcal{X}} \phi(x) \quad (37)$$

<sup>4</sup> we showed that  $\Gamma^{(1)}(k)$  for edge  $e$  can be written as  $c(e)/v_e$

The principle of the CE for optimization is to translate (37) into an *associated stochastic problem* and then solve it adaptively as the simulation of a rare event. If  $\gamma^*$  is the optimum of  $\phi$  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  $\pi(\cdot, \eta) | \eta \in \Upsilon$ , to draw samples around  $\gamma^*$ . The learning stage consists in solving an optimization problem to minimize the *Kullback-Leibler* “pseudo-distance” for improving the performance simulation in the tail of the underlying density. Unlike other local random search algorithms like simulated annealing which assume 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.  $\pi(\cdot, \eta) | \eta \in \Upsilon$ , the algorithm for the optimization proceeds as follows :

- (1) Initialize  $\eta_t = \eta_0$  (here  $t$  is different from the time index in previous section).
- (2) Generate a sample of size  $N$   $(x_i^t)_{1 \leq i \leq N}$  from  $\pi(\cdot, \eta_t)$ , compute  $(\phi(x_i^t))_{1 \leq i \leq N}$  and order them from smallest to biggest. Estimate  $\gamma_t$  as the  $(1 - \rho)$  sample percentile.
- (3) update  $\lambda_t$  with :

$$\eta_{t+1} = \operatorname{argmax}_{\eta} \frac{1}{N} \sum_{i=1}^N I[\phi(x_i^t) \geq \gamma_t] \ln \pi(x_i^t, \eta) \quad (38)$$

- (4) repeat from step 2 until “convergence”.
- (5) assume convergence is reached at  $t = t^*$ , an optimal value for  $\phi$  can be obtained from a sampling made with  $\pi(\cdot, \eta_{t^*})$ .

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

$$\eta_{t+1} = \nu \tilde{\eta}_{t+1} + (1 - \nu) \eta_t \quad (39)$$

### 5.2 Application to the path planning task

To apply the CE, we need to define a family of pdfs to generate admissible trajectories. We introduce the family of probability matrices  $\mathbf{P}_{ss'} = (p_{ss'})$  with  $(s, s') \in V \times V$ .  $p_{ss'}$  is the probability to decide to go towards node  $s'$  from nodes  $s$ . These matrices are often sparse as many nodes have limited neighborhood  $\mathcal{V}(s)$  due to the map configuration (border, forbidden areas...) and bounded motion velocity.

$$\mathbf{P}_{ss'} = \begin{pmatrix} p_{s_1 s_1} & p_{s_1 s_2} & \cdots & p_{s_1 s_{|V|-1}} & p_{s_1 s_{|V|}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{s_{|V|} s_1} & p_{s_{|V|} s_2} & \cdots & p_{s_{|V|} s_{|V|-1}} & p_{s_{|V|} s_{|V|}} \end{pmatrix} \quad (40)$$

with  $\forall s, \mathbf{P}_{ss'}(s, \cdot)$  is a discrete probability law such as :

$$\mathbf{P}_{ss'}(s, s_i) = p_{s s_i}, \quad i = \{1, \dots, |V|\}, \quad \text{with} \quad \sum_{i=1}^{|V|} p_{s s_i} = 1$$

At each iteration of the CE algorithm, we can generate  $N$  *admissible trajectories*  $\tau$  with travel time  $T_\tau \leq \Delta_m$  in accordance with  $\mathbf{P}_{ss'}$  and an *authorized transition matrix*  $\delta$  for constraint to satisfy (35). Let us denote  $s_{i_k}$  and  $e_{k-1}$  the node reach at step  $k - 1$  and the edge used to

reach it. We introduce  $E_{k,k-1}$  the set of possible edges for displacement at step  $k$  in  $s_{i_k}$  knowing that edge  $e_{k-1}$  was taken and  $\tilde{P}_{s_{i_k}}(\cdot)$  is the normalized restriction of  $P_{s_{i_k}}(\cdot)$  to  $E_{k,k-1}$ . During this generation procedure, due to the maneuver constraints, the set  $E_{k,k-1}$  may be empty or the final state may be not attainable, so the associated value to that trajectory is set to  $-\infty$ , else the associated cost is evaluation of the risk function. Given  $N$   $\{\tau_j\}_{j=1}^N$  samples of path with associated cost  $\{r(\tau_j)\}_{j=1}^N$  at iteration  $t$ , the update stage for the elements of  $\mathbf{P}_{ss'}$  is given by the following expression Celeste et al. [2008]:

$$p_{ss'} = \frac{\sum_{j=1}^N I[\{r(\tau_j) \geq \gamma_t\}] \cdot I[\{\tau_j \in \chi_{ss'}\}]}{\sum_{j=1}^N I[\{r(\tau_j) \geq \gamma_t\}] \cdot I[\{\tau_j \in \chi_s\}]} \quad (41)$$

where  $\{\tau_j \in \chi_s\}$  means that the trajectory contains a visit to state  $s$  and  $\{\tau_j \in \chi_{ss'}\}$  means that the trajectory contains a visit to state  $s$  to reach  $s'$ . For the first iteration,  $\forall s, \mathbf{P}_{ss'}(s, \cdot)$  is taken as a uniform probability density function on admissible nodes.

## 6. EXPERIMENT

We introduce a short example with a map composed of several features with different level of accuracy on their coordinates. The best part of the map is on the upper left side of the environment in  $\mathcal{D} = [-10, -10, 110, 110]$  (see Fig. 3). The graph is a regular grid (red crosses on the figure) with the same resolution in  $x$  and  $y$  equal to  $dx = dy = 20$ . The sensor FOV is limited and characterized by a minimum and maximum range detection  $d_- = 10$  and  $d_+ = 40$  and a half aperture angle  $\phi_m = 90 \text{ deg.}$ . Moreover, the headings variation between two following time steps must be bounded by  $\pi/2$ . For the optimization, the path must be composed of  $K = 20$  elementary branches and the CE algorithm is run with 5000 trajectory samples at each of the 200 iterations. The selection rate is  $\rho = 0.2$ , therefore the 2000 best paths contributed to the update stage of the  $\mathbf{P}_{ss'}$  matrix. The smoothing parameter is  $\nu = 0.4$ . We compute the optimal path for  $\lambda = 0, 3$  and  $8$ . The first case is the path obtained from the maximum mean cost. The results are plotted on figures (Fig.3) and (Fig.4). On the mean cost path, the mobile tries to see a maximum number of features located on each side of its current trajectories without taking into account the level of uncertainty on their position. This behavior seems coherent as maximizing  $\Gamma^{(1)}(\cdot)$  is equivalent to improve coverage and triangulation performance. On the opposite side, when  $\lambda$  increases, we can notice that the optimized robot maneuvers to sense a maximum number of features but with high accurate positions (located on the right side) in order to reduce the uncertainty (*or the risk*) on the localization metric.

## 7. CONCLUSIONS AND PERSPECTIVE

In this paper, we investigated a framework to design the optimal path for a mobile robot for map-based localization. A closed form measure of performance directly linked with the Fisher Information Matrix is derived and adapted to take into account the map probabilistic uncertainty. Classical optimization method like Dynamic programming is useless to solve the derived stochastic optimization problem. As a consequence, a learning based approach using

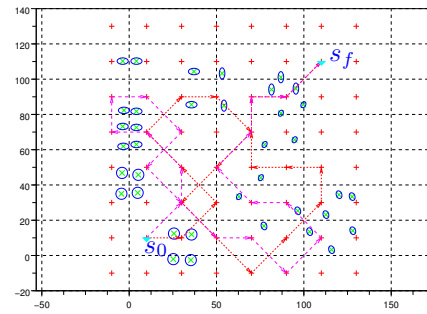


Fig. 3. path planning for  $\lambda = 0$  (red) and  $\lambda = 3$  (magenta).

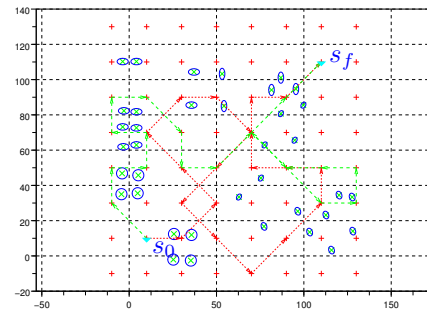


Fig. 4. path planning for  $\lambda = 0$  (red) and  $\lambda = 8$  (green).

the CE algorithm was used. One example was presented to illustrate the global reasoning. It confirms that the approach is a valuable one to select paths based on the trade-off between maximum mean and limited risky localisation performance. Future work will concentrate on the evaluation of more experiments and on the generalization of the approach to optimized also the sequence of velocities which was not tackled here.

## REFERENCES

- H.L. Van Trees. Detection, Estimation and Modulation Theory. *New York Wiley*, 1968.
- F. Celeste, F. Dambreville, and J.-P. Le Cadre. Path planning for multiple features based localization. In *Proceedings of the ICINCO'08, Madeira, 2008*.
- S.J. Julier, and J.K. Uhlmann. A general method for approximating Nonlinear transformation of probability distribution. Technical report, RRG, dept. of Engineering Science, University of Oxford, 1996.
- A. Ruszczyński, and A. Shapiro. Optimization of convex risk function. *Mathematics of operations research*, volume 31, No. 3 pages 433–452, 2006.
- P. Artzner, F. Delbaen, J.-M. Eber and D. Heath. Coherent measures of risk, *Math. Finance*, 9, pages 203–228, 1999.
- S. Paris, and J.-P. Le Cadre. Trajectory planning for terrain-aided navigation, *Proceedings of the Int. Conference on information fusion*, 2002.
- R.Y. Rubinstein, and D. Kroese. *The Cross-Entropy Method: A uniform approach for Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning* Information Science Statistics, 2004.