# Planification for Terrain-Aided Navigation

Sébastien Paris
IRISA/INRIA
Campus de Beaulieu
35042 Rennes, France.
sparis@irisa.fr

Jean-Pierre Le Cadre *
IRISA/CNRS
Campus de Beaulieu
35042 Rennes, France.
lecadre@irisa.fr

**Abstract** – *The purpose of this paper is to investigate the planification of a mobile trajectory in order to use its own motion for improving its position estimation . This optimization procedure relies upon two basic ingredients: 1) a reference measurement map of the area of interest is available before departure. 2) real time measurement sensors placed aboard the mobile. The main objective is to plan a trajectory which minimizes the localization error along the path or/and at the arrival area. The general framework of the Markov decision process coupled with an auxiliary local cost function are the basic ingredients of a sub-optimal algorithm. Quality of the optimization scheme is evaluated by deriving the Posterior Cramér-Rao bounds of the non linear discrete-time system.*

**Keywords:** Terrain-Aided Navigation, optimization, Markov Decision Process, Posterior Cramér-Rao Bounds.

## 1 Introduction

For military or civilian applications, terrain-aided navigation (TAN) permits the self-localization of a mobile without any help of exterior systems (see [1, 2, 3]). This task can successfully be achieved by correlating informations given by onboard sensors at the sequence of positions with a reference measurement map previously stacked. To a large extent, the difficulty comes from the strong nonlinearity of the measurement versus position, resulting in divergence of the classic EKF position filters. To remedy this problem, Viterbi algorithm (see [2]) or more recent particle filters (PF) (see [1, 3]) have been developed. They are both feasible and reliable. A main property of PF is to reach the Posterior Cramér-Rao bounds (PCRB) [4, 5] only in a few iterations. The PCRB measures the maximum information which can be extracted from a dynamic system when both measurement and state are assumed to be random. For TAN, the PCRB is evaluated for a given nominal trajectory coupled with an error diffusion model of the state vector.

A crucial task for operators is to schedule a path maximizing the PCRB collected among it. This off-line optimization will indirectly improve the position estimation of the mobile. A such *a priori* procedure can be formulated within the Markov decision process (MDP) framework [6], when the system state is completely observed. However, if the state is only partially observed (e.g. via non-linear scalar measurements), then Partially observed Markov decision process (POMDP) [7] is the general framework. We stress that this framework is quite general but also very demanding (especially for memory requirement). For our problem, the MDP framework will be the workhorse for the planification tool. Taking into account precision requirements, adaptivity is useless.

In order to simplify the problem, the optimized path is supposed to be a multileg one (sequence of constant headings with constant velocity). The reference measurement map is also regularly gridded, each point of the grid corresponds to an end point of one leg. We also suppose that the mobile path starts and ends at two points of the grid. An example of measurement map (Colorado state area taken by landsat) as well as a path trajectory is provided in figure (1). We first assume that the path optimization problem is equivalent to finding the best decision sequence maximizing an auxiliary convex cost function. Both state and decision spaces are supposed to be discrete and finite. For the sequel, a decision is assimilated to an elementary move of the mobile between two points of the map. This cost function tends to represent as well as possible the terrain variation information collected during the mobile motion.

*This work has been supported by LRBA (Laboratoire de recherches balistiques et aérodynamiques), Vernon, FRANCE

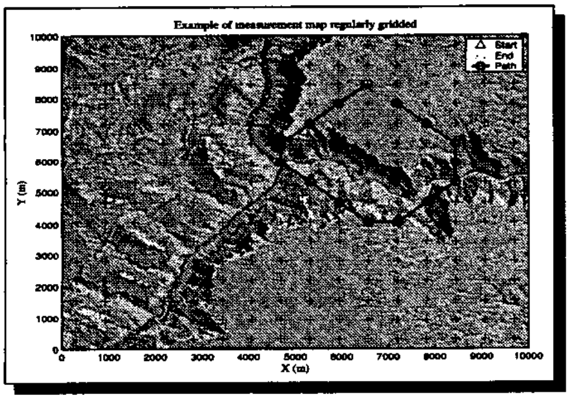

Figure 1: Example of a map regularly gridded with a nominal path.

The rest of the paper is articulated in four parts. The second section describes the basic ingredient of the MDP processes. Section three introduces a path optimization algorithm based on the MDP framework in order to incorporate maneuverer contraints. Section four is an overview of the PCRB for discrete-time non linear dynamic system. The PCRB permits to validate quality of our result and is followed by conclusions and discussions.

## 2 Problem formulation

Let us define the state vector by $\boldsymbol{x}(t_k) \triangleq [r_x(t_k), v_x(t_k), r_y(t_k), v_x(t_k)]^T$, where $(r_x(t_k), r_y(t_k))$ denotes the mobile position at time $t_k$ and $v_x(t_k), v_x(t_k)$ its velocity. As mentioned previously, the position is supposed to be one of the map grid point. If we denote by $Nx$ and $Ny$ respectively the cardinalities of points gridding the map in both axis, the cardinality of the position set is $N_x N_y$. For the example of figure (1), we have $N_x = N_y = 16$. The velocity vector $[v_x(t_k), v_y(t_k)]^T$ takes only 8 different values in the set $\{-\|V(t_k)\|, 0, \|V(t_k)\|\} \times \{-\|V(t_k)\|, 0, \|V(t_k)\|\} \backslash \{0\} \times \{0\}$, where $\|V(t_k)\|$ is velocity modulus at time $t_k$ [1]. The position denoted $\boldsymbol{y}(t_k)$ is depending from state vector via the following (observing) relation $\boldsymbol{y}(t_k) \triangleq \boldsymbol{B}\boldsymbol{x}(t_k)$, where $\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. In order to simplify notations, we will use for the sequel the underscript $k$ to design time index $t_k$; for example $\boldsymbol{x}_k$ stands for $\boldsymbol{x}(t_k)$.

### 2.1 Markov decision process framework

The underlying assumption is that the optimal path must be a realization of a Markov chain driven at each time $k$ by the optimal decision which maximizes, over a finite time horizon, the expectation of a given cost function.

The general aim of the MDP is to determine the sequence of decisions $\{\boldsymbol{d}_k^\star(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)\}$, $k = 1, \ldots, K$ which maximizes a cost function related to the state sequence of the Markov chain. $\boldsymbol{S}_k$ denotes the position space of the path at time $k$. Assuming now that the state $\boldsymbol{x}_{k-1}$ is equal to a realization $j$, $j \in \boldsymbol{S}_{k-1}$ and the decision $\boldsymbol{d}_{k-1} = d$, $d \in \boldsymbol{D}_{k-1}(i)$ is taken. Realizations $j$ or $i$ of the Markov chain represent particular couple $(r_{k,x}, r_{k,y})$. The MDP is characterized with:

1. the realization $j$ of the state $x_k$ drawn according to $p_{ji}(d) \triangleq \Pr(\boldsymbol{x}_k = i | \boldsymbol{x}_{k-1} = j, \boldsymbol{d}_{k-1} = d)$ and
2. the cost of the transition from state $j$ to state $i$ under decision $d$ is denoted $c_{ji}(d)$.

Using the Bellman principle of optimality, the MDP problem may be solved by the following recursions [8, 6]:

$$\boldsymbol{J}_k^\star(i) = \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ \sum_{j \in \boldsymbol{S}_{k-1}} \left[ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right] p_{ji}(d) \right\}$$

$$\boldsymbol{d}_k^\star(i) = \arg \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ \sum_{j \in \boldsymbol{S}_{k-1}} \left[ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right] p_{ji}(d) \right\}, \tag{1}$$

$k = 2, \ldots, K$, $i \in \boldsymbol{S}_k$. The sequence $\{\boldsymbol{d}_k^\star(i)\}$, $k = 1, \ldots, K$, $i \in \boldsymbol{S}_k$ represents the *decision plan*. This plan can be interpreted as the sequence of decisions which *globally* maximizes the cost function. However it may happens that it is impossible to find a path starting at a desired point and ending at a specific point and satisfying specific constraints (e.g. length, maneuvers, etc.).

Due to operational considerations, we assume for the sequel that for each decision $\boldsymbol{d}_k$ we can associate only one ending state, *i.e.* each decision $\boldsymbol{d}_k$ represents a deterministic basic move from $j$ to $i$. Equation (1) is simplified as follows:

$$\begin{aligned} \boldsymbol{J}_k^\star(i) &= \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right\} \\ \boldsymbol{d}_k^\star(i) &= \arg \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right\}, \end{aligned} \tag{2}$$

[1] Corresponding to the vertices and the axis lengths of a square centered at the origin

$k = 2, \dots, K$, $i \in \boldsymbol{S}_k$.

We can also remark that decision $\boldsymbol{d}_k^\star(i)$ is taken in the set $\boldsymbol{D}_{k-1}(i)$, independently of the decision $\boldsymbol{d}_{k-1}^\star(j) \in \boldsymbol{D}_{k-2}(j)$, $j \in \boldsymbol{S}_{k-1}$ previously estimated. The last remark is quite important, for instance for long time horizon $k$. In this case, it may be possible to obtain a "bang-bang" decision plan between $k$, $k+1$ and $k+2$, *i.e.* going from state $j$ to state $i$ and then returning in state $j$ (see figure (4)). This possibly occurs if $c_{ji}(d)$ and $c_{ij}(d')$, $d \in \boldsymbol{D}_{k-1}(i)$, $d' \in \boldsymbol{D}_k(j)$ are important compared to other costs in different state area. In this occurrence, maneuvering constraints are generally violated.

# 3 A modified MDP algorithm for TAN

In order to render decisions $\boldsymbol{d}_k^\star(i)$ dependent of the previous decisions taken at time $k-1$, the following modification in the MDP algorithm recursion is proposed:

$$\begin{aligned} \boldsymbol{J}_k^\star(i) &= \max_{d \in \boldsymbol{D}_{k-1}(i) \cap \boldsymbol{D}_{k-2}(j^\star)} \left\{ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right\} \\ \boldsymbol{d}_k^\star(i) &= \arg \max_{d \in \boldsymbol{D}_{k-1}(i) \cap \boldsymbol{D}_{k-2}(j^\star)} \left\{ c_{ji}(d) + \boldsymbol{J}_{k-1}^\star(j) \right\}, \end{aligned} \quad (3)$$

$j^\star \triangleq \boldsymbol{d}_{k-1}^\star(i), k = 2, \dots, K$, $i \in \boldsymbol{S}_k$.

In practice, equation (3) can be reformulated as follows:

$$\begin{aligned} \boldsymbol{J}_k^\star(i) &= \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ c_{ji}(d)\delta(d,d') + \boldsymbol{J}_{k-1}^\star(j) \right\} \\ \boldsymbol{d}_k^\star(i) &= \arg \max_{d \in \boldsymbol{D}_{k-1}(i)} \left\{ c_{ji}(d)\delta(d,d') + \boldsymbol{J}_{k-1}^\star(j) \right\}, \end{aligned} \quad (4)$$

$k = 2, \dots, K$, $i \in \boldsymbol{S}_k$ and $d' = \boldsymbol{D}_{k-2}(j^\star)$. $\delta(d,d')$ is a matrix defining the authorized decisions whom can be taken at time $k-1$ versus the previously decisions taken at time $k-2$.

## 3.1 A practical issue for the modified MDP algorithm: maximization of the local energy

For a concrete issue of the modified MDP algorithm, we must define four following quantities:

1. the set of the state $\boldsymbol{S}_k$,
2. the set of decisions $\boldsymbol{D}_{k-1}(i)$, *i.e.* the catalog of the elementary moves associated with the state $i$,
3. the authorized transition decision matrix $\delta(d,d')$, $d \in \boldsymbol{D}_{k-1}(i), d' \in \boldsymbol{D}_{k-2}(j^\star)$ and
4. the cost of each basic moves $c_{ji}(d)$.

The set $\boldsymbol{S}_k$ is assumed to be independent of time index, *i.e.* $\boldsymbol{S}_k = \boldsymbol{S}$. The set of decisions $\boldsymbol{D}_{k-1}(i)$ is also assumed time independent, *i.e.* $\boldsymbol{D}_{k-1}(i) = \boldsymbol{D}(i)$ and defined by the 8 adjacents neighbor states $j$ whom conduct to the state $i$. If state $i$ do not have any ancestor $j \in \boldsymbol{S}$ *-e.g.* border case of the map for example- the associated decision is 0. Then $\boldsymbol{D}(i) = \boldsymbol{D} = \{0, 1, \dots, 8\}$. The total number of legs, which grid the map is $R = 8(N_x-2)(N_y-2)+10\left((N_x-2)+(N_y-2)\right)+12$. The top of the figure (2) shows the $N_xN_y$ states and the $R$ admissible decisions where $N_x = N_y = 3$ for this example.

The cost function must take into account the information variation between two adjacent states. In [9, 10], the mutual information is chosen to define the local cost function. Another possibility is to compute the entropy of the time series collected between state $j$ and $i$. However, we chose to compute the local energy of the time series collected from $j$ to $i$.

### 3.1.1 A local measurement of the energy as cost function

Each legs for all admissible decision between state $j$ to state $i$ is regularly discretized in $P$ points; data are retrieved by a bilinear interpolation from the map and stacked in the matrix $\boldsymbol{Z}$ $(P \times R)$ as showed in the middle of figure (2).

For each column $r = 1, \dots, R$ of $\boldsymbol{Z}$, the local energy denoted $E_r$ is computed as follows.

$$E_r = \sqrt[\alpha]{\sum_{i=1}^{P-1} \left(\boldsymbol{Z}(i+1,r) - \boldsymbol{Z}(i,r)\right)^2}, \quad (5)$$

where $\alpha$ is given by

$$\alpha = 2\log\left(\frac{\max_{r=1,\dots,R}\{\sum_{i=1}^{P-1}\left(\boldsymbol{Z}(i+1,r) - \boldsymbol{Z}(i,r)\right)^2\}}{\min_{r=1,\dots,R}\{\sum_{i=1}^{P-1}\left(\boldsymbol{Z}(i+1,r) - \boldsymbol{Z}(i,r)\right)^2}\right) \quad (6)$$

Bottom of the figure (2) shows values of $E_r$, $r = 1, \dots, \boldsymbol{R}$ and $k = 1$. As expected, we can see that the local energy is proportional to the local variation of the map.

### 3.1.2 An efficient way to reduce memory burden and time computations

Assuming that for each decision $\boldsymbol{D}(i)$, a unique state $j$ can be associated with $i$, an original data structure can be deduced in order to improve algorithm performance. Two matrices must be computed *a priori*, says

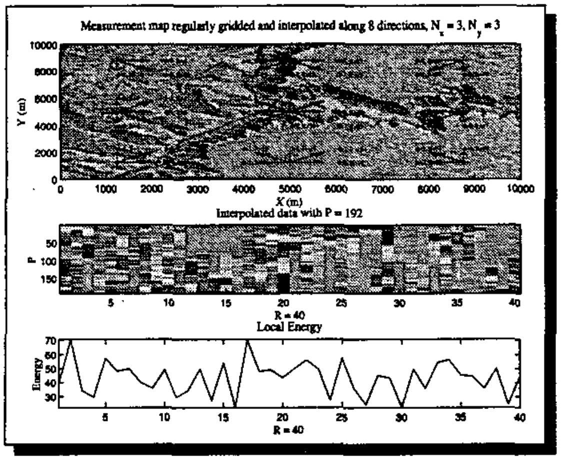

Figure 2: Local energy associated with each leg.

| $d \in D \backslash i \in S$ | $i = 1$ | $\cdots$ | $i = N_xN_y$ |
|---|---|---|---|
| $d = 1$ | $j(i,d)$ | $\cdots$ | $j(i,d)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $d = 8$ | $j(i,d)$ | $\cdots$ | $j(i,d)$ |

Table 1: Matrix $\boldsymbol{I}$ of elements $j(i,d) \in \boldsymbol{S}_{k-1}$. This matrix represents all admissible previous states $j$ at time $k-1$.

| $d \in D \backslash i \in S$ | $i = 1$ | $\cdots$ | $i = N_xN_y$ |
|---|---|---|---|
| $d = 1$ | $c_{ji}(d)$ | $\cdots$ | $c_{ji}(d)$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $d = 8$ | $c_{ji}(d)$ | $\cdots$ | $c_{ji}(d)$ |

Table 2: Matrix $\boldsymbol{C} = \{c_{ji}(d)\}$ associated with the matrix $\boldsymbol{I}$.

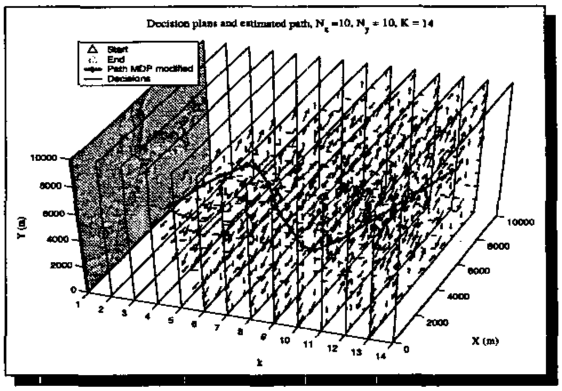

Figure 3: Recursions of the modified MDP algorithm. The corresponding estimated path is displayed.

$\boldsymbol{I}$ and $\boldsymbol{C}$, both are $(8 \times N_xN_y)$. In the first hand, the matrix $\boldsymbol{I}$ represents for all $i \in \boldsymbol{S}$, the admissible previous states $j \in \boldsymbol{S}$ associated with the decisions $d \in \boldsymbol{D}$. $(j(i,d) = 1, \ldots, N_xN_y\ )$ if state $j$ is admissible or 0 if not. In the second hand, the $\boldsymbol{C}$ matrix components are the local energy costs computed with relation (5). If the state $j(i,d)$ is not admissible, *i.e.* $j(i,d) = 0$, the cost is assumed to be null. Tables (1,2) show the structure of the two matrices. A third matrix denotes $\boldsymbol{\delta}$, $(8 \times 8)$ is also required and defined by the following equation:

$$\boldsymbol{\delta} = \left(\begin{array}{c|cccccccc} d\backslash d' & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right) \tag{7}$$

This matrix reflects the authorized decision $d$ at time $k-1$ allowed versus decisions $d'$ at time $k-2$. If $\boldsymbol{\delta}(d,d') = 1$, $\forall\{d,d'\} \in \boldsymbol{D}^2$, we retrieve the classic MDP algorithm. The table (3) shows the pseudo-code of the modified MDP algorithm for TAN applications.

This new algorithm can be easily extended; for example:

- if start/end area are no more one point of the grid;
- to discard states associated with unauthorized area of the map;
- to optimize with more realistic basic moves given by a maneuverer's catalog defined by operator.

The last three points represent the main contribution of this algorithm.

An example is presented in figures (3,4,5). The two first one display decision plans versus time of the modified and the classic MDP algorithm respectively, with $N_x = N_y = 10$ and $K = 14$. Corresponding optimized mobile trajectories are also plotted on the map. In the third one (fig. 5) optimized paths are compared on a real situation. We can see that maneuvering constraints are now satisfied by using modified MDP.

```
Given $i_{start}, i_{end} \in S$

1) Initialization

$J_1(i_{start}) = c_1 > 0$
$J_1(i) = 0,\ i \in S \backslash \{i_{start}\}$
$d_1(i) = 0,\ i \in S$

2) Recursion

For $k = 2, \dots, K$ and $i = 1, \dots, S$
a)
$V_{k-1} = \{J_{k-1}(s) \neq 0,\ s \in S\}$
$T_{k-1} = \{d_{k-1}(r),\ r \in V_{k-1}\}$
b) $j \in V_{k-1}, t \in T_{k-1}$
  $J_k(i) = \max_{d \in D_{k-1}(i)} \{C_{ji}(d)\delta(d,t) + J_{k-1}(j)\}$
  $d_k^*(i) = \arg \max_{d \in D_{k-1}(i)} \{C_{ji}(d)\delta(d,t) + J_{k-1}(j)\}$
c) $V_k = \{J_k(s) = 0,\ s \in S\}$
d) $d_k^*(s) = 0,\ s \in V_k$
End for

3) Back-tracking

$i_K^* = i_{end}$
$d_K^* = d_K(i_K^*)$

For $k = K-1, \dots, 1$
$i_k^* = I\left(d_{k+1}^*, i_{k+1}^*\right)$
$d_k^* = d_{k+1}(i_k^*)$
End for
```

Table 3: Modified MDP algorithm for TAN.

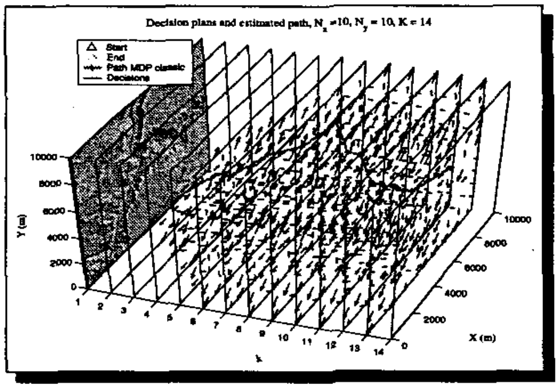

Figure 4: Recursions of the classic MDP algorithm. The corresponding estimated path is displayed.

## 4 Overview of the Posterior Cramér-Rao Bounds

In the previous section, an original algorithm has been described which aim is to plan the mobile path maximizing an additive auxiliary cost function based on the local energy. However, our initial objective is to improve the estimation of the mobile position which may be achieved by using various methods, e.g. particle filtering. Performance of all these algorithms can be compared with the PCRB associated with the dynamic system and a given nominal trajectory. This path can be the output of the previous optimization step.

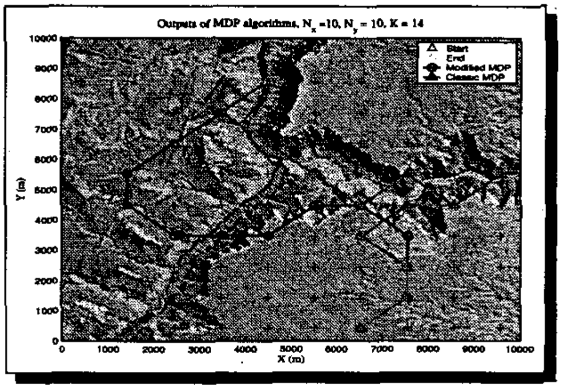

Figure 5: Outputs of Classic/Modified MDP algorithms.

We assume TAN systems are modeled with the following non-linear dynamic system:

$$\begin{cases} x_1 & = & w_1 \\ x_{k+1} & = & \Phi_k x_k + u_k + w_k \\ z_k & = & H_k(x_k) + v_k. \end{cases} \tag{8}$$

State $x_k$ dimension is $d = 4$, measurement $z_k$ dimension is denoted $m$. $\Phi_k$ represents the transition matrix of the linear state equation and $H_k(x_k)$ is the observation matrix. This last one is nothing else the measurement available *via* the reference map (possibly vectorial). The term $u_k$ is the system control (mobile maneuvers). For our case, the control sequence $U_{1:k} \triangleq \{u_1, \dots, u_k\}$ is deduced from the output sequence $I_{1:k}^* \triangleq \{i_1^*, \dots, i_k^*\}$ of the optimization step. Assuming that velocity modulus is constant from one leg to another and a constant sampling interval $T = (t_{k+1} - t_k)$, $\forall k = 1, \dots, K$ the law $u_k$ is computed as follows:

$$u_k = \begin{bmatrix} 0 \\ v_{k+1,x}^* - v_{k,x}^* \\ 0 \\ v_{k+1,y}^* - v_{k,y}^*, \end{bmatrix} \tag{9}$$

where $v_{k,x}^* \; = \; \|V_k\| \begin{bmatrix} \cos\left(\arctan\left(\frac{r_{k,y}^*}{r_{k,x}^*}\right)\right) \\ \sin\left(\arctan\left(\frac{r_{k,y}^*}{r_{k,x}^*}\right)\right) \end{bmatrix}$ and

$$\|V_k\| = \frac{\sqrt{\left(r_{k,x+1}^* - r_{k,x}^*\right)^2 + \left(r_{k,y+1}^* - r_{k,y}^*\right)^2}}{T}.$$

We denote the state sequence collected from departure to time $k$ by $\boldsymbol{X}_{1:k} \triangleq \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k\}$ and the associated measurement sequence by $\boldsymbol{Z}_{1:k} \triangleq \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_k\}$.

In equation (8), $\boldsymbol{w}_1$, $\boldsymbol{w}_k$ and $\boldsymbol{v}_1$ are respectively initial, current state process noise and the measurement noise process.

### 4.1 Posterior Cramér-Rao bounds

The PCRB measures the minimum mean square errors of any unbiased estimator when both measurement $\boldsymbol{Z}$ and states $\boldsymbol{X}$ are two random processes. This approach departs from of the classical Cramér-Rao where the state is considered unknown but deterministic [11, 12]. The PCRB is evaluated as the inverse of the Fisher information matrix (FIM) denoted $\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{Z})$. For filtering case modeled by equation (8), the FIM is computed for every time increment and denoted $\boldsymbol{F}_k(\boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k}, \boldsymbol{U}_{1:k})$. As $\boldsymbol{U}_{1:k}$ is deterministic (constructed according to optimization outputs), the FIM ( associated to a nominal trajectory) simply reads $\boldsymbol{F}_k(\boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k})$.

The FIM $(kd \times kd)$ is defined as follows:

$$\begin{aligned} \boldsymbol{F}_k(\boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k}) &\triangleq -E\left\{\Delta_{\boldsymbol{X}_{1:k}}^{\boldsymbol{X}_{1:k}} \left[\ln\left(p(\boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k})\right)\right]\right\} \\ &= \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \\ \boldsymbol{B}_k^T & \boldsymbol{P}_k \end{bmatrix}, \end{aligned} \tag{10}$$

where the *Laplacian* operator $\Delta_{\psi}^{\boldsymbol{\theta}}$ $(d \times d)$ is defined by

$$\Delta_{\psi}^{\boldsymbol{\theta}} \triangleq \nabla_{\psi} \nabla_{\boldsymbol{\theta}}^T = \begin{bmatrix} \dfrac{\partial^2}{\partial\psi_1\partial\theta_1} & \cdots & \dfrac{\partial^2}{\partial\psi_1\partial\theta_d} \\ \vdots & \cdots & \vdots \\ \dfrac{\partial^2}{\partial\psi_d\partial\theta_1} & \cdots & \dfrac{\partial^2}{\partial\psi_d\partial\theta_d} \end{bmatrix}. \tag{11}$$

The PCRB denoted $\boldsymbol{P}_k$ $(d \times d)$ is the right lower block matrix of the matrix $\boldsymbol{F}_k(\boldsymbol{X}_{1:k}, \boldsymbol{Z}_{1:k})$. Using the Schur block inversion matrix lemma [13], an elegant recursive expression of the PCRB can be derived (see [4, 5]):

$$\boldsymbol{P}_{k+1} = \boldsymbol{\Pi}_{k+1} + \boldsymbol{D}_k^{22} - \boldsymbol{D}_k^{21}\left[\boldsymbol{D}_k^{11} + \boldsymbol{P}_k\right]^{-1}\boldsymbol{D}_k^{12}, \tag{12}$$

with

$$\begin{cases} \boldsymbol{D}_k^{11} & \triangleq & E_{\boldsymbol{x}_k}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\left[\ln\left(p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right)\right]\right\} \\ \boldsymbol{D}_k^{12} & \triangleq & E_{\boldsymbol{x}_k}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_{k+1}}\left[\ln\left(p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right)\right]\right\} \\ \boldsymbol{D}_k^{21} & \triangleq & E_{\boldsymbol{x}_k}\left\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_k}\left[\ln\left(p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right)\right]\right\} = \boldsymbol{D}_{12}^T \\ \boldsymbol{D}_k^{22} & \triangleq & E_{\boldsymbol{x}_k}\left\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}}\left[\ln\left(p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\right)\right]\right\} \\ \boldsymbol{\Pi}_{k+1} & \triangleq & E_{\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}}\left\{-\Delta_{\boldsymbol{x}_{k+1}}^{\boldsymbol{x}_{k+1}}\left[\ln\left(p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1})\right)\right]\right\} \end{cases} \tag{13}$$

For the sequel, we particularize the last equation according to the equation (8) and the two following assumptions:

1. constant sampling interval denotes $T = (t_{k+1} - t_k), \forall k = 1, \dots, K$, so that $\boldsymbol{\Phi}_k = \boldsymbol{\Phi}$
2. zero mean Gaussian processes for $\boldsymbol{w}_1$, $\boldsymbol{w}_k$, $k = 1, \dots, K-1$ and $\boldsymbol{v}_k$ for noise processes. Their covariance matrices are respectively $\boldsymbol{Q}_1$, $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$. $\boldsymbol{R}_k$ is supposed to be dependant of $\boldsymbol{y}_k$.

Equation (13) is simplified as follows:

$$\begin{cases} \boldsymbol{D}_k^{11} & = & \boldsymbol{\Phi}^T\boldsymbol{Q}_k^{-1}\boldsymbol{\Phi} \\ \boldsymbol{D}_k^{12} & = & -\boldsymbol{\Phi}^T\boldsymbol{Q}_k^{-1} \\ \boldsymbol{D}_k^{21} & = & -\boldsymbol{Q}_k^{-1}\boldsymbol{\Phi} \\ \boldsymbol{D}_k^{22} & = & \boldsymbol{Q}_k^{-1}, \end{cases} \tag{14}$$

and

$$\boldsymbol{\Pi}_{k+1} = E_{\boldsymbol{x}_{k+1}}\left\{\nabla_{\boldsymbol{x}_{k+1}}\left[\boldsymbol{H}^T(\boldsymbol{x}_{k+1})\right]\boldsymbol{R}_{k+1}^{-1}\nabla_{\boldsymbol{x}_{k+1}}^T\left[\boldsymbol{H}^T(\boldsymbol{x}_{k+1})\right]\right\} \tag{15}$$

#### 4.1.1 Evaluation of $\Pi_{k+1}$ for TAN applications

The main difficulty in the computation of the PCRB for TAN applications comes from the evaluation of $\boldsymbol{\Pi}_{k+1}$. For the daily case, given a reference map and eventually a measurement covariance error map, $\boldsymbol{\Pi}_{k+1}$ must be evaluated numerically by intensive Monte-Carlo runs. Notice also measurements $\boldsymbol{z}_k$ and measurement covariance matrices $\boldsymbol{R}_k$ depend only of the current position *i.e.* $\boldsymbol{y}_k$. $\boldsymbol{\Pi}_{k+1}$ is evaluated in three steps for every time increment $k$:

1. Generate $M$ samples says $\boldsymbol{x}_{k+1}^i$ according to the equation (8). At $k = 1$, the first $M$ samples $\boldsymbol{x}_1^i$ are drawn according $\boldsymbol{Q}_1$. Out-ranging samples are eliminated in the evaluation of $\boldsymbol{\Pi}_{k+1}$.

2. For each previous samples, compute covariance matrices $R^i_{k+1}$ (using bilinear interpolation).
3. Evaluate numerically, for each sample, the local gradient $\nabla_{x_{k+1}}\left[H^T\left(x^i_{k+1}\right)\right]$ with the step $[h_x, h_y]$.

The PCRB recursion of the equation (12) is simplified as:

$$\left\{\begin{array}{lcl} P_1 & = & Q_1^{-1} \\ P_{k+1} & = & \Pi_{k+1} + \left[\Phi P_k^{-1}\Phi^T + Q_k\right]^{-1}, \end{array}\right. \quad (16)$$

$k = 1, \dots, K-1$. Computation of equation (16) will permit us to judge the quality of the *a priori* optimization algorithm based on the local energy cost function.

A direct globally optimization of (16) can't be realized either with a dynamic programming point of view or branch & bound algorithm. The reason is there is not any principle of comparaison available for (16), *i.e.* if $f(A) \succcurlyeq f(B)$ and $C \succcurlyeq 0 \nRightarrow f(A+C) \succcurlyeq f(B+C)$.

## 4.2 Example of planification

In the following example, results of the optimization procedure for the two MDP algorithms are presented. The covariance matrix of the Gaussian state noise process is defined by (see [14, 6] for details)

$$Q_k = \sigma^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix}, \quad (17)$$

where $\sigma^2 = 1000\ m/h$ and $T = 0.1\ h$. The initial covariance state is

$$Q_1 = \begin{bmatrix} \sigma^2_{r,x} & 0 & 0 & 0 \\ 0 & \sigma^2_{v,x} & 0 & 0 \\ 0 & 0 & \sigma^2_{r,y} & 0 \\ 0 & 0 & 0 & \sigma^2_{v,x} \end{bmatrix}, \quad (18)$$

where $\sigma_{r,x} = \sigma_{r,y} = 500\ m$, $\sigma_{v,x} = \sigma_{v,y} = 0.01\ m/h$.

The covariance of the measurement process is supposed to be uniform on the available map and equal to $R_k = 40\%$ of the total standard variation of the map. $N_x = N_y = 15$ and $K = 18$ for the simulations. For the computation of the PCRB, $M = 500$ Monte-Carlo trajectories have been generated. In order to compare quality of our optimization, a random path has been also generated. The figure (6) shows the three paths. The initial confidence ellipsoid is also displayed.

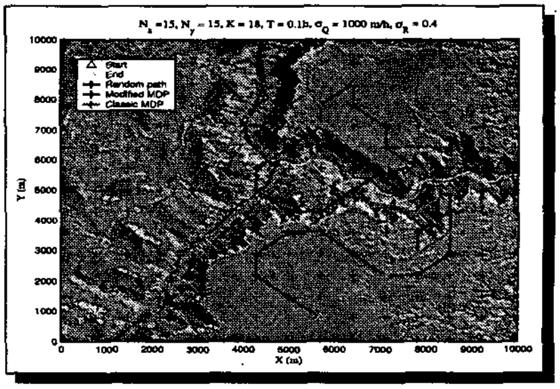

Figure 6: A random path and outputs of the modified/classic MDP algorithms.

The figure (7) shows the PCRB computed on the three paths. Upon figure is the PCRB on the position $y_k$ and bottom figure is the PCRB of the velocity. The position's PCRB of the random path is increased when the path is located near a flat area. In only few legs, the position PCRB collapses from 500 $m$ to 10 $cm$ of precision. This is a piece of evidence how the measurement can dramatically improve the precision of the mobile position, even if this value (10 $cm$) seems to be the asymptotic precision for this given scenario. If operators desire a quickly accurate localization, the optimized path should converge as soon as possible to the ultimate value.

As expected, position PCRB of the optimized paths are very stable around the asymptotic PCRB. The figure (8) shows the trace of $F_k^{-1}$ divided by the cumulative distance versus time $k$. The function can be interpreted as the optimization efficiency during the path. The last ratio measures how fast the PCRB diminish versus the distance already covered since departure.

We can see for both optimized paths that the optimization efficiency is very low compared to the random path. This means optimized paths improved the localization of the mobile since departure.

# 5 Perspectives and conclusions

This paper presents an original method based on a MDP algorithm to solve scheduling path problem for TAN systems. The main contribution of this the algorithm is to integrate very easily maneuverer contraints and authorized basic moves. The optimization is based on the maximization of a local energy function which measures the terrain elevation variation's. Performance of the optimization algorithm is evaluated by means of the PCRB and validates our approach. However, direct and global maximization of the PCRB

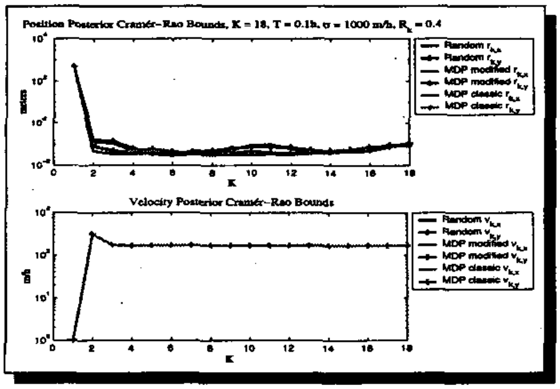

Figure 7: Posterior Cramér-Rao bounds along the three trajectories.

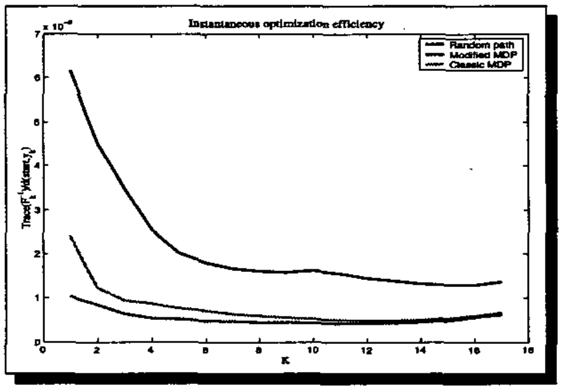

Figure 8: Instantaneous efficiency of the optimization.

seems unfeasible by means of "standard" control algorithms, this is the reason why suboptimal algorithms have been developed instead. Future directions in this work concern the use of *simulation* methods for jointly approximating the PCRB and the best sequence of controls.

## References

[1] N. Bergman and P. Tichavsky, "Two cramér-rao bounds for terrain-aided navigation," *Submitted to IEEE Transactions on Aerospace And Electronic Systems.*

[2] R. Enns and D. Morrell, "Terrain-aided navigation using the viterbi algorithm," *journal of guidance, control, and dynamics*, vol. 18, no. 8, November-December 1995.

[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing Special issue on Monte-Carlo methods for statiscal signal processing.*

[4] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior cramér-rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 46, no. 5, May 1998.

[5] M. Šimandl, J. Královec, and P. Tichavský, "Filtering, predictive, and smoothing cramér-rao bounds for discrete-time nonlinear dynamic systems," *Automatica*, vol. 37, 2001.

[6] J.-P. Le Cadre and O. Trémois, "Bearings-only tracking for maneuvering sources," *IEEE Transactions on Aerospace And Electronic Systems*, vol. 34, no. 1, January 1998.

[7] O. Trémois and J.-P. Le Cadre, "Optimal observer trajectory in bearings-only tracking," *IEE Proc.-Radar, Sonar Navig.*, vol. 146, no. 1, February 1999.

[8] D. P. Bertsekas, *Dynamic programming: Deterministic and stochastic models.* Prentice Hall, 1987.

[9] A. Logothetis, A. Isaksson, and R. J. Evans, "An information theoretic approach to observer path design for bearings-only tracking," *Proceedings of the 36th conference on decisions & control, san diago, California USA*, December 1997.

[10] A. Logothetis, A. Isaksson, and R. J. Evans, "Comparaison of suboptimal strategies for optimal own-ship maneuvers in bearings-only tracking," *Proceedings of the 36th conference on decisions & control, philadelphia, Pennsylvania USA*, June 1998.

[11] B. Z. Bobrovsky, E. Mayer-Wolf, and M. Zakai, "Some classes of global cramér-rao bounds," *The annals of Statistics*, vol. 15, no. 4, 1987.

[12] P. E. Ferreira, "Extending fisher's measure of information," *Biometrika*, vol. 68, no. 3, 1981.

[13] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *SIAM*, vol. 23, no. 1, January 1981.

[14] X. Rong Li and V. P. Jilkov, "A survey of maneuvering target tracking: dynamic models," *Proceeding of SPIE conference on signal and data processing of small targets, Orlando, FL, USA*, 2000.