Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Robust trajectory for mobile robot

A. Pruski & S. Rohmer
LAEI, University of Metz, 57045 Metz cedex 1, France

ABSTRACT

This paper deals with the problem of mobile path planning. We investigate the case for which the obstacle positions are well known but the mobile location is defined with uncertainties. We propose an algorithm which finds a robust path, if it exists, from a source point to a goal point for a non holonomic robot. The proposed method is based on the Multivalue Coding model.

INTRODUCTION

Mobile robot path planning is a problem that is investigated by numerous research teams. The solutions proposed consider the problem either as purely geometrical [1][2][3], or include the mobile kinematic characteristics [4][5] or take into account the mobile [6] or the obstacles[7] location knowledge uncertainties. Most path planners consider that the mobile and obstacle uncertainties are taken into account by a navigation module that uses the environment and position sensor information in order either to fit theoretical trajectory or to perform an obstacle avoidance. Manoeuvers are performed in both cases. The disadvantage lies in the computing cost and the frequent sensor activation. Our work is focused on automatic path planning for wheelchairs for the disabled. It is necessary that the proposed trajectory should be robust and that the probability to reach the goal should be the highest possible. Such a trajectory avoids permanent location computing, limits associated sensor treatment and allows a comfortable and smooth motion for the person seated on the mobile. This paper deals with an automatic path planning for mobile robot that takes the location uncertainties into account.

THE PROBLEM STATEMENT

Let $\Omega \subset R^2$ a closed set of polygons representing the environment and $\Gamma \subset R^2$ the set of the positions of a rigid object A such as $\Omega \cap \Gamma = \emptyset$. The set Γ is composed by the element Xi defined by the triplet $Xi = (xi, yi, \Theta i)$ where xi and yi represent the A reference point position and Θi the orientation of a vector crossing the reference point. The problem consists in finding a list Π of the Xi element such as

1- $X0=(x0,y0,\Theta 0)$ the source point;
2- $Xf = (xf, yf, \Theta f)$ the final point;
3- $(x_{i-1} - x_i)\, tg\Theta_i - (y_{i-1} - y_i) = 0$;
4- $\Theta_{i+1} - \Theta_i = \beta \quad \beta \in [0, 2\pi]$;

The two last expressions represent the non holonomy kinematic constraint for no constraint on the Θ variation

5- P(Xi) represents the probability distribution associated to Xi. We consider that P(Xi) is a multivariate gaussian distribution expressed as:

$$P(Xi) = \frac{1}{(2\pi\, det(C))^{1/2})^{n}} \exp(-1/2\, (Xi-X)^{t} C^{-1} (Xi-X)) \qquad (1)$$

with C the covariance matrix, X the mean vector , Xi the A position vector.

6-Let δi an equiprobability ellipsoid associated to the Xi point and let Δ the set of the ellipsoids associated to the path Π

The problem consists in finding a class of paths Π^* that minimize the function $f(d,R)$ with d the distance and R a risk function.

Definition1 Let two ellipsoids $\varepsilon 1$ and $\varepsilon 2$ with $\varepsilon 2 \subset \varepsilon 1$ with the same center and the same axis orientation, we define the risk function R by $M\varepsilon 2/M\varepsilon 1$ with $M\varepsilon 2$ and $M\varepsilon 1$ the length of the major axis of respectively $M\varepsilon 1$ and $M\varepsilon 2$.

BACKGROUND AND OVERVIEW.

Let a rigid object A defined in R^n and Ω a closed set defined on R^m such as ($m =< n$), Γ the set of the R^n regions such as $\Gamma^* = Proj(\Gamma) \rightarrow R^n$ and $\Gamma^* \cap \Omega = \emptyset$. Γ is called the space configuration. Numerous authors [2][3] have used the Minkowski difference in order to determine the space configuration. In our case , the object A is a mobile robot which we consider holonome in a first approach. Then n=3, two dimension for the position x,y and one dimension for the orientation Θ.

The path planning methods using graph techniques need an environment decomposition in discrete areas each of whose cell have particular characteristics. We propose to use rectangloids cells each of whose dimension represents an interval:

$$K=[xmin,xmax]x[ymin,ymax]x[\Theta min,\Theta max]$$

This representation environment was used by a number of authors under different forms. ZHU and LATOMBE [3] propose rectangloids whose intervals are the greatest possible and non overlapped. Faverjon [8] proposes octrees for which we write

$$K=[kx2^{u},(kx+1)2^{u}-1] \times [ky2^{u},(ky+1)2^{u}-1] \times [k\Theta 2^{u},(k\Theta+1)2^{u}-1]$$

with ki ϵ $[0,2^{n-u}-1]$ for a possible decomposition in 2^n cells in each dimension. This decomposition allows a tree representation which has interesting properties for path planning. However the dimension x,y and Θ are bounded by the term u which avoids overlapping but may be a hindrance in terms of cell number. We propose a representation allowing dimension independance in order to describe the free space by greater cells to minimize the memory space and to allow cell overlapping.

$$K=[kx2^u,(kx+1)2^u-1]x[ky2^v,(ky+1)2^v-1]x[k\Theta 2^q,(k\Theta+1)2^q-1]$$

On the other hand we propose the use of multivalue numbers (MVN) to represent each interval in each dimension. The MVN have two interesting advantages:

a- Dimension independance is guaranteed, allowing a greater choice of rectangloids than with octrees.

b-The MVN represents an interval under three forms:

- the tree representation;
- the set representation;
- the numerical representation.

Each form allows a particular treatment: the tree form allows to treat the connectivity relations, the set form treats the operations like unions, intersections, complements... of intervals, and the numerical representation allows to link the geometrical coordinates (position,orientation) to the cell representation. The two former representations do not cumulate these properties. The rectangloid representation [3] has a tree structure that only allows to define the connexity relations without taking into account the geometrical information. The octrees is a purely computer science representation whose relations between cells and numerical datas are described by analysing the position in the tree.

Detail informations on the properties, definitions, treatments and coding are described in [9][10].

Brief MVN overview

Definition 2 A MVN T is defined on R^q by a basis B such as

$$T=a_{n-1} B^{n-1}+... + a_0 B^0$$

with x $\in$ X; a_n $\in$ B^+ ; $B^+=\{\{x\},X\}$; X=[0,B-1]

we note that

$\forall$ i $\in$ [0,n]; Q = {ail ai =X} then Card(T) > 1 if Q\=$\emptyset$

for example B=5

T=1X2=> $(T)_{10}$ ={102,112,122,132,142};

Following this definition a MVN represents at the same time a set of numerical values and a multiple values number.

Definition 3 The continuous value MVN (CMVN) are a subset of the MVN such as

if a_0=X then a_j = X only if a_i=X with j=i+1

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Definition 4 A multidimensionnel MVN is a set of MVN. In a binary computer all numbers are represented in basis 2. The 2^+ basis representation needs three states. Another bit added to the two others is needed.

Definition 5 In a computer , a CMVN is represented by a set of two binary words called CODE and VALIDATION. CODE represents the ordered set of the elements ai with ai ∈ [0,1] and the VALIDATION associates to each element ai of CODE one bit defining either an explicit state ai ∈ [0,1] with the convention bi=1 or a non explicit state ai={0,1} with the convention bi=0.

For example

T = {101000,111100}= {CODE,VALIDATION}

T= {40,41,42,43};

Definition 6 A MultivalueCode Tree (MCT) is a tree structure representing a CMVN. Each element ai of T may be represented by a three type node:

- a A type node reprenting a X state;
- a B type node son of A representing a 1 state;
- a C type node son of A representing a 0 state.

A one dimension CMVN is represented by a set of nodes starting from a source node to a leaf node representing each explicite element ai from the left to the right.

Properties:

Card([T]= NOT (VALIDATION)

Min[T]= CODE AND VALIDATION

Max[T]= CODE OR NOT(VALIDATION)

Path planning

The environment is modeled by CMVN according to an algorithm which is developed in [11]. We note that the reference point is the center of the crossing the wheels. The path planning developed in detail in the same paper is based on the heuristic A* algorithm.

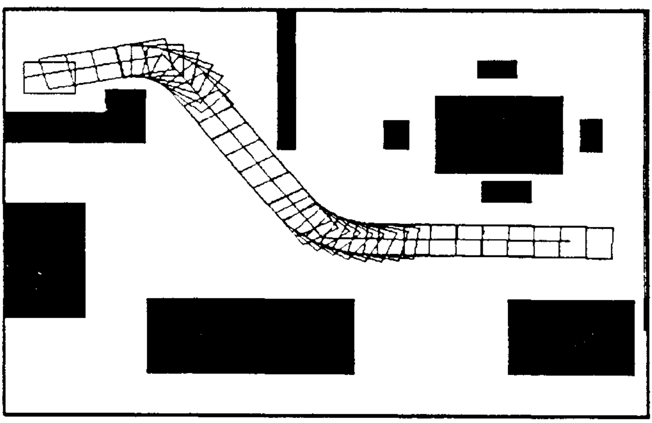

Figure 1: Path planning example

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

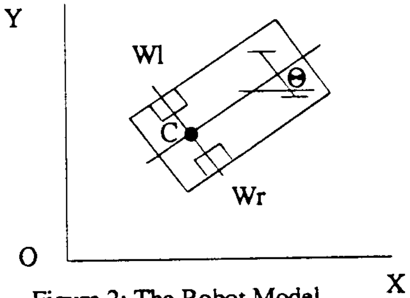

Figure 2: The Robot Model

The robot model

Our mobile robot is a two rear power wheel system as shown on figure 2. Our mobile robot has two degrees of freedom noted D for the displacement of point C and Θ for the rotation. We write that the C position is given by the relations

$$x_i = x_{i-1} + \Delta x \quad \text{and} \quad y_i = y_{i-1} + \Delta y.$$

$$[\Delta X] = (\Delta x \ \Delta y)^t = (D\cos\Theta \ \ D\sin\Theta)^t$$

$$\Delta[\Delta X] = J\,(\Delta D \ \Delta\Theta\,) = \begin{pmatrix} \cos\Theta & -D\sin\Theta \\ \sin\Theta & D\cos\Theta \end{pmatrix} \begin{pmatrix} \Delta D \\ \Delta\Theta \end{pmatrix}$$

The displacement ΔD and the orientation $\Delta\Theta$ are deduced from the odometric information Ul and Ur coming from the left and right wheels. We write that

$$\Delta D = (\Delta Ul + \Delta Ur)/2 \text{ and } \Delta\Theta = (\Delta Ul - \Delta Ur)/L$$

with L the distance between the wheels. The rotation sensors uncertainties, the errors accumulating and the slippage generate uncertainties on measures and thus on the position of the mobile. We consider, as some authors [12][13] that a random value εl and εr is superimposed on Ul and Ur. We suppose that εl and εr are independent, gaussian, have a zero mean value and that their variances are respectively σ_l^2 and σ_r^2

We deduce the covariance matrix with

$$\sigma_D^2 = (\sigma_l^2 + \sigma_r^2)/4\,,\ \sigma_\Theta^2 = (\sigma_l^2 + \sigma_r^2)/L^2\,,$$
$$\sigma_{D\Theta}^2 = (\sigma_l^2 - \sigma_r^2)/2L.$$

We consider that the variances σ_l^2 and σ_r^2 are equal that allow to write $\sigma_{D\Theta}^2 = 0$. We obtain the covariance matrix

$$\Delta C = \ = J \ \ E(\Delta D\ \Delta\Theta\,)^t \ \ J^t = J \begin{pmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_\Theta^2 \end{pmatrix} J^t$$

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Thus $C_i = C_{i-1} + \Delta C$

The mobile robot trajectory model

We have seen that the robot has two degrees of freedom: translation and rotation. We estimate that the mobile trajectory from a point $[X_{i-1}]$ to a point $[X_i]$ for a given probability is included in the track defined by the equiprobability ellipsoid from $[X_{i-1}]$ to $[X_i]$.
The ellipsoid equation [14] is given by

$$k^2 = (Xi-X)^t C^{-1} (Xi-X)$$

Lines T1 and T2 are the tangents to the ellipsoids from [Xi-1] to [Xi].

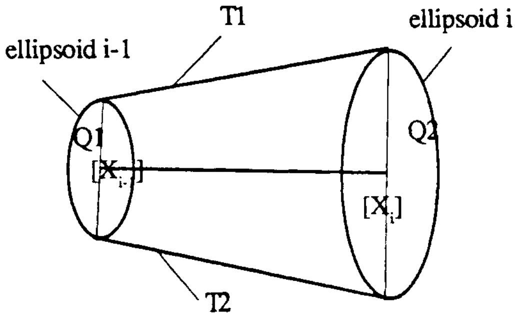

Figure 3: The equiprobability ellipsoïds

TRAJECTORY VALIDATION

We have considered above that the real trajectory for a given probability is contained in the track generated by the equiprobability ellipsoid displacement. We modelize that track by a polygon defined by the tangents T1 and T2 and by the lines Q1 and Q2 joining the extremities of T1 and T2. The track must be entirely included in the space configuration in order to make the displacement with a given probability possible. The verification is made in the following manner. First we extract the obstacle edges. These datas are soon available because they were determined when coding the space configuration [11]. Three cases occurs:

-case 1: there exists an intersection between an obstacle and one of limits T1 or T2;

-case 2: an intersection exists with the two limits T1 and T2;

- case 3: whole inclusion of an obstacle in the track.

Case 1

In that case, we must verify if T1 and T2 are located in the configuration space. The characteristics of lines T1 and T2 are known . We generate these lines with an interpolation algorithm of the Bresenham type in order to successively create each

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

point which represents an MVN of cardinality one. The points created belong the 3D space (x, y and Θ). The verification of the existence of the point in the configuration space is quickly executed on the MVC tree. If the point does not exist on the tree then we virtually slide on the obstacle toward the track axis until the obstacle disappears. The displacement is then reproduced on the axis and the point $[X_i]$ is replaced by P (figure 4). Then a new verification is started in order to verify the new trajectory with the new orientation in the configuration space. And then we generate the new track and verification of the consistence of the new lines T1 and T2 in the configuration space.

Case 2

The second case shown in figure 5 represents the crossing of a door theshold. A motion toward the track axis does not avoid a probable collision. We compute a risk

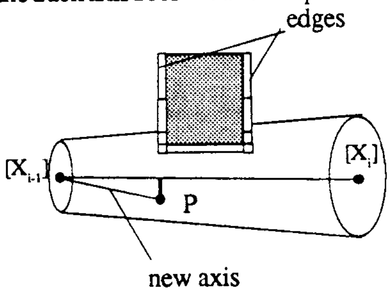

Figure 4: Case 1

coefficient R, according to the closest collision point to the axis. The MVN containing the point $[X_{i-1}]$ is then affected by the risk coefficient so that gi = gi R. with gi the cost computed by the A* algorithm to join the source node and the i node. We propose then to start the A* algorithm again while putting in the open list only the MVN belonging to the trajectory from the MVN (or nodes) including the source point to the MVN including the $[X_{i-1}]$ point. This procedure runs in two parts. First the A* algorithm determines a trajectory whose evaluation function only considers the distance cost. Then the track analysis either validates the trajectory or modifies the evaluation function values. This procedure may be executed in only one step by each developed node which must be validated by the track analysis. Such procedure would be time consuming.

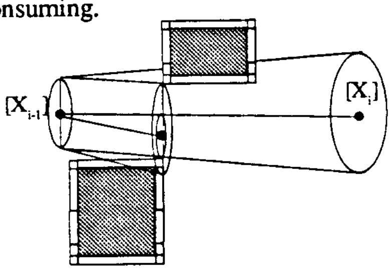

Figure 5: Case 2

Case 3

In the third case the obstacle is entirely included in the track. This statement is verified if all obstacle edges are located in the track. Then we activate the same procedure as in case 1.

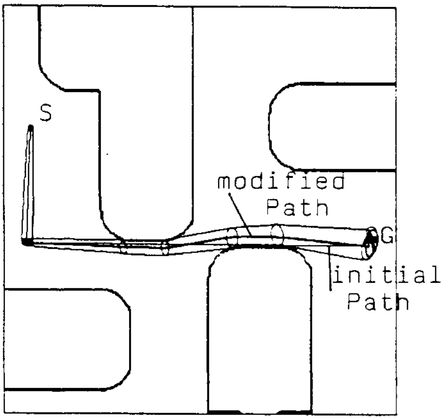

Figure 6: A Robust Path Example.

CONCLUSION

In this paper we describe a path planning method for a mobile robot. This method allows to define from the path planning stage a robust trajectory for which we consider the mobile location uncertainties. The obstacle positions are well known. Actually we have implemented on a PC-486 in C language, the algorithm dealing with the three cases. The running time depends of the number on obstacles in collision with the track. The computing time is of 1 to 2 seconds.

This robust trajectory has the advantage limiting (or avoiding) the computing related to the obstacles avoiding manoeuvers but also defining a probability area in which the mobile is located around the goal. Inversely, if the localisation precision of the mobile around the goal is imposed then it is possible to determine the precision needed at the origin of the trajectory.

In the actual work state we consider that the mobile can successively execute translations and rotations around the wheel axis center. This is a particular case of a non holonomic robot. The presented algorithm takes into account these characteristics by verifying if each point is in the configuration space as described in case 1. Our algorithm is a non resolution complete algorithm. Our actual aim consists in modifying the algorithm in order to find a solution each time , if it exists.

REFERENCES

1. Barraquand, J., Latombe, J.C. 'A monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom', pp 1712-1717, *Proc. of the IEEE Conf. on Robotics and Automation*, Cincinnati,1990.
2. Lozano-Perez, T. 'Spatial planning: a configuration approach', *IEEE Trans. on Computers*, Vol C-32, n°2, pp108-120, Feb. 1983,
3. Zhu, D., Latombe,J.C. 'Constraint reformulation in a hierarchical path planner, pp1918-1923,*Proc. of the IEEE Conf. on Robotics and Automation*, Cincinnati Ohio, 1990
4. Laumont, J.P. 'Feasible trajectories for mobile robots with kinematics and environment constraints, pp346-354, *Proc. of the Inter. Conf. on Autonomous Robots*, Amsterdam, 1986
5. Jacobs P., 'Canny J. Planning Smooth Paths for Mobile Robots', pp2-7, *Proc. of the IEEE Conf. on Robotics and Automation*, Scottsdale Arizona, 1989
6. Jacobs P, Canny J., 'Robust Motion Planning for Mobile Robots', pp 2-7 , *Proc. of the IEEE Conf. on Robotics and Automation*, Cincinnati, Ohio,1990
7. De Rougemont M., Diaz-Frias J.F., 'A theory of Robust Planning', *Proc. of the IEEE Conf. on Robotics and Automation*, Nice France, 1992
8. Faverjon B, 'Object Level Programming using an Octree in the Configuration Space of a Manipulator', pp 1406-1412, *Proc. of the IEEE Conf. on Robotics and Automation*, San Francisco Ca, 1986.
9. Pruski A. 'Multivalue coding for image and solid processing',*Revue Internationale de CFAO et d'Infographie*, Vol 6, N°2, pp 135-152, 1991
10. Pruski A., 'Multivalue Coding: Application to Autonomous Robots', *Robotica*, Vol 10, pp 125-133, 1992
11. Pruski A. Rohmer S. 'Multivalue Coding: Application to autonomous robot path planning with rotations', pp 694-699, *Proc. of the IEEE Conf. on Robotics and Automation*, Sacramento CA, 1991.
12. Crowley J.L., 'Asynchronous Control of Orientation and Displacement in a Robot Vehicle', pp 1277-1282, *Proc. of the IEEE Conf. on Robotics and Automation*, Scottsdale Arizona, 1989
13. Ming Wang C., 'Location estimation and uncertainty analysis for mobile robots', pp 1230-1235, *Proc. of the IEEE Conf. on Robotics and Automation*, 1988
14. Smith R.C., Cheesman P. 'On the representation and estimation of Spatial Uncertainties', *Int. Journal of Robotics Research*, 5(4) Winter, 1987.