Automation and Robotics in Construction XVI © 1999 by UC3M

FUZZY CONTROL BASED ON A REFLEX AVOIDANCE FOR A MOBILE ROBOT

K. BELMEKKI

Département d'électricité, ENSET d'Oran, BP 1523, Oran, Algérie.

Abstract :
In this work, a robot A is moving in a dynamically changing space W, with imprevisible static obstacles and moving objects. We propose for that and in order to control the robot's movement, a reflecting loop based on an analysis of the robot's space. This technique leads to a better estimation of the collision risk and offers a way to effectively avoid obstacles.

1. Introduction :

In the global reaction approach, the robot's movement is directly induced from its local perception. It is continously changing by the evolution of the local plane. A heuristic analysis is implemented in the decision on to react to the robot's environment. A sensor information is included in this analysis.
In this approach, the robot is represented as a point in a set of spaces and it is considered as a moving particle under a field of artificial forces induced from the obstacles. These forces act as a repulsive potential when they originate from the obstacles and as an attractive potential when they are from the target to reach. The resulting force field is defined as the sum of all potentials.

$$\forall q \in C_f,\ U(q) = U_{att}(q) + U_{rep}(q) \qquad (1)$$

Therfore the force exerced upon the robot is :

$$\forall q \in C_f,\ F(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \qquad (2)$$

The movement is then iteratively computed from this force.
The attraction potential : The commonly used function is of the parabolic type :

$$U_{att}(q) = \tfrac{1}{2}\xi \,||\, q - q_{fin} \,||^2 \quad (\xi > 0) \qquad (3)$$

The derived force is then :

$$F_{att} = -\xi\,(q - q_{fin}) \qquad (4)$$

The repulsive potential : This potental is derived from the idea of considering a barrier around the obstacles. The robot motion is then not affected beyond a certain distance.
These constraints are in respect of function in the type of :

$$U_{rep} = \tfrac{1}{2}\xi\,(1/r(q) - 1/r_0) \quad \text{si } p(q) \le r_0$$
$$\text{ou}$$
$$= 0 \quad \text{si } p(q) = r_0 \qquad (5)$$

$p(q)$ is the distance between the obstacle and the space set and r_0 the effect distance of the obstacle.

2. Robot and environment settings :

A mobile robot A i modelised by a rectangle in the W plane. A coordinate system R_A is tied to it.
A reflex action consists of generating a motion in the settings space C_f, fig. 1.

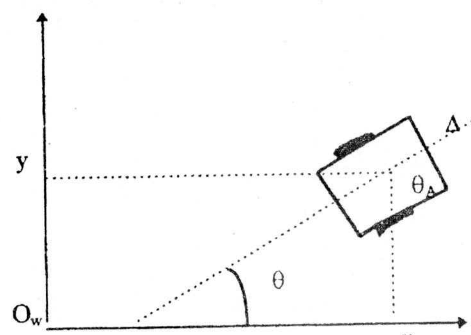


Fig. 1. Robot and environment settings

The setting vector X in the system space is :

$X = [x, y, \theta, q_d, q_g]$, where q_d is the articulation variable of the right wheel and q_g that of the left wheel.
We suppose the propulsion wheels suffer no friction and the relative speed betwen two points wheel-ground is zero.
The mobile robot equations are :

$$x\,\text{Cos}\theta + y\,\text{Sin}\theta - r/2\,(q_d - q_g) = 0$$
$$x\,\text{Sin}\theta - y\,\text{Cos}\theta = 0$$
$$\theta - r/2R\,(q_d - q_g) = 0 \qquad (6)$$

r is the wheel radius and R is the distance between the wheel and point M.
When the robot moves, O_A describes a curve γ tangent to the mobile axis. Every obstacle be it static or mobile B_i in W occupies a space noted $B_i(q)$.
The reflex action in the control of the robot is based on the response to an extrenal excitation which is

considered as an intrusion of an obstacle in the local space A. The motion in unknown environment needs a perception system to overcome the event of imprevisible obstacle and its presence.

The detection region Z_d is then $Z_d = E_d \cup E_d$. Where E_d is the right half space, E_g is the left half space, C_{v1} and C_{v2} are the vision fields on E_g and C_{v1} and C_{v2} are the vision fields on E_d.

Ultrsonic transducers are used to give the distance information needed in a fast and simple processing. Three faces of the robot are taken into account ; the left, right and front faces. 4 transducers are needed for each half space and therefore a total of 8 to cover the detection region Z_d. Every transducer is associated with a distance variable d_{ij} between the j^{th} transducer and the first accountered obstacle. At every sample time we have a vector data caracterising the environent as $\{d_{11},d_{12},d_{21},d_{22},d_{31},d_{32},d_{41},d_{42}\}$ derived from the transducers.

3. Estimating the transducers data :

Let D_i be the excursion universe of the variable d_{ij} in the i field. The number of lexical variables belonging to D_i determine the total number of settings which is given $N=[\text{card}(D_i)]^2$ (N=16 in this example). We can then define the N rules to offer the settings of the obstacles in the vision area.We have chosen belonging functions of the type , fig. 2 :

$L(d,d0,d1)$, $\Lambda(d,d0,d1,d2)$, $\Lambda(d,d1,d2,1)$, $\Gamma(d,d2,1)$

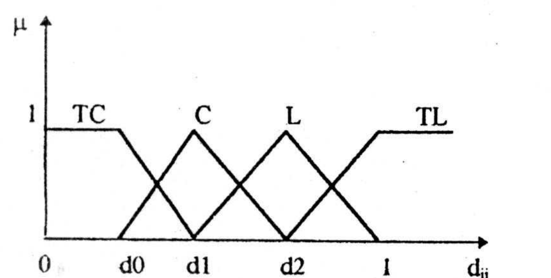


fig.2.partitioning the universe distance

The lexical variables asociated with this repartition so defined, are as follows : (TC : Too Short, C : Short, L : Long, TL : Too Long). These caracterise the distance d_{ij} between the transducer and the first echo.

The TC variable form defines a security region around the robot. When d_{ij} becomes less than d0, it is considered as TC. So we get a fuzzy variable dfined as :

$\mu_d = (\mu_{TC}(d),0,0,0) \Leftrightarrow \mu_d = (1,0,0,0)$

This enable us to control the action to perform. The construction of the fuzzy under set TC in [0,d0] defines a secure region around the robot.

The more the robot speed increases and the more the vision area should large and vice versa. In order to do this, we have to modify in line the variable d_{ij} for every transducer with respect to speed. The belonging functions are affected by a coefficient σ_i given by :

$\sigma_i = d_{min} + k_i V_A$

For every vision field C_{vi}, we have a k_i parameter. The functions are then :

$\mu_{TC}(d) = L(d,d0,\sigma_i,d1.\sigma_i)$
$\mu_C(d) = \Lambda(d,d0,\sigma_i,d2.\sigma_i)$
$\mu_L(d) = \Lambda(d,d1.\sigma_i,d2.\sigma_i,\sigma_i)$
$\mu_{TL}(d) = \Gamma(d,d2.\sigma_i,\sigma_i)$

In this process, we can vary the secure distance δ_{di} as :

$$\delta_{di} = d0\,(d_{min} + k_i V_A) \qquad (7)$$

δ_{di} varies in the same way as the system inertia giving us a mean to anticipate the motion to avoid obstacles for relatively large speeds.

4. Constraint index control :

For each obstacle is asociated a constraint index. This allows us to measure the risk of collision for the robot and gives us a qualitative space and cinematic constraint for a setting A(q). The rule basis for an index controller results in a ualitative information on the posibilities of the robot motion in A.

The rule is expressed as :

if <distance obtained from transducer 1> is A_l **and**
if <distance obtained from transducer 2> is A_m
then <constraint index> is B_{lm}. We note this rule, R_{lm}

with A_l and $A_m \in D$ and B_{lm} a lexical variable caracterising the constraint index to R_{lm} ($l,m \in [1,4]$).

Every index I_{lm} is the consequent in the rule R_{lm} . This gives the following rule table :

constraint index		d_{i2}			
		TC	C	L	TL
d_{i1}	TC	I_{11}	I_{12}	I_{13}	I_{14}
	C	I_{21}	I_{22}	I_{23}	I_{24}
	L	I_{31}	I_{32}	I_{33}	I_{34}
	TL	I_{41}	I_{42}	I_{43}	I_{44}

fig.3 constraint index rule table

This constraint indices universe is composed of fuzzy singletons whose variables are { 0,1,2,3}. A constrained setting is given an index I of 0 whereas a free setting is given a value 3. I_i is given by :

$$I_i = \frac{\sum_l \sum_m \alpha_{lm} I_{lm}}{\sum_l \sum_m \alpha_{lm}} \qquad (8)$$

with $\alpha_{lm} = \mu(d_{il}).\ \mu(d_{im})$

the fuzzy universe of the variable I_{dg} (constraint index for the right space E_d and left space E_d) is identical to the variable I_i. That is a set of four fuzzy singletons {0,1,2,3}. The rules taken into account are :

R_{lm} : **if** $I_{1,4}$ is A_l and $I_{2,3}$ is A_m **then** $I_{d,g}$ is C_{lm} . This gives us the following table :

I_{dg}		$I_{2,3}$			
		F	M	G	TG
$I_{1,4}$	F	0	0	0	0
	M	0	1	1	2
	G	1	1	3	3
	TG	1	2	3	3

fig.4. Constraint index rules table in E_g and E_d

5. Orientation controller :

The orientation control module is composed of a fuzzy controller whose inputs are the constraint indices I_g and I_d , fig. 6 . This unit gives as a function of the constraint ratio between the left and right spaces, the direction which minimizes the large constraint. For example if I_d is greater than I_g then the E_d is less constraining and therefore the orientation is that of the right space. The universe support of the variable $I_{g,d}$ is [0,3]. We take the triangular form for the belonging functions and everywhere equal to the input variable of the index controller. The orientation parameter is noted α_E in the robot coordinate system R_A. This variable set is a set of seven singletons whose supports are the points (-π/4,-π/6, -π/2, 0,π/2, π/4, π/6). The relative direction takes the real values in the interval [-π/4,π/4]. A choice on the seven singletons gives a sharper control in the orientation.

α_E		I_d			
		0	1	2	3
I_g	0	0	π/12	π/6	π/4
	1	π/12	0	π/2	π/6
	2	-π/6	-π/2	0	π/12
	3	-π/4	-π/6	-π/2	0

fig.5. Orientation rules table

5. Linear speed controller :

The necessary space for the robot to manoeuver becomes more important when its speed is higher. The nominal speed of the robot A is V_{nom} and the instantaneous speed can be given by $V_A = \lambda_E \cdot V_{nom}$. λ_E is a weighing coefficient and it is a function of the local perturbation. It is qualitatively defined by the doublet (I_d, I_g) with $\lambda_E \in [0,1]$. $\lambda_E = 0$ corresponds to a very onstrained free space (I_d =0, I_g=0) and leads to stop the robot. $\lambda_E = 1$ corresponds to a free space (I_d =3, I_g=3) and therefore to maximal speed.

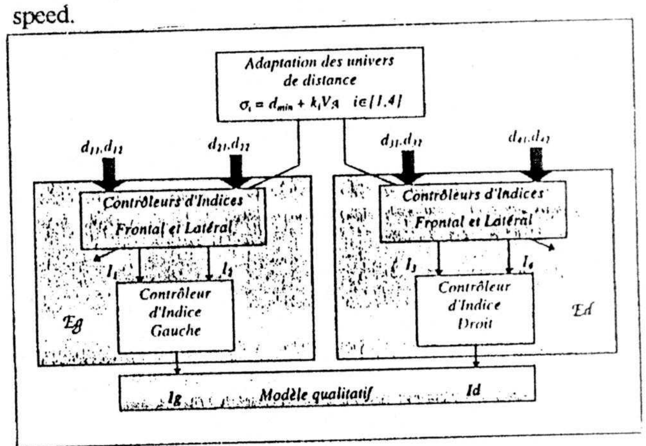


Fig.6. Constraint indices (I_g, I_d) computing in the half rightand left spaces.

6. Simulation results :

We give simulation examples of the evolution of the robot in a fixed environment with different initial settings. The environment area is 15mx15m. There is no target for the robot to each. Its only task is to avoid collision in its work space.

The belonging functions of the universes D_i, i∈[1,2,3,4], are :

$\mu_{TC}(d) = L(d, 0.1\sigma_i, 0.4\sigma_i)$

$\mu_C\ (d) = \Lambda(d, 0.1\sigma_i, 0.4\sigma_i, 0.7\sigma_i)$

$\mu_L\ (d) = \Lambda(d, 0.4\sigma_i, 0.7\sigma_i, \sigma_i)$

$\mu_{TL}\ (d) = \Gamma(d, 0.7\sigma_i, \sigma_i)$

with $\sigma_i = d_{min} + k_i\ \lambda_E.V_{nom}$. We have chosen $\sigma_1=\sigma_4=1.0 + 2.0\lambda_E.V_{nom}$ and $\sigma_2=\sigma_3=1.0 + 2.0\ \lambda_E.V_{nom}$

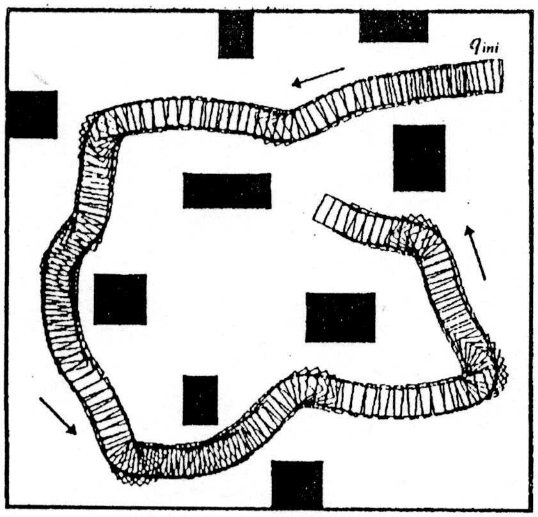
We found in the four ewamples, fig. 7, that there was no collision which proves the efficiency of the propose approach. However there was no global planning for the robot , the local perception was limited and the way taken by the robot was no otpimum. Morover in the fourth example the robot hit a dead end ; this setting corresponds to a symmetric qualitative model.

7. Références :

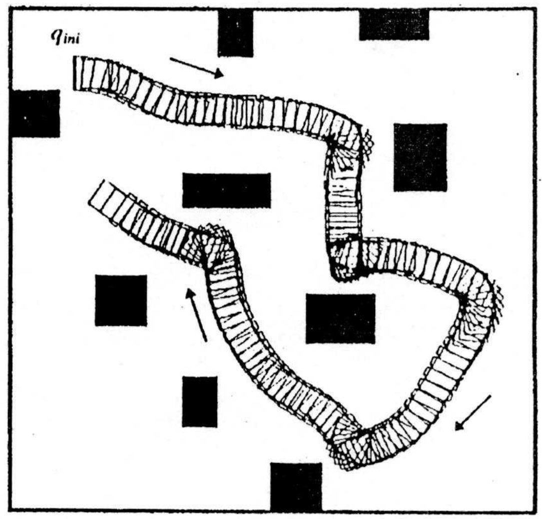
[1] R.C.Arking , Motor scheme based navigation for a mobile robot :An approach to programming by behavior. Int. Conf. On robotics and automation pp264.

[2] O.Kathib Real time obstacle avoidance for manipulators and mobile robots. The Int.Journal of Robotics Research Vol.5 n°1 pp90-98, 1986.

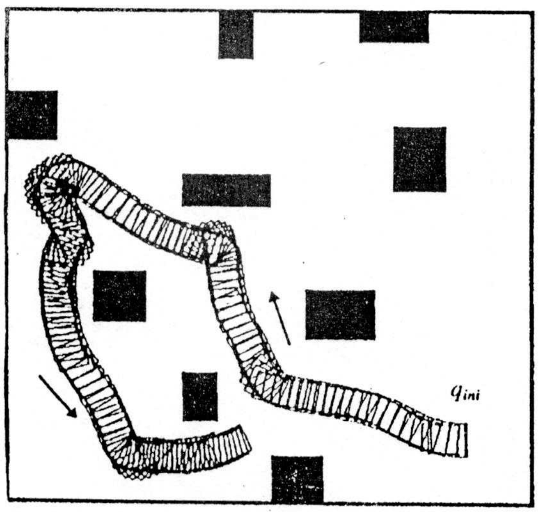
[3] S. Cameron Obstacle avoidance and path planning. Industrial Robot Vol. 12 n°5 pp 9-14, 1994.



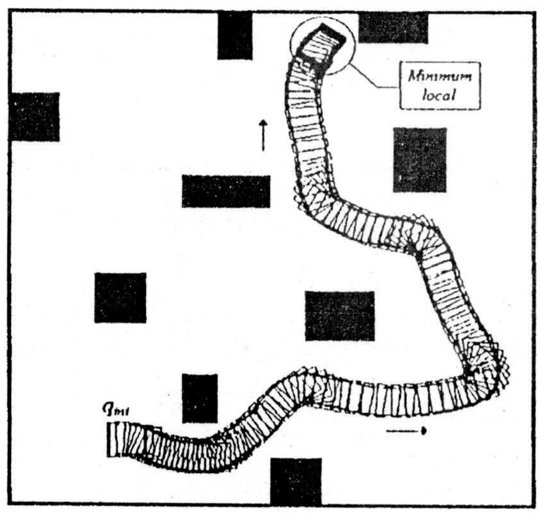
(a) : q_{ini} : $x=14$, $y=13$, $\theta=\pi$.



(b) : q_{ini} : $x=2$, $y=13$, $\theta=0$.



(c) : q_{ini} : $x=13$, $y=2$, $\theta=\pi$.



(d) : q_{ini} : $x=3$, $y=2$, $\theta=0$.

fig.7. Robot motion examples in a constraint enviroment with no target.