Temporal Neural Network applied in Reactive Navigation of Mobile Robot

L. Mesbahi [1] F. Hendel [2] N.E Berrached[2] A. Benyettou[1]
[1] Laboratoire SIMPA, Dépt. Computer science, USTO. Oran, Algérie
[2] Laboratoire LARESI, Dépt. Electronique, U.S.T.Oran, Algérie
Mesbahi_99@yahoo.com fat_hendel@msn.com neberrached@netscape.net aek.benyettou@email.com

***Abstract* -We present in this article, the realisation of reactive navigation module based on neural networks of temporal radial basis functions « TRBF », while using an orthogonal least square algorithm « OLS ». Applied to a type structured like interior of building, the robot must assure his task of navigation mildly all while avoiding obstacles without wandering, with the possibility to take in account the taken decisions in his past lasting trajectory.**
***Keywords* -Reactive navigation, TRBF, OLS, Mobile Robot.**

I. INTRODUCTION

The navigation is considered like a basis function of the complete system of mobile robot while basing on information on the nature of the environment.
The general idea consists in associating an elementary displacement of the robot to information of situation. This information is the same type that we use by in recognition of the environment. It is a vector of inputs about distances robot-measures counter environment among the 1024 possible [1].
For this input, the network should associate an elementary displacement of type: « Advance of one step, to turn on right or left in one step » [2].
For the recognition of situations the robot must have a panoramic vision on 360 degrees, independent of its movement [3].
Here in opposite, the sense of displacement of the mobile seems indispensable. In addition, it is necessary to recall that neuronal module to achieve is foreseen to function in link with the system of perception and the system of auctioneers commands, while following the following decision chain (to see Fig.1) [1] :

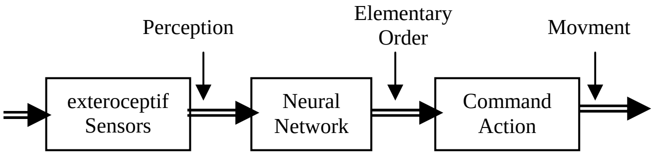

Fig.1 Module of navigation

We are not evidently here in case of the generation and the follow-up of an optimal trajectory; we ask nevertheless for the chain of navigation to be globally surest possible [4]. We have opts for a neuronal module based on temporal radial basis functions using Orthogonal Least Square contexts, because we think that it adjusts good with this type of problem (overlapping or oscillations owing a confused situation), in this case the taken decision insertion in the past becomes important and useful. So that we can integrate the temporal notion, we needed to play on the optimisation of all parameters, in object:

- To have an integral solution in a reliable hardware carried easily and fast.
- To use of less expensive sensors.
- Hardiness towards noises and the unforeseen shortcomings.

After this introduction we pass to section 2 to define the application about the navigation of mobile robot. In the section 3 we show how using a temporal Radial Basis Function in chain decision, based on the OLS algorithm (Orthogonal Least Square). Results and simulation are presented in the section 4. Finally we conclude by commenting the application while proposing some perspectives.

II. APPLICATION

A. Problem definition

We consider eleven elementary situations that a robot can frequently meet inside a building: passage, impasse, corner, piece, wall, left angle, input, right angle, crossing, T-crossing and output (see Fig.2)

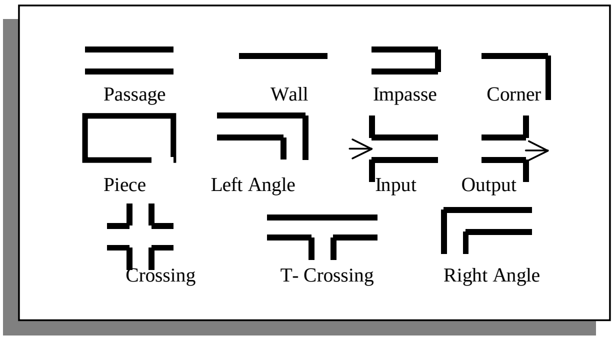

Fig.2 Elementary situations

In this environment the mobile robot must assure its stain of navigation mildly all while choosing an optimal course, for this effect we introduced the probabilistic notion in the action with hold in amount of the temporal aspect.
We must recall here that the sense of displacement of the robot has an influence on the creation of the training basis and in this goal we must consider the half-plan like source of information for actions of the robot (to turn on the right or on the left or Advancing).

B. Preparation of the Learning basis

For many situations of environment, the mobile is placed in uncertain way in Np different positions, with an uncertain initial orientation. Then R rotations of a step θ given are done there, creating R_i « examples » for every Npi position. We have thus: ($\sum_{i=1}$ Np_i* (R_i +1) examples with i : 1.. s ; s represents the number of chosen situations for the navigation. For every example, the vector of information containing N distances is recorded and a decision of elementary order of movement is chosen (see Fig.3, it shows the taken decisions according to the main direction of the axis of the robot in an environment of type passage).

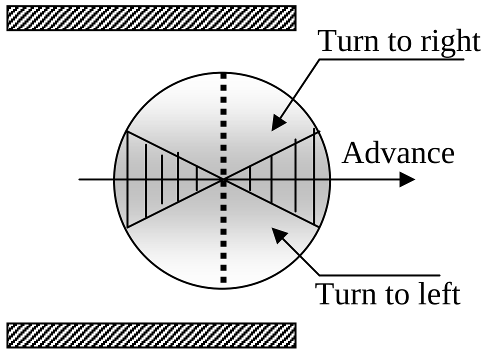

Fig.3. Decisions of displacement according to robot orientation in a passage.

At the time of the creation of the basis, we chose decisions in order to direct the mobile toward a trajectory situated toward the middle of the environment (for reason of displacement security).

The following Fig. 4 shows for a particular environment, that orders have been worked out of the training basis. We distributed this environment in 4 zones of Z1 to Z4.

- If the robot is in a zone « Z1 », from it position and it initial direction one will create R « examples » by P rotation of 10°. For every case, the chosen order will be compliant, according to the direction of the main axis, to the Fig. 3.

- Suppose now that the robot is in a zone « Z2 », with an initial orientation in direction of the left wall. We create as much then of « examples » by rotations of 10° toward the right that it is necessary so that the axis of the robot rejoins the axis of the passage. For every case an order « Turn To Right » will be associated. If the initial orientation moves away the mobile of the wall, we will associate an order « Advance » and none example won't be creates.

By duality, the same procedure has been used for the zone « Z3 ». If the robot is in the zone « Z4 », one will make it turn until to be in a compatible direction with the trajectory indicated on the face.

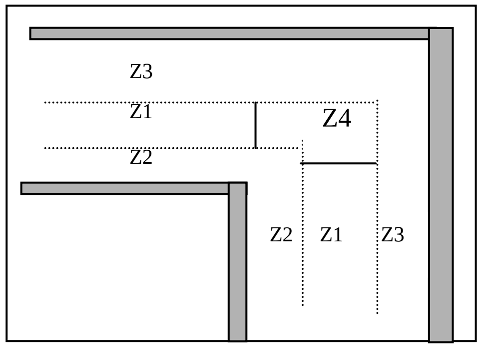

Fig.4. Decisions of displacement in function of the position of the robot in an angle.

Therefore we get an organized basis of a stationary number of examples ; every example understands 9 measures on the half-plan before (180°), knowing that we used situations (passage, crossing, T-crossing , wall, impasse , corner, angle, door and piece).

C. Advantages of temporal approach

This approach permits us to associate to resources our network a certain degree of confidence under a certain probabilistic angle, while basing on neural approximation to take the optimal decision, in order to avoid to knock itself to the wall or to ride.
Another problem concerning the oscillation of the robot especially in the impasse, in this case one of solutions is to introduce to the vector of sequence measures of examples holds in stationary time delay, as we choose a number of units known in the training for all actions to undertake.

III. TEMPORAL RBF APPROACH

The classic RBF can be formed to accomplish tasks of the recognition of shapes with no linear and complex shape; they are limited to treat some static models, rather than to treat shapes that are in temporal nature [5].
The Temporal RBF, as ATDNN, LSTM..., is proposed to defeat this limitation. Networks with this capacity can play an important role in the domain of applications that has properties varying like temporal signals and the dynamic shapes. As to take part of advantages of the classic RBFS in the approximation and the recognition, the objective come closer toward a behaviour wanted by a collection of functions, named kernels [6]. A kernel is characterized by a centre and a receptor field r, these kernels can be chosen by k-means or the vector quantification.
In general functions of temporal discrimination of the K class, is written under the following shape [5]:

$$y_k(t_n) = \sum_{j=1}^{m1} w_{jk}\varphi\left(\sum_{l=0}^{p} w_j(l)x(t_n - l) + b_j\right) + b_0 \qquad (1)$$

where $x(tn)=[x(t_n), x(t_n\text{-}1),... x(t_n\text{-}p)]^t$; b_j : the bias; p: is the memory order and m1 is the dimension of the hidden layer.

In the following part we describe the architecture of the network and the training algorithm.

A. Network Architecture

We propose the following architecture see Fig.5 and Fig.6:

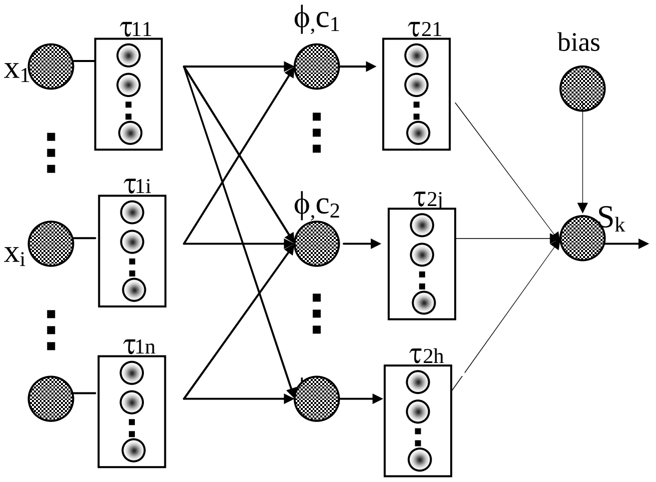

Fig.5 Representation of the TRBF network

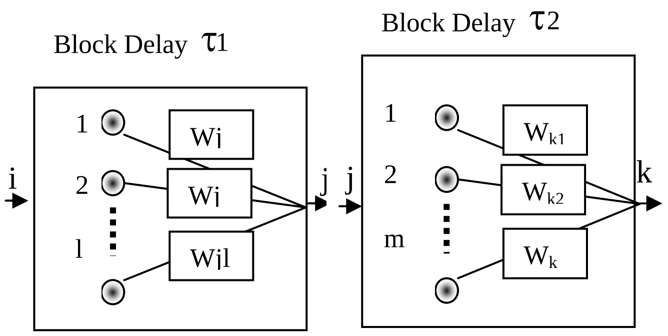

Fig.6 Representation of the delay block of (τ1,τ2)

Maintenant nous définissons:

$$S[k]=\varphi[W_{0k}+\sum_{j=1}^{m}\sum_{i=1}^{h}Yh[i][j]W_{kij}] \qquad (2)$$

Where φ: is a function of activation (Linear or Sigmoid etc) and W_{0k} are the slant. The kernel function is:

$$Yh[i][j]=\phi_{j,\sigma i}(\|C_i-X\|) \qquad (3)$$

Where i=1.. h, h: is the number of centres.
j=1.. m, m: is the dimension of the hidden time delay (τ2).
Dimension of Here = dimension of X = n x l.
n = number of feature of the input vector.
l = is the dimension of time delay of input (τ1).
$\phi_{j,\sigma i}$ = the function core characterized by the time delay of j delay with a receiving field σi (see Fig.7.)

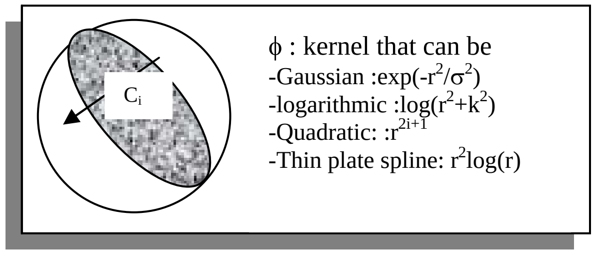

Fig.7 Kernel example

C. Training Algorithm

For this algorithm OLS « Orthogonal Least Square », we supposes that the kernel function (is fixed and that is the even for every hidden cell, the initial whole of centres must be fixed also. Therefore this algorithm permits to make an incremental training [6], [7], [8]. The OLS algorithm, conceived to the origin for the identification of no linear systems, can apply to the RBF network that can be considered like a particular case of the regression model linear definite by:

$$d = (P\ \theta + E) \qquad (4)$$

W is orthogonal image of *P:*

$$d= (W A\ \theta+E) \qquad (5)$$

This equation is used for iterative construction of the RBF network as criteria of selection. Hence an initial whole of M centers, the network is constructed to every iteration, by the addition of the center that possesses the value $[\text{maximal err}]_i$, and we takes the corresponding G_i.

$$G=A\ \theta \qquad (6)$$

Iteratively, we calculate elements of α and W by:

$$\alpha_{jk}^{\ i}=W^t_j*p_i/ W^t_j*W_j \qquad (7)$$

$$W^i_k=p_i-\sum_{j=1}^{k-1}\alpha_{jk}^{\ i}*W_j \qquad (8)$$

The criteria of iteration stop known of Akaike:

$$1-\sum^{m}_{i=1}err_i>\varepsilon \qquad (9)$$

In end of iterations, we calculate weights θ_i according to the system, A contains α values:

$$G=A*\theta \qquad (10)$$

IV. RESULTS AND SIMULATION

A. Parameters of training

We used the method of function networks to combined temporal radial basis with the OLS, applied at every creation of a corresponding network to such action.
We use data normalized in a vector of entrance to 9 measurements with time delay of 2 unit delay to the entrance with hidden layer that calculates its number of hidden neurons following an incremental approach, while following the criteria of Akaike (1-sum_erreurs > threshold) for the corresponding stop.
However we have limited the training by the choice of only one gaussian kernel instead of a mixture of kernels (risk of a big complexity).
We played on the value of the receiving field of the core that is worth between 1 and the spread of the training basis. In order to accelerate the process of count us made some up to

date stakes on the basis of training, either by the method to either center-reduce it by the shift of data (x'=2*x-1) [1,8].

B. Choices of kernels

Certain authors proposing to choose a variety of kernels in the training like thin: plate spline, the gaussian and multiquadratic kernels, but it asks for an enormous count time to choose the best center with the best kernel, in our survey, we fixed the kernel.

C. Training and Test rates

Table 1 Training Rates

Action	Turn on left	Advance	Turn on right	Global rate
Rate of training	95.88%	96.46%	97.9%	96.74%
Rate of recognition	94.23%	95.88%	97.53%	95.88%

Looking at the above table, we notice after these results that the navigation reacts well with the TRBF training. It comes back to the reduced class number that enters in conflict (3 classes) on the first hand, and to the fact that examples of training base has been chosen minutely.

D. Tolerance to noise

The Table.2 presents the gotten results while adding to data of the validation basis a gaussian noise of spread (variable between 0.01 and 0.1. One notes that until β=0.05 there nearly is not any reduction of performances. It assures practically that we could replace the telemeter laser by another sensor. When choosing a gaussian noise we got these results on Table. 2:

Table.2 Comparison between of noised data with a gaussian noise, while playing on the factor of spread β .

Action	Turn on left	Advance	Turn on right	Global rate
β=0.01	94.23%	95.88%	97.53%	96.74%
β=0.03	95.06%	93.41%	96.7%	95.05%
β=0.05	93.8%	92.5%	95.47%	93.92%
β=0.08	90.9%	89.3%	93.41%	91.20%
β=0.1	88.88%	87.65%	92.18%	89.57%

After this results, we notice that gaussian noise, in spite of increase of the spread type, didn't drag a total deterioration on the global rate until a spread β =0.1. We conclude that the margin of the spread type that keeps the best performances belongs to the interval [0.01, 0.05].

E. Simulation in unknown environment

The good results in environments of training incited us to do other tests to validate the capacity of the network to make sail correctly the robot in very different situations

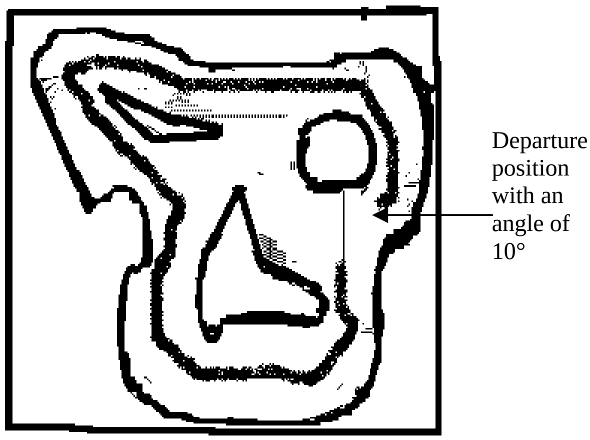

Fig.8. Environment test

The Fig. 8 shows that results gotten in a course through passage of shapes bent and of variable widths obliging the trajectory to have some various curvature radiuses. We note a very good navigation practically with a regular trajectory without to-stroke. All small obstacle placed close to the centre of the scene is perfectly avoided.

V. CONCLUSION

We presented in this article the realization of a reactive navigation module based on the use of a network like TRBF while introducing a stationary time delay to input vector. A training basis and a validation basis have been worked out from elementary environments. In every case, a decision of displacement is chosen among three possibilities: to Advance, to turn on the right, to turn on the left. The done choice generally aims to bring closer the robot of a median trajectory in the environment to cross. We tested various disruption influences then. While adding to measures of distance with a gaussian noise of spread variable, we showed the hardiness of network screw to screw of a measure noise can go until 5%. It guarantees that our network will be able to function correctly with sensors of modest performance.

Finally, the whole of simulations has been done while letting the robot sail in buckle closed under the conduct of the network. Beginning in the simple labyrinths, organized of various assemblies of situations learned, these simulations continued in cases, more and more distant of those of the training basis.
In general, the navigation took place with success. The described trajectory is always very soft; avoid all met obstacles, permits robot turns there in impasse in general way, resemble to what would make an animal or a man in the same situation. In spite of its extreme simplicity, the solution developed for the reactive navigation of our method seems robust, comfortably efficient, and transposable toward other sensors.

REFERENCE

[1] S. A. Alan. Recognition of environment and reactive navigation of an autonomous mobile robot by network of neurons, *Academic doctorate thesis*, Evry Val d'Essonne, 1996.

[2] C. Moez. Scheduling of movements for an autonomous mobile robot all ground: approach by using physic models, *Thesis of Doctorate*, Polytechnic School of Grenoble, 1995.

[3] C. Laugier, T. F. Philip Garnier, I. Parantchic and A. Scheuer. Sensor based control architecture for a car like vehicle, *RR 3552*, INRIA,1998.

[4] E. Soruchiari. Self organizing neural network for trajectory control and task coordination of a mobile robot, *Connection Science*, vol. 2, n°3, 1990.

[5] D.T. Lin. The adaptive time delay neural network characterization and application to pattern recognition, prediction and signal processing, *Thesis report PHD*, University of Maryland, 1994.

[6] V. Demian. Conception and parallel algorithm analysis for networks neuronauxes of KOHONEN and function to radial basis (RBF) , *Thesis Doctorate*, University Claude Bernard Lyon, 1995.

[7] L. Mesbahi and A. Benyettou. A new contribution towards a temporal radial basis function applied to Mackey-Glass time series, *ICNNAI'03*, Minsk, Byelorussia, 12, 13, 14 Nov 2003.

[8] L. Mesbahi and A. Benyettou. Application of radial basis functions to recognition of environment for mobile robot, *RFIA' 04, Toulouse, France, 2004.*