

# Application des fonctions à base radiale pour la reconnaissance d'environnement d'un robot mobile

## Application of Radial Basis Function in Environments Recognition for Mobile Robot

L. Mesbahi

A. Benyettou

Laboratoire Signal-Image-Parole(SIMPA), Dépt. Informatique

Faculté des Sciences, Université des Sciences et de la Technologie d'Oran –USTO, Algérie.

BP. 1505, Oran El-Mnaouer 31000,

Tel : (213)41420608 – Fax (213)41420680

Mél : [mesbahi\\_99@yahoo.com](mailto:mesbahi_99@yahoo.com), [ack.benyettou@email.com](mailto:ack.benyettou@email.com)

### Résumé

*Dans cet article, nous avons réalisé un classifieur neuronal bayésien à fonctions de base radiale, basé sur une version améliorée de l'algorithme OLS (apprentissage rapide et incrémentale – création automatique des neurones cachés, etc.). Appliqué à un type structuré d'environnement comme à l'intérieur de bâtiment, ce classifieur doit assurer une localisation sémantique fondée sur une approche réaliste. La volonté a été de rendre l'approche de discrimination la plus générale possible, en utilisant une représentation générique et puissante des connaissances à base de probabilités conditionnelles, et en utilisant pour la manipulation de ces connaissances, des classifieurs Bayésiens (introduction des probabilités a priori – coûts d'erreurs – rejets, etc.), ces classifieurs peuvent être générés par le biais des réseaux de neurones (peu coûteux – faciles à intégrer – rapides, etc.). Donc au lieu d'avoir une décision binaire, il est plus informatif de prendre une décision de savoir qu'il y a 90% de chances que le robot mobile se trouve dans une situation impasse par exemple, plutôt que d'avoir comme seule indication «situation: impasse».*

### Mots Clef

Classifieur neuronal bayésien, fonctions à bases radiales, robot mobile, localisation sémantique.

### Abstract

In this article, we have used a Bayesian neural classifier, which is based on probabilistic approach with Radial Basis Functions "RBF" and based on an improved version of Orthogonal Least Square algorithm "OLS" (fast and incremental learning - automatic creation of hidden neurones, etc.). Applied to a structural environment like inside a building, this classifier must assure a semantic localization, established on a realistic approach. The will wish to have a discrimination approach in the most possible case, by using a generic and powerful representation of knowledge based on conditional probabilities, which have applied with Bayesian classifier (introduction of prior probability - error costs - case of throws, etc.). This classifiers have been generated by neural network. Therefore instead to have a binary decision such for

example, a hard decision of impasse, the mobile robot decides for 90% of impasse situation.

### Keywords

Bayesian neural classifier, Radial basis function, mobile robot, semantic localization.

### 1 Introduction

Dans un environnement quelconque, un robot mobile doit être capable de se localiser à tout moment si on le lui demande, en donnant alors sa position en coordonnées géométriques.

Dans notre étude, il s'agit plutôt d'une localisation sémantique d'un robot qui reconnaît une telle situation, par exemple (couloir - impasse – mur, etc.) et cela dans un environnement bien structuré et non connu de type immeuble.

Vu la tâche très difficile de ce robot, on a un grand intérêt pour les techniques neuronales :

- Tolérance au bruit, les données provenant des capteurs du robot qui sont généralement bruitées.
- Adaptation au changement temps réel.
- Capacité d'apprentissage et de généralisation.

Pour cela, on a opté pour la mise au point d'un classifieur bayésien qui, à partir des données des capteurs fournies par un télémètre laser, on trouve l'environnement simple du robot sous un angle probabiliste. Cela sera certainement indispensable pour créer un dialogue fiable entre la reconnaissance et la navigation [3]. Il est en effet beaucoup plus informatif de prendre une décision de savoir qu'il y a 90% de chance que le robot se trouve dans un couloir plutôt que d'avoir comme seule indication «situation: couloir». Ainsi la formulation bayésienne est plus adaptée à un tel problème par rapport à d'autres approches [11].

Une classification à base radiale, basée sur un apprentissage incrémentale à l'image du cerveau humain, permet de corriger quelques caractéristiques des réseaux multicouches qui marquent leur lenteur d'apprentissage et surtout le caractère global de leur apprentissage qui oblige généralement de repartir à zéro si nous modifions la base

de données. La performance des réseaux RBF dépend du nombre et position des noyaux, leurs formes et la méthode utilisée pour l'apprentissage. Parmi les stratégies d'apprentissage utilisées, on peut citer [6]:

- Stratégie de sélection aléatoire des centres.
- Stratégie d'emploi des méthodes non supervisées pour le choix des centres.
- Stratégie d'emplois des méthodes supervisées.

En plus notre approche est basée sur l'algorithme d'OLS, qui permet l'introduction des probabilités a priori des situations, les coûts d'erreurs et la possibilité d'ajouter des situations sans refaire l'apprentissage de nouveau, tout cela pour un temps d'apprentissage relativement acceptable.

Ce papier est organisé comme suit : section 2 définit la formulation du problème, la section 3 présente la solution par les RBF avec leur apport, la section 4 concerne l'application avec un commentaire des résultats obtenus et nous concluons par des remarques dans la section 5.

## 2. Application

### 2.1 Définition du problème

Notre environnement concerne onze situations qu'un robot peut rencontrer à l'intérieur de bâtiment : couloir, impasse, coin, pièce, mur, angle-gauche, entrée, angle-droit, croisement, croisement en T et sortie (voir Fig.1).

On remarque que cette modélisation n'est pas figée, d'où on peut ajouter ou supprimer des situations [1][3].

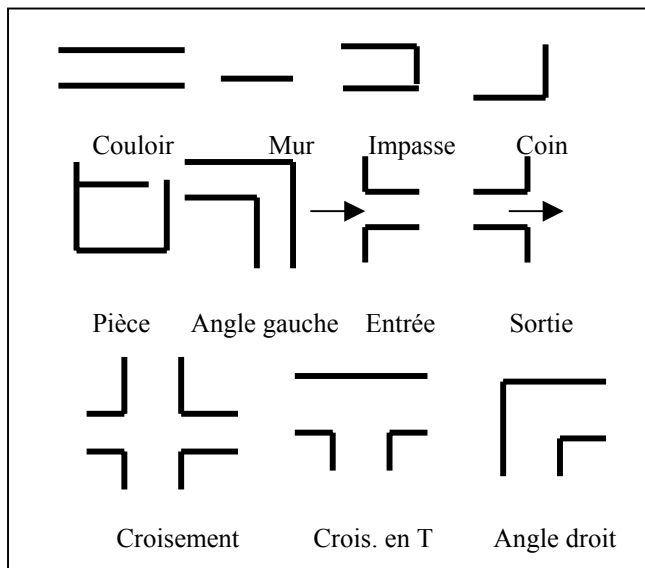


Fig.1. Environnement d'un immeuble à 11 situations en 2D.

Dans cet environnement, le robot doit répondre «je suis devant un mur», «je suis dans un couloir», etc. En plus il doit résoudre le problème de confusion et de chevauchement entre les différentes situations durant sa mission.

A cet effet, nous avons introduit le degré d'occurrence de la probabilité d'une telle situation avec le minimum d'erreur [16]. Cela nous emmène à utiliser la classification bayésienne [12].

### 2.2 Intérêt de l'approche bayésienne

Cette approche nous permet d'associer aux ressources du réseau un certain degré de confiance sous un angle probabiliste en se basant sur l'approximation d'une probabilité a posteriori pour prendre la décision optimale. Par exemple (voir Fig.2), nous remarquons que dans la zone d'étude il y a un risque de confusion entre (couloir - tourner à droite - croisement en T- croisement). Dans ce contexte, nous devons utiliser une technique optimale de pénalisation (symétrique ou diagonale) qui permet de renforcer la probabilité a posteriori en introduisant les coûts d'erreurs et les probabilités a priori tout en respectant le taux d'erreur optimal [12][17].

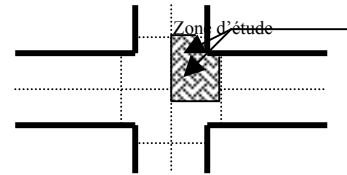


Fig.2. Zone d'étude pour un croisement

Les règles de décision bayésienne sont les suivantes :

- $P(\Omega_i)$ : probabilité d'occurrence de la classe  $\Omega_i$ .
- $p(x/\Omega_i)$ : fonction de densité de probabilité conditionnelle d'occurrence de  $x$  sachant que  $x \in \Omega_i$ .
- $p(\Omega_i/x)$ : fonction de densité de probabilité a posteriori d'occurrence de la classe  $\Omega_i$ , connaissant le vecteur  $x$ .
- $p(x)$ : fonction de densité de probabilité d'occurrence du vecteur  $x$ .

On décide que :

$$x \in \Omega_i \text{ si } p(\Omega_i/x) \geq p(\Omega_j/x), \quad \forall i \neq j \quad (1)$$

En appliquant la formule de Bayes :

$$p(\Omega_i/x) \cdot p(x) = p(x/\Omega_i) P(\Omega_i). \quad (2)$$

Il vient que :

$$p(x/\Omega_i) P(\Omega_i) \geq p(x/\Omega_j) P(\Omega_j), \quad \forall i \neq j \quad (3)$$

Dans le cas où les classes ne sont pas disjointement séparables [4], on remarque un chevauchement entre les classes, ce qui donne un coût d'erreur [8][9][15], de la forme :

$$R(\Omega_i/X) = \sum_{j=1}^{j=nbr\_situation} r(\Omega_i/\Omega_j) * p(\Omega_j/X) \quad (4)$$

dont  $r(\Omega_i/\Omega_j)$  est le risque de classer  $x$  dans  $\Omega_i$  sachant qu'il appartient à  $\Omega_j$ .

## 3. Approche des réseaux de fonctions à base Radiales

Les réseaux de fonctions à base radiale sont utilisés dans la classification, l'approximation et la reconnaissance de parole, leur but est d'approximer un comportement désiré par une collection de fonction, appelées noyaux, un noyau est caractérisé par un centre  $C_i$  et des champs récepteurs  $r$ . Les réseaux de fonctions à base radiale ont été étudiés et appliqués dans différents disciplines vue leurs caractéristiques d'approximation, d'interpolation,

l'introduction des densités de fonction et leur taux de convergence (travaux de Michelli 1988 ; Powell 1992 ) [8].

Il a été démontré que les RBF peuvent approximer n'importe quelle fonction continue à plusieurs variables dans un domaine compact, si un nombre suffisant de neurones est présenté [11].

### 3.1 Estimation probabiliste

Il y a plusieurs techniques d'estimation, parmi lesquelles on trouve le maximum d'entropie [15], la technique de la descente de gradient avec la quantification vectorielle [18], [2]. Notre estimation est une variante de l'estimation de Parzen et elle doit suivre les mêmes conditions de consistance. Le problème majeur est de choisir le noyau convenablement (centre et champ récepteur) (voir Fig.3).

Donc si le noyau est étroit, la densité estimée devient discontinue et marqué par des piques ; si le noyau est large, l'estimation devient avec une grande tolérance et ne montre pas assez de détails dans la densité [14]. Notre choix donc est tombé sur le noyau gaussien (voir Fig.4) vue ses caractéristiques asymptotiques. En général le meilleur noyau qui permet d'estimer une fonction de densité dépend des paramètres de cette dernière et la taille d'échantillons disponible. Notre jugement est laissé lors de l'application de ces estimateurs.

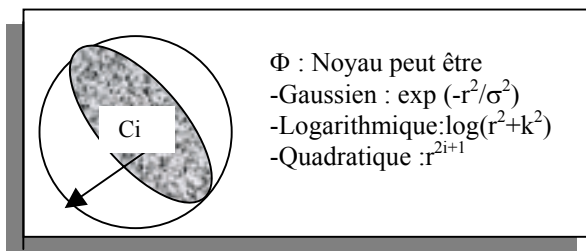


Fig.3. Exemple d'un noyau

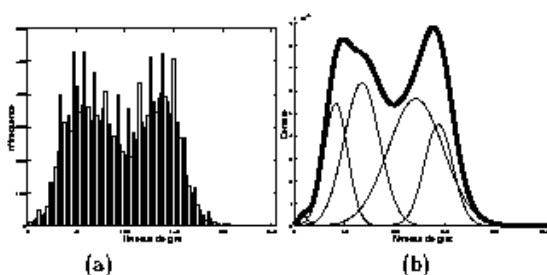


Fig.4. Approximation par le mélange gaussien

(a)-Histogramme de niveau de gris image  
 (b)-Approximation par un mélange de noyaux gaussien

Nous montrons maintenant le modèle du réseau RBF (voir Fig.5.) :

$$F_k(x) = w_0.k + \sum w_{k,i} \cdot \phi(|x - c_i|) \quad (5)$$

$F_k(x)$  est considérée comme un estimateur de probabilité conditionnelle  $p(x/\Omega_i)$  pour la décision Bayésienne.

$F_k(x)$  est un bon estimateur de  $p(x/\Omega_i)$ , parce que l'approche par noyau est une application des fenêtres de

Parzen, qui respectent bien les conditions d'un bon estimateur universel [10], [13].

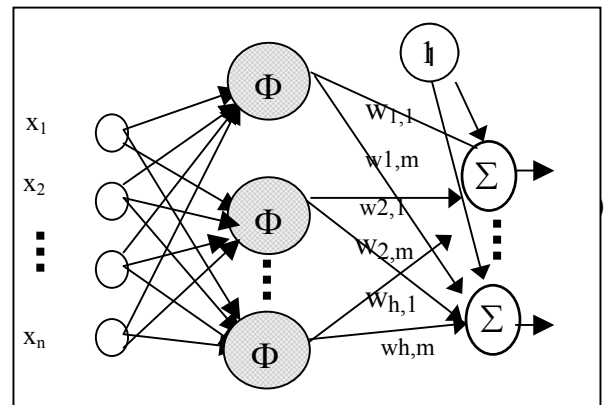


Fig.5. Architecture d'un réseau RBF

### 3.2 Algorithme d'apprentissage OLS

Pour cet algorithme OLS « Orthogonal Least Square », on suppose que la fonction noyau  $\phi$  est fixée et qu'elle est la même pour chaque cellule cachée, l'ensemble initial des  $h$  centres doit être également fixé.

Donc cet algorithme permet de faire un apprentissage incrémentale [7] :

- premièrement il fait la séparation linéaire entre la couche d'entrée et la couche cachée, il crée les neurones cachés automatiquement en appliquant l'orthogonalisation de Gram-Schmidt, qui permet d'éliminer les redondances d'informations, d'autres méthodes consistent à utiliser les algorithmes génétiques pour minimiser le nombre de neurones cachés avec une bonne généralisation [9].
- deuxièmement faire de l'apprentissage entre la couche cachée et la couche de sortie, en calculant les poids synaptiques tout en se basant sur la méthode des moindres carrés.

L'algorithme OLS, conçue à l'origine pour l'identification des systèmes non linéaires, peut s'appliquer au réseau RBF qui peut être considéré comme un cas particulier du modèle de régression linéaire définie par :

$$d(t) = \sum P_i(t) \theta_i + \varepsilon(t) \quad (6)$$

Ou  $d(t)$  : la sortie désirée à l'instant  $t$

$\theta_i$  : sont les paramètres recherchés.

$\varepsilon(t)$  : l'erreur d'approximation de  $d(t)$ .

$P_i$  : Des fonctions fixées de  $x(t)$

(Regresseur)

$$P_i(t) = P_i(x(t)).$$

La fonction calculée par un réseau RBF est la même que la fonction de la formule (6) L'analogie est la suivante :

$d(t)$  : La sortie désirée du réseau pour le  $t^{\text{ième}}$  exemple.

$P_i(t)$  : La sortie de la  $i^{\text{ième}}$  cellule cachée pour le  $t^{\text{ième}}$  exemple.

La constante  $w_0$  (biais) peut être obtenue par la définition du  $P_i(t) = 1$  correspondant. Par la suite, nous allons considérer les notations suivantes :

$N$  : Le nombre d'exemples de la base d'apprentissage.

M : Le nombre initial des centres.

d : Le vecteur des sorties désirées :

$$d = [d(1) \dots d(N)]^T.$$

P : La matrice des sorties de la couche cachée :

$$P = [P_1 \dots P_M].$$

P<sub>i</sub> : Le vecteur de sortie de la i<sup>ème</sup> cellule cachée :

$$P_i = [P_i(1) \dots P_i(N)]^T.$$

θ : Le vecteur des poids de la couche de sortie :

$$\theta = [\theta_1 \dots \theta_M].$$

E : Le vecteur des erreurs entre les sorties calculées par le réseau et les sorties désirées :

$$E = [\varepsilon(1) \dots \varepsilon(N)]^T.$$

Avec les notations définies ci-dessus, l'équation (6) peut s'écrire sous la forme matricielle, de la manière suivante :

$$d = (P \theta + E) \quad (7)$$

La résolution du système d'équation (7) est un problème trivial. Le vecteur des solutions θ peut être défini par la méthode des moindres carrés.

L'idée originale de la méthode OLS réside dans la transformation de la matrice P en une matrice avec des colonnes orthogonales une à une. L'orthogonalisation des colonnes P<sub>i</sub> peut être obtenue par la décomposition de la matrice P en deux matrices W et A :

$$P = W A. \quad (8)$$

Où W : de taille N x M est l'image orthogonale de la matrice P.

A : de taille M x M est une matrice triangulaire supérieure qui contient les coefficients d'orthogonalisation.

La matrice A est définie comme suit :

$$A = \begin{bmatrix} 1 & \alpha_{1,2} & \dots & \alpha_{1M} \\ 0 & 1 & \dots & \alpha_{2,M} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \alpha_{M-2,M-1} & \alpha_{M-2,M} \\ 0 & 0 & \dots & 0 & 1 & \alpha_{M-1,M} \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

L'espace engendré par les vecteurs P<sub>i</sub> est le même espace engendré par les vecteurs W<sub>i</sub>, et le système d'équation (8) peut être réécrit comme suit :

$$d = W^*G + E \quad (9)$$

tel que G=A. θ est la solution recherchée.

$$\text{Notons : } H = W^T \cdot W E \quad (10)$$

puisque les colonnes de la matrice W sont orthogonales une à une, H est une matrice diagonale avec des éléments h<sub>i</sub> tel que :

$$h_i = w_i^T w_i = \sum_{t=1}^N w_i(t) \cdot w_i(t), \quad 1 \leq i \leq M \quad (11)$$

C'est cette propriété qui rend la méthode OLS intéressante, et ceci pour la raison suivante : la solution orthogonale G est calculée par :

$$G = H^{-1} \cdot W^T d \quad (12)$$

peut être écrite :

$$G_i = w_i^T d / (w_i^T \cdot w_i), \quad 1 \leq i \leq M \quad (13)$$

Ceci signifie que les éléments G<sub>i</sub> de la solution orthogonale G ne dépendent que de la colonne w<sub>i</sub> ou autrement dit, de l'image orthogonale de la sortie calculée par chaque centre. Cette partie définit le quotient de la réduction de l'erreur d'approximation introduite par chaque vecteur w<sub>i</sub> :

$$[err]_i = G_i^2 \cdot w_i^T \cdot w_i / (d^T \cdot d), \quad 1 \leq i \leq M \quad (14)$$

Cette équation est utilisée pour la construction itérative du réseau RBF comme critère de sélection. Partant d'un ensemble initial de M centres, le réseau est construit à chaque itération, par l'ajout du centre qui possède la valeur [err]<sub>i</sub> maximale, et on prend le G<sub>i</sub> correspondant.

A chaque itération on calcule les éléments de A et W par :

$$\alpha_{jk}^i = W_j^* p_i / W_j^* W_j \quad (15)$$

$$W_{j \neq k}^i = p_i - \sum_{k=1}^{k-1} \alpha_{jk}^i W_j \quad (16)$$

Le critère d'arrêt des itérations connu de Akaike :

$$1 - \sum_{i=1}^m [err]_i > \varepsilon \quad (17)$$

En fin d'itérations, on calcule les poids de pondération θ<sub>i</sub> selon le système :

$$G = A^* \theta \quad (18)$$

## 4. Résultats et simulation

### 4.1 Ajustement des Paramètres OLS

Nous avons pris les paramètres suivants :

1- Une base de 528 exemples contenant 48 exemples pour chaque situation (voir fig.1.), ou chaque exemple est un vecteur de 16 composantes caractérisant les distances autour du robot sur 360 degrés, simulant une vision panoramique du Robot.

1-Un noyau gaussien

2-Les données sont normalisées sur [0,1] et centrées-réduites par la moyenne et l'écart type de la base d'apprentissage.

3-Le champ récepteur est pris égale à 0.8.

4-le critère d'arrêt d'Akaike avec seuil=0.04 .

### 4.2 Choix des noyaux

Certain auteurs proposant de choisir une variété de noyaux dans l'apprentissage du thin plate spline, la gaussienne et multiquadratique, mais cela demande un temps de calcul énorme pour choisir le meilleur centre avec le meilleur noyau, dans notre étude, on a fixé le noyau pour tous l'apprentissage à titre d'exemple, et on a fait une comparaison entre le noyau de gauss et multiquadratique (r=3.5) [4], [5]. On a remarqué que le noyau de Gauss est le meilleur, vu ses caractéristiques asymptotiques et statistiques.

### 4.3 Fin des itérations

Afin de trouver le meilleur réseau, nous avons utilisé la méthode « spy », qui a été développée par Helene Paugam Moisy [4] et était utilisée avec succès dans différentes

applications neuronales. Cette méthode permet de converger vers la solution optimale en utilisant le critère d'arrêt des itérations en utilisant le seuil d'Akaike.

#### 4.4 Coûts de pénalisation

Dans ce cas on a choisi la stratégie de la pénalisation diagonale et sans rejet, qui consiste à affecter un coût d'erreur à la décision de la classe pour éviter une confusion flagrante dans la séparation entre les différentes situations.

On peut sanctionner plus fortement certaines décisions erronées que d'autres.

Ceci se formule comme suit :

$$R(\Omega_i/X) = \sum_{j=1}^{j=nbr\_situation} r(\Omega_i/\Omega_j) * p(\Omega_j/X) \quad (20)$$

$$r(\Omega_i/\Omega_j) = \begin{cases} -H_i & \text{si } i=j \\ 0 & \text{si } i \neq j \end{cases} \quad (21)$$

Si on applique cette règle à la fonction d'erreur de l'équation (20) on obtient :

$$R(\Omega_i/X) = -H_i * p(\Omega_i/X) \quad (22)$$

Cela peut s'expliquer par une récompense pour les bonnes décisions et il n'y a ni pénalisation ni récompense pour les mauvaises décisions.

Par critère de minimisation, on décide  $\Omega_i$  si :

$$H_i * p(\Omega_i/X) > H_j * p(\Omega_j/X), \forall j \neq i \quad (23)$$

En appliquant la formule de Bayes :

$$H_i * \frac{p(X/\Omega_i) * P(\Omega_i)}{p(X)} > H_j * \frac{p(X/\Omega_j) * P(\Omega_j)}{p(X)} \quad \forall j \neq i \quad (24)$$

D'une autre façon on définit la probabilité a posteriori [13], [16]:

$$p(\Omega_i/X) = \frac{p(X/\Omega_i) * P(\Omega_i)}{p(X)} \quad (25)$$

sachant que

$$p(X) = \sum_{j=1}^{j=nbr\_situation} p(X/\Omega_j) * P(\Omega_j) \quad (26)$$

Nous avons considéré que les situations sont équiprobables :

$$P(\Omega_i) = P(\Omega_j), \forall j \neq i \quad (27)$$

Donc l'équation (25) s'écrit :

$$p(\Omega_i/X) = \frac{p(X/\Omega_i)}{\sum_{j=1}^{j=nbr\_situation} p(X/\Omega_j)} \quad (28)$$

Et puisque le coût  $H_i$  égale à 1 alors remplaçons (28) et le coût dans (24) on aura (29) :

Pour choisir une situation  $i$  il faut que :

$$\frac{p(X/\Omega_i)}{\sum_{m=1}^{m=nbr\_situation} p(X/\Omega_m)} > \frac{p(X/\Omega_j)}{\sum_{m=1}^{m=nbr\_situation} p(X/\Omega_m)} \quad \forall j \neq i \quad (29)$$

#### 4.5 Résolution des indécisions

Que doit répondre le réseau lorsqu'il se trouve dans des situations d'indécision ? Une solution consiste à inclure

ces situations d'indécision dans l'une des classes entre lesquelles le réseau hésite, généralement l'indécision survient entre la classe "couloir" et une autre classe. Si on veut résoudre la plupart des conflits l'aspect couloir est le plus commun dans la majorité des situations. La classe "couloir" serait désignée en cas de conflit.

#### 4.6 Taux de reconnaissance

On a pris pour la base d'apprentissage 48 exemples pour chaque situation et pour la base de généralisation 12 exemples pour chaque situation.

Après application de notre algorithme nous avons obtenu une architecture neuronales (16 neurones d'entrée+49 neurones cachés+1 neurone de sortie) pour chaque situation, avec un pourcentage d'apprentissage de 100% et 92.42% en généralisation ( voir Table1).

|                | Couloir | Mur     | Impasse |
|----------------|---------|---------|---------|
| Apprentissage  | 48      | 48      | 48      |
| Généralisation | 12      | 12      | 12      |
| Coin           | Crois   | Crois_T | Pièce   |
| 48             | 48      | 48      | 48      |
| 12             | 12      | 12      | 12      |
| Angle_D        | Angle_G | Entrée  | Sortie  |
| 48             | 48      | 48      | 48      |
| 9              | 8       | 9       | 12      |

Table.1. Taux de reconnaissance des situations

#### 4.7 Tolérance aux bruits

Dans le but d'optimiser le coût de revient des robots mobiles, nous avons testé notre classifieur bayésien face à des données bruitées.

| Ecart type du Bruit | Couloir | Mur     | Impasse |
|---------------------|---------|---------|---------|
| 0.1                 | 12      | 12      | 12      |
| 0.5                 | 12      | 12      | 12      |
| 0.6                 | 10      | 12      | 12      |
| Coin                | Crois   | Crois_T | Pièce   |
| 12                  | 12      | 12      | 12      |
| 12                  | 8       | 12      | 12      |
| 12                  | 8       | 12      | 12      |
| Angle_D             | Angle_G | Entrée  | Sortie  |
| 9                   | 8       | 12      | 12      |
| 9                   | 8       | 12      | 10      |
| 9                   | 7       | 11      | 10      |

Table.2. Taux de reconnaissance des situations par addition d'un bruit Gaussien.

Les résultats que nous avons obtenu représentent des taux de reconnaissance aux environ des 92.4%, 90.15 et 87.12% respectivement au bruits dont l'écart type est 0.1, 0.5 et 0.6.

D'après les résultats donnés par ce tableau, on a remarqué que le réseau réagit bien lors de l'addition d'un bruit de

type Gaussien, qui représente des perturbations au niveau de la saisie des données par un capteur ultrason, ce dernier est moins coûteux qu'un télémètre laser. Donc il serait intéressant de doter notre robot mobile par des capacités de perception avec un coût faible, tout en gardant ses performances intactes.

#### 4.8 Adaptation au changement des tailles d'environnements

| Facteurs D'homothétie | Couloir | Mur     | Impasse |
|-----------------------|---------|---------|---------|
| 1                     | 12      | 12      | 12      |
| 1.1                   | 12      | 11      | 12      |
| 0.9                   | 8       | 12      | 12      |
| Coin                  | Crois   | Crois T | Pièce   |
| 12                    | 8       | 12      | 12      |
| 12                    | 8       | 12      | 12      |
| 12                    | 8       | 12      | 12      |
| Angle D               | Angle G | Entrée  | Sortie  |
| 9                     | 12      | 12      | 12      |
| 9                     | 8       | 9       | 10      |
| 9                     | 8       | 11      | 11      |

Table. 3. Taux de reconnaissance des situations par changement des tailles d'environnement

Les résultats représentent des taux de reconnaissance aux environ des 92.4%,88%et 88% respectivement au facteurs 1,1.1 et 0.9.D'après les résultats donnés par la table 3, on a remarqué que le réseau réagit bien lors de la multiplication par les facteurs d'homothétie qui simulent des changements des tailles des environnements pour le robot mobile. Ceci prouve le bon comportement des RBF face aux changements des tailles des situations visitées par le robot.

#### 4.9 Matrice de Confusion

Le but de la matrice de confusion est de voir et sentir le chevauchement et le conflit des situations. Dans cette matrice nous avons 9 situations à raison de 48 exemples pour chaque situation (voir table 4).

|         | couloir | Impasse | Angle  | Porte  | Crois  |
|---------|---------|---------|--------|--------|--------|
| couloir | 100%    | 0%      | 0%     | 0%     | 0%     |
| Impasse | 2.08%   | 97.91%  | 0%     | 0%     | 0%     |
| Angle   | 0%      | 0%      | 95.83% | 0%     | 0%     |
| porte   | 0%      | 0%      | 0%     | 81.25% | 14.58% |
| Crois   | 0%      | 0%      | 0%     | 0%     | 100%   |
| CroisT  | 0%      | 0%      | 4.16%  | 0%     | 6.25%  |
| Piece   | 0%      | 0%      | 0%     | 0%     | 0%     |
| mur     | 0%      | 0%      | 0%     | 0%     | 0%     |
| Coin    | 0%      | 0%      | 4.16%  | 0%     | 0%     |

|         | CroisT | Piece | mur    | Coin   |
|---------|--------|-------|--------|--------|
| couloir | 0%     | 0%    | 0%     | 0%     |
| Impasse | 0%     | 0%    | 0%     | 0%     |
| Angle   | 4.16%  | 0%    | 0%     | 0%     |
| Porte   | 0%     | 4.16% | 0%     | 0%     |
| Crois   | 0%     | 0%    | 0%     | 0%     |
| CroisT  | 87.5%  | 0%    | 2.08%  | 0%     |
| Piece   | 0%     | 100%  | 0%     | 0%     |
| mur     | 0%     | 4.16% | 95.83% | 0%     |
| Coin    | 0%     | 0%    | 0%     | 95.83% |

Table.4 . Représentation d'une matrice de confusion

D'après la table 4 le taux de confusion est de 5.1%, cela veut dire que le classifieur est satisfaisant, toutefois on peut dire que le robot a quelques difficultés dans le cas d'un passage par la porte ou devant un croisementT, par contre on remarque un bon comportement dans le couloir, croisement et pièce. Nous pouvons dire que pour avoir un bon classifieur, il faut que les éléments diagonaux de la matrice soient maximums.

#### 4.10 Effet de pénalisation

Nous appliquons la décision bayésienne au labyrinthe suivant (voir Fig.6), avec:

Supposition 1: les probabilités à priori sont uniformes et appliquons une pénalisation diagonale aux différents coûts d'erreurs tel que :  $C(\text{impasse})=1.5$ ,  $C(\text{Angle})=1.2$ ,  $C(\text{Crois-T})=0.2$ ,  $C(\text{Crois})=1$ ,  $C(\text{porte})=1.1$ ,  $C(\text{Pièce})=0.9$ ,  $C(\text{mur})=1$ ,  $C(\text{Coin})=1$  et  $C(\text{Impasse})=1$ , (voir tableau 5) .

Supposition 2: Dans la table 6 , nous gardons les mêmes coûts d'erreurs mais avec des probabilité à priori

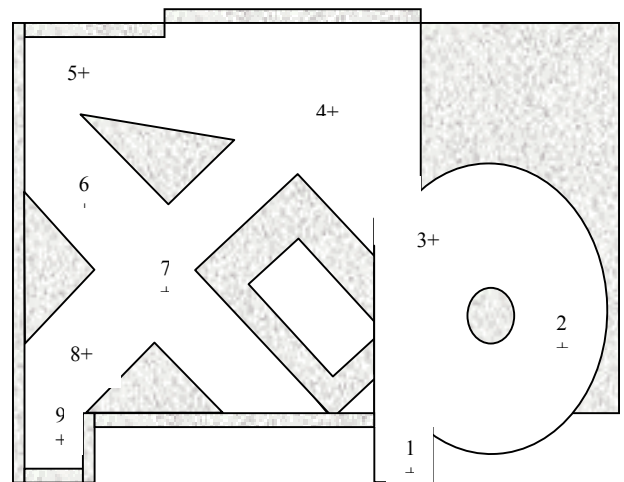


Fig.6. Exemple d'un labyrinthe

différentes tel que :  $P(\text{couloir})=0.10$ ,  $P(\text{Angle})=0.20$ ,  $P(\text{CroisT})=0$ ,  $P(\text{Impasse})=0.10$ ,  $P(\text{mur})=0.10$ ,  $P(\text{Coin})=0.05$ ,  $P(\text{Pièce})=0.1$ ,  $P(\text{Porte})=0.25$  et  $P(\text{Crois})=0.10$ .

Nous n'avons pas décrit tous les positons par limitation du volume de ce papier.

| pos | Situation D'ordre 1 en % | Situation D'ordre 2 en % | Situation D'ordre 3 en % | Situation D'ordre 4 en % | Remarques           |
|-----|--------------------------|--------------------------|--------------------------|--------------------------|---------------------|
| 1   | Impasse 100              | /                        | /                        | /                        | Position Certaine   |
| 2   | Angle 89.5               | porte 5.5                | CroisT 5                 | /                        | Probabilité Evident |
| 4   | Piece 98.4               | Impasse 4.3              | /                        | /                        | Probabilité Evident |
| 6   | porte 59.4               | couloir 34               | CroisT 6.6               | /                        | Equivalence         |
| 8   | couloir 54               | Angle 42.1               | Impasse 3.9              | /                        | Indécision          |

Table. 5. Décision des positions avec supposition 1

Nous remarquons qu'en passant de la table5 à la table 6 , qu'il y'a une grande amélioration du fait que la position 8 est passé d'une indécision à un cas d'équivalence , la position 2 est passée d'un cas évident à un cas certain.

| Pos | Situation<br>D'ordre 1<br>en % | Situation<br>ordre 2<br>en % | Situation<br>ordre 3<br>en % | Situation<br>ordre 4<br>en % | Remarques                                                                                         |
|-----|--------------------------------|------------------------------|------------------------------|------------------------------|---------------------------------------------------------------------------------------------------|
| 1   | Impasse 100                    | /                            | /                            | /                            | Position Certaine<br>Position Certaine<br>Probabilité Evident<br>Position Certaine<br>Equivalence |
| 2   | Angle 100                      | /                            | /                            | /                            |                                                                                                   |
| 4   | Pièce 98                       | Porte 2                      | /                            | /                            |                                                                                                   |
| 6   | porte 100                      | /                            | /                            | /                            |                                                                                                   |
| 8   | couloir 58.3                   | Angle 33.6                   | Impasse 8.1                  | /                            |                                                                                                   |

Table. 6. Décision des positions avec supposition 2

Comme conclusion nous pourrions dire que l'introduction des probabilités à priori et les coûts d'erreurs sont nécessaires pour une bonne décision.

## 5. Conclusion

Cette étude entre dans le cadre général de la classification. Nous avons opté à mettre en œuvre un classifieur bayésien qui utilise des fonctions à base radiale à contextes des moindres carrés orthogonales, car nous pensons qu'il s'adapte bien avec ce type de problème, vue qu'il a donné de bons résultats, en plus il nous a permis de construire des centres ou des clusters tel qu'ils sont dans la base d'apprentissage et fait l'apprentissage en même temps.

Nous avons pu faire introduire les données par une caméras, mais cela pourrait entraîner plusieurs ramifications des images, cela veut dire que l'image 3D d'une caméras représente, plus de corrélations entre les deux plans xoy et yoz. Dans les prochains travaux nous allons voir la réaction des RBF lors de l'introduction des techniques de quantification vectorielles et l'aspect temporel.

Nous estimons que ce travail sert à approfondir la généralité de l'approche neuronale bayésienne dans les applications de classification.

## Bibliographie

- [1] S. A. Alan, *Reconnaissance d'environnement et navigation réactive d'un robot mobile autonome par réseau de neurones*, thèse de doctorat universitaire D'Evry Val d'Essonne, 1996.
- [2] S. Billings and L.Z. Guang, *Radial basis function network configuration using genetic algorithms*, Neural Network, vol8, n°. 6, pp. 877-890, 1995.
- [3] B. Chen et Luo, *Orthogonal least squares method and their application to non linear system*, Inter. Journal of Control, p 1873-1896.
- [4] P. Common, *Classification bayésienne distribuée*, Revue Technique Thomson CSF, Vol. 22, N°4, Dec. 1988.
- [5] V. Demian, *Conception et analyse d'algorithme parallèle pour les réseaux neuronaux de KOHONEN et de fonction à base radiale (RBF)*, thèse doctorat Université Claude Bernard Lyon, 1995.
- [6] A.M. Djafari et G. Demoment, *Utilisation de l'entropie dans les problèmes de restauration et de reconstruction d'images*, Traitement de signal, Vol. 5, N°4, 1988.
- [7] J. Freeman and D. Saad, *Radial basis function networks: generalisation in over realizable and unrealizable scenarios*, Pergaman, Neural Network, vol N°9, pp1521- 1529, 1996.
- [8] J. Herault et C. Jutten, *Réseaux de neuronaux et traitement de signal*, Hermes, 1994.
- [9] B. Horne and R. H. Don, *Progress in supervised neural networks*, IEEE, Signal Processing Magazine, Jan. 1993.
- [10] A. Judistky, O. Lepski, *Confidence intervals for regression estimation*, RR 3580, INRIA, 1998.
- [11] N. B. Karyanis, *Reformulated radial basis function neural network trained by gradient descent*, IEEE transaction on neural network, Vol.10, N°3, pp 657-669, May 1999.
- [12] Z. Mekkakia, L. Mesbahi et M. Lakehal, *initialising of RBF centers by Som Algorithm*, MS'2002, Inter. Conf. On Modelling and Simulation in Technical and Social Sciences, Girona, Catalonia, pp717-723, Spain 25-27, June2002.
- [13] K.T. Mikhali and C.L. Aristidis, *Shared kernel model for class conditional density Estimation*, IEEE transaction on neural network, vol. 12, N°.5, pp. 987-996, September 2001.
- [14] K. Miroslav, *Decision tree can initialize radial basis functions networks*, IEEE transaction on Neural Network, vol 9, n°5, 1998.
- [15] H. Takahiko, *Decision rule for pattern classification by intergrating interval features values*, IEEE transaction on pattern analysis and machine intelligence, vol20, n°4, 1998.
- [16] C. W. Therien, *Decision estimation and classification*, John Willey and Sons, 1989.
- [17] J.T. Tou and R. C. Gonzalez, *Pattern recognition principles*, Addison Wesley Publishing Compagny, 1974.
- [18] G. Zheng and S. Billings, *Radial basis function network configuration using mutual information and the orthogonal least squares algorithms*, Neural Network, vol9, n°9, pp1619-1637, 1996.