

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna
Faculté des Sciences de l'Ingénieur
Département d'Electronique

Thèse

Préparée au
Laboratoire d'Electronique Avancée (LEA) Batna

Par :

Djarah Djalal

Ingénieur d'état en électronique

En vue de l'obtention du diplôme de :
Magister en robotique

Thème

**Application des Réseaux de Neurones
Pour la Gestion d'un Système de Perception
Pour un Robot Mobile D'intérieur**

Soutenu le :/...../2006

Membres du Jury

<i>Ameddah Djamel</i>	<i>Président de Jury</i>	<i>Maître de Conférences Université de Batna</i>
<i>Louchene Ahmed</i>	<i>Rapporteur</i>	<i>Maître de Conférences Université de Batna</i>
<i>Baarir Zineddine</i>	<i>Examineur</i>	<i>Maître de Conférences Université de Biskra</i>
<i>Benoudjit Nabil</i>	<i>Examineur</i>	<i>Maître de Conférences Université de Batna</i>

**Application des Réseaux de Neurones
Pour la Gestion d'un Système de
Perception
Pour un Robot Mobile D'intérieur**

Dédicace

*A la mémoire de mon père.
A toute ma famille et mes amis.*

REMERCIEMENTS

Ce travail a été effectué au sein du laboratoire d'Electronique Avancée –LEA- à L'Université Batna , dirigé par Monsieur Ahmed Louchene maître de conférences à l'Université de Batna que je désire le remercier , pour m'avoir dirigé et soutenu au cours de la réalisation de ce travail.

Sa disponibilité et ses conseils ont été indispensables à la concrétisation de ce travail, Je remercie également mes coéquipiers de travail qui partageaient le même laboratoire et avec lesquels j'ai eu des discussions des plus intéressantes et fructueuses sur ce travail.

SOMMAIRE

Introduction Générale	01
Chapitre 1	
Présentation générale des réseaux de neurones	03
1.1. Introduction	03
1.2. Réseaux de Neurones	04
1.2.1. neurones artificiels	04
1.2.1.1 Présentation	04
1.2.1.2. Description mathématique	04
1.2.1.3 Les variantes	05
1.2.2. Les différents types des RNA	05
1.2.2.1. Les réseaux « fedd-forward »	06
a- Les perceptrons	06
* Le perceptron mono-couche	06
* Le perceptron multicouches (PMC)	06
b- Les réseaux à fonction radiale	06
1.2.2.2. Les réseaux « feedback »	07
a- Les cartes auto-organisatrices de Kohonen	07
b- Les réseaux de Hopfield	07
c- Les ARTs	08
1.2.3. L'apprentissage des RNA	08
1.2.3.1. Les types d'apprentissage	09
* Le mode supervisé	09
* Le renforcement	09
* Le mode non supervisé (ou auto-organisationnel)	09
* Le mode hybride	09
1.2.3.2. Les règles d'apprentissage	10

* Règle de correction d'erreurs	10
* Apprentissage de boltzmann	10
* Règle de Hebb	10
* Règle d'apprentissage par compétitions	10
1.2.4. Quelques exemples d'applications	13
1.2.4.1. La catégorisation de données	13
1.2.4.2. L'approximation de fonctions	13
1.2.4.3. Problèmes de prédiction/prévision	13
1.2.4.4. L'optimisation	13
1.2.4.5. Mémoire adressable par le contenu	14
1.2.4.6. Le contrôle	14
1.2.4.7. La classification de données	14
1.3. Reconnaissance des formes	15
1.3.1. Méthode classique	15
1.3.2. Méthodes modernes ou par réseaux de neurones	15
1.3.3. Methode par série de RN dite type de diabolo	17
1.4 Conclusion	18

Chapitre 2

Système De Perception Pour Robot Mobile	19
2.1. Introduction	19
2.2. perception	19
2.2.1. capteurs en robotique mobile	20
2.2.1.1. les capteurs proprioceptifs	20
2.2.1.2. les capteurs extéroceptifs	25
a- Télémètres à ultrason	26
b- Télémètres à infrarouge	27
c- Télémètres laser	28

d- Modèle probabiliste	30
2.2.1.3. les Cameras	31
a- Cameras simples	31
b- Cameras stéréoscopiques	33
c- Cameras panoramiques	33
2.2.1.4 Autres capteurs	35
a- Les capteurs tactiles	35
b- les boussoles	36
c- les balises	36
d- le GPS	37
2.3. Evitement d'obstacles	37
2.4. Conclusion	38

Chapitre 3

Description du Réseau de Neurones Développé	39
3.1. Introduction	39
3.2. Acquisition des données	41
3.2.1 Collecte des données	41
3.2.2. Analyse des données	46
3.2.3. La mise en forme des données	48
3.2.4. Séparation des bases des données	49
3.3 Conception du réseau de neurones	49
3.3.1. Choix d'un réseau de neurones	49
3.3.1.1. Utilisation de la boîte à outils matlab réseau de neurones	50
3.3.2. L'apprentissage du réseau de neurones	53
3.3.2.1. Bases d'apprentissage	54
3.3.2.2 Convergence	56
3.3.3. Validation et test	61

3.3.3.1. Exemples de base de test	61
3.3.3.2. Etapes de validation	64
a- Taux de reconnaissance	64
b- Résolution des indécisions	66
c- Test d'invariance par déformation, translation et homothétie	66
c-1- Déformation	66
c-2- Translation	68
c-3- Homothétie	69
d- Matrice de confusion	72
3.5. Conclusion	73

Conclusion Générale

Conclusion Générale	74
---------------------	----

Bibliographie

Bibliographie

Introduction Générale

Introduction

L'application des modèles connexionnistes dans des domaines aussi variés que le contrôle de systèmes, la robotique, l'économie ou les sciences cognitives, a permis de mettre en évidence la puissance de ces modèles non linéaires.

La puissance d'un robot mobile autonome réside dans son autonomie ; l'exécution d'une trajectoire est assurée sans l'aide d'un agent externe. Pour doter une telle machine de cette capacité, elle doit être en mesure de se comporter d'une façon intelligente dans des situations imprévues. En effet, des systèmes de perceptions capables de fournir au robot le maximum d'information sur son environnement d'évolution sont indispensables.

Les méthodes modernes de contrôle et de traitement, telles que la logique floue et les réseaux de neurones, nous ont incités à choisir l'une d'elles pour la gestion d'un système de perception pour robot mobile d'intérieur. Les réseaux de neurones sont mieux adaptés aux problèmes de classification. Il s'agit de déduire la forme de l'obstacle se trouvant sur le chemin du robot pour soit générer les commandes pour son évitement, soit procéder à la modélisation de l'environnement.

Dans le présent travail, nous sommes limités à une classification géométrique des obstacles qui entrent dans une zone critique du champs de vision du système de perception. Les données d'entrée du réseau sont obtenues par la simulation d'un télémètre laser monté en tourelle et placé à l'avant du robot mobile.

Le mémoire résumant le travail réalisé est organisé autour de quatre grandes parties. La première partie est une présentation généraliste des réseaux de neurones et de leurs performances en reconnaissance de formes ; à ce niveau on a essayé d'introduire le sujet en insistant sur le vocabulaire et les notions nécessaires à la compréhension de la suite. Le lecteur qui n'est pas familier des

réseaux de neurones pourra l'utiliser comme une base pour entamer des travaux dans ce domaine.

La seconde partie est une présentation générale des systèmes de perception dans le domaine de la robotique mobile ainsi que quelques notions sur l'évitement d'obstacle lors de l'exécution d'une trajectoire sera donné.

La troisième partie constitue le coeur du mémoire : elle contient la description détaillée de la conception du réseau de neurones utilisé. Les résultats des programmes de simulation d'un télémètre développés pour la collecte d'information sur l'environnement du robot sont aussi présentés dans cette partie.

Une conclusion générale avec des perspectives résumera le présent mémoire.

Présentation Générale des Réseaux de Neurones

Présentation Générales des Réseaux de Neurones

1.1. Introduction

Ce chapitre présente le fonctionnement général d'un réseau de neurones ainsi que des exemples de performances que l'on peut obtenir en reconnaissance de formes. Il ne s'agit pas de la présentation d'un travail personnel, mais de l'introduction des connaissances et du vocabulaire nécessaires à la compréhension des parties suivantes de ce document. Les notions utilisées ultérieurement se trouvent toutes dans ces pages qui pourront ainsi servir de référence. Le lecteur familier des réseaux de neurones ne trouvera ici rien de nouveau.

Pourquoi des neurones artificiels? Capter une image, la numériser, la segmenter en éléments de contours, détecter un objet mobile, le reconnaître quelle que soit sa position et estimer sa profondeur. Capter le son d'une voix au milieu d'un brouhaha et du bruit ambiant et reconnaître les mots qui sont prononcés. Capter et analyser les deux cents signaux issus d'un processus industriel et en déduire si tout est conforme ou si une avarie se prépare : voici quelques problèmes pourtant courants dans les sciences de l'informatique mais dont les solutions, encore incomplètes, impliquent de multiples efforts de recherche dans la communauté scientifique.

Les caractéristiques essentielles des réseaux de neurones réels que nous conserverons dans les modèles mathématiques étudiés, concernent le grand nombre de connexions, la non-linéarité des relations entrée-sortie et la faculté de "plasticité" ou d'adaptabilité. Ces caractéristiques, même simplifiées, leur confèrent déjà de multiples possibilités en traitement des signaux et des informations ainsi que la faculté d'apprendre à classer, à reconnaître des formes ou à réaliser des tâches complexes.

1.2. Réseaux De Neurones

1.2.1. Neurones Artificiels

Par définition, un réseau de neurones est un assemblage d'éléments, d'unités ou de noeuds processeurs pour lequel un sous-groupe fait un traitement indépendant et passe le résultat à un deuxième sous-groupe [3]. Les capacités de traitement du réseau sont stockées dans les forces (ou poids) des connections inter unités qui est obtenu par un processus d'adaptation ou d'apprentissage a un ensemble de patrons de référence. Ces réseaux sont un type d'intelligence artificielle qui tente d'imiter le fonctionnement d'un cerveau humain. Au lieu d'utiliser un model digital, dans lequel toutes les opérations manipulent des zéros et des uns, un réseau neuronique procède en créant des connections entre les noeuds de traitement - l'équivalent informatique d'un neurone. L'organisation et les poids déterminent les sorties.

1.2.1.1. Présentation

Le neurone formel est donc une modélisation mathématique qui reprend les grands principes du fonctionnement du neurone biologique et particulièrement, la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même «valeur» [2], [4], [5] (pour simplifier disons que les connexions entre les neurones sont plus ou moins fortes), les auteurs ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (coefficients de pondération).

1.2.1.2. Description mathématique

On peut donc dire que pour un nombre compris entre j ($=1$) et un nombre quelconque n , le neurone formel va calculer la somme de ses entrées ($x_1, \dots, x_n$), pondérées par les poids synaptiques ($w_1, \dots, w_n$) et la comparée à son seuil téta. Si le résultat est supérieur au seuil, alors on renvoie **1**, sinon on renvoie **0** [5].

La formule mathématique s'écrit comme :

$$y = f\left(\sum_{j=1}^n w_j x_j - \theta\right)$$

1.2.1.3. Les variantes

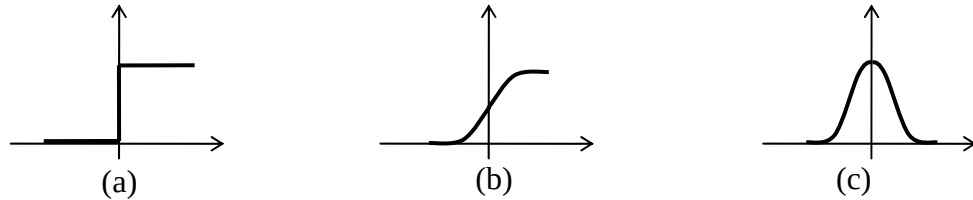


Figure 1.1. Les fonctions d'activation
 (a) : fonction à seuil, (b) : sigmoïde, (c) : gaussienne

Dans sa première version, le neurone formel était donc implémenté avec une fonction à seuil (a), mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations[3], comme des sigmoïdes (b) ou des gaussiennes (c) tel qu'il est illustré par la figure 1-1.

Il est à noter que les fonctions d'activation sigmoïdes sont de loin les plus utilisées dans les réseaux de neurones artificiels.

1.2.2. Les Différents Types Des RNA

On va reprendre ici la classification précédente en séparant les réseaux « feed-forward » et les réseaux « feedback » [2]. Pour ce qui est des modes et des règles d'apprentissage on essaye de donner un aperçu que nous jugeons suffisant au niveau des sections suivantes.

1.2.2.1. Les réseaux "feed-forward" :

a- Les Perceptrons

o Le Perceptron mono-couche

C'est historiquement le premier RNA, c'est le Perceptron de Rosenblatt. C'est un réseau simple, puisque il ne se compose que d'une couche d'entrée et d'une couche de sortie. Il est calqué, à la base, sur le système visuel et de ce fait il a été connu dans un but premier de reconnaissance des formes. Cependant, il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples (telle "ET" ou "OU"). Sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur[3] (ou selon la règle de Hebb) .

o Le Perceptron multicouches (PMC) :

C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté pour les couches d'entrée et de sortie) et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués, et notamment le fameux problème du XOR [4]. Il suit aussi un apprentissage supervisé selon la règle de correction de l'erreur.

b- Les réseaux à fonction radiale

Ce sont les réseaux que l'on nomme aussi RBF (pour "Radial Basic Functions"). L'architecture est la même que pour les PMC cependant, les fonctions de base utilisées ici sont des fonctions Gaussiennes. Les RBF seront donc employés dans les mêmes types de problèmes que les PMC à savoir, en

classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RBF est le mode hybride [5] et les règles sont soit, la règle de correction de l'erreur soit, la règle d'apprentissage par compétition.

1.2.2.2. Les réseaux "feedback" :

a- Les cartes auto-organisatrices de Kohonen :

Ce type de réseau, aussi appelé SOM « Self Organised Maps », est un réseau à apprentissage non supervisé, un apprentissage qui établit une carte discrète, ordonnée topologiquement en fonction de patterns d'entrée. Le réseau forme ainsi une sorte de treillis où, chaque noeud du treillis est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation [3].

Les LVQ (pour « Learning Vector Quantization ») sont un cas particulier des SOM, où seul le noeud ayant la meilleure corrélation est adapté. On parle, pour ce type de réseau où seul le « vainqueur » est sélectionné, de réseaux à compétition.

b- Les réseaux de Hopfield :

Les réseaux de Hopfield sont donc des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non supervisé.

c-. Les ARTs :

Les réseaux ART "Adaptive Resonance Theory" sont des réseaux à apprentissage par compétition. Le problème majeur qu'il se pose dans ce type de réseaux est le dilemme « stabilité/plasticité ». En effet, dans un apprentissage par compétition, rien ne garantit que les catégories formées vont rester stables. La seule possibilité, pour assurer la stabilité, serait que le coefficient d'apprentissage tende vers zéro, mais le réseau perdrait alors sa plasticité [2]. Les ART ont été connus spécifiquement pour contourner ce problème. Dans ce genre de réseau, les vecteurs de poids ne seront adaptés que si l'entrée fournie est suffisamment proche, d'un prototype déjà connu par le réseau. On parlera alors de résonance. A l'inverse, si l'entrée s'éloigne trop des prototypes existants, une nouvelle catégorie va alors se créer, avec pour prototype, l'entrée qui a engendré sa création. Il est à noter qu'il existe deux principaux types de réseaux ART : les ART-1 pour des entrées binaires et les ART-2 pour des entrées continues. Le mode d'apprentissage des ART peut être supervisé ou non.

1.2.3. L'apprentissage des RNA

Pour un RNA, l'apprentissage peut être considéré comme le problème de la mise à jour des poids des connexions au sein du réseau [6], afin de réussir la tâche qui lui est demandée.

L'apprentissage est la caractéristique principale des RNA et il peut se faire de différentes manières et selon différentes règles.

1.2.3.1. Les Types D'apprentissage :

o Le mode supervisé :

Dans cette sorte d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie [20]. Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie, c'est-à-dire celle attendue, correspondant à une entrée donnée.

o Le renforcement :

Le renforcement est en fait une sorte d'apprentissage supervisé et certains auteurs le classe d'ailleurs, dans la catégorie des modes supervisés. Dans cette approche le réseau doit apprendre la corrélation entrée/sortie via une estimation de son erreur, c'est-à-dire du rapport échec/succès. Le réseau va donc tendre à maximiser un index de performance [21] qui lui est fourni, appelé signal de renforcement. Le système étant capable ici, de savoir si la réponse qu'il fournit est correcte ou non, mais il ne connaît pas la bonne réponse.

o Le mode non supervisé (ou auto-organisationnel) :

Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statistiques de l'entrée et établir des catégories, en attribuant et en optimisant une valeur de qualité, aux catégories reconnues.

o Le mode hybride :

Le mode hybride reprend en fait les deux autres approches, puisque une partie des poids va être déterminée par apprentissage supervisé et l'autre partie par apprentissage non supervisé.

1.2.3.2. Les Règles d'apprentissage :

o Règle de correction d'erreurs :

Cette règle s'inscrit dans le paradigme d'apprentissage supervisé, c'est-à-dire, dans le cas où on fournit au réseau, une entrée et la sortie correspondante. Si on considère y , la sortie calculée par le réseau et d , la sortie désirée, le principe de cette règle est d'utiliser l'erreur $(d-y)$, afin de modifier les connexions et de diminuer ainsi l'erreur globale du système. Le réseau va donc s'adapter jusqu'à ce que y soit égal à d .

o Apprentissage de Boltzmann :

Ce qu'il faut savoir tout d'abord, c'est que les réseaux de Boltzmann sont des réseaux symétriques récurrents et qu'ils possèdent deux sous-groupes de cellules, le premier étant relié à l'environnement (cellules dites visibles) et le second ne l'étant pas (cellules dites cachées). Cette règle d'apprentissage est de type stochastique (= qui relève partiellement du hasard) et elle consiste à ajuster les poids des connexions, de telle sorte que l'état des cellules visibles satisfasse une distribution probabiliste souhaitée.

o Règle de Hebb :

Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique va aller croissant. Il est à noter ici que l'apprentissage est localisé, c'est-à-dire que la modification d'un poids synaptique w_{ij} ne dépend que de l'activation d'un neurone i et d'un autre neurone j .

o Règle d'apprentissage par compétitions :

La particularité de cette règle, c'est qu'ici l'apprentissage ne concerne qu'un seul neurone. Le principe de cet apprentissage est de regrouper les données en catégories. Les patrons similaires vont donc être rangés dans une même classe, en

se basant sur les corrélations des données, et seront représentés par un seul neurone, on parle de « winner-take-all ».

Dans un réseau à compétition simple, chaque neurone de sortie est connecté aux neurones de la couche d'entrée, aux autres cellules de la couche de sortie (connexions inhibitrices) et à elle-même (connexion excitatrice) [3]. La sortie va donc dépendre de la compétition entre les connexions inhibitrices et excitatrices.

Paradigme	Règle d'apprentissage	Architecture	Algorithme	Tâches
Supervisé	Correction d'erreur	Perception simple ou multi-couches	Perception rétro propagation, adaline, madaline	Classification, approximation de fonctions, prédiction, contrôle
	Bolzman	Récurrente	Apprentissage de Boltzmann	Classification
	Hebb	Multi-couches non bouclés	Analyse de discriminants linéaires	Analyse de données, classification
	Par compétition	à compétition ART	LVQ ARTMap	Catégorisation au sein d'une classe, compression de données Classification, catégorisation au sein d'une classe
Non supervisé	Correction d'erreur	Multi-couches non bouclés	Projection de Sammon	Analyse de données
	Hebb	à compétition	Analyse en composantes principales	Analyse de données, compression de données
	Par compétition	Cartes de Kohonen	VQ	Catégorisation, compression de données
		ART	SOM ART-1, ART-2	Catégorisation, analyse de données Catégorisation
Hybride	Correction d'erreur et par compétition	RBF	RBF	Classification, approximation de fonctions, prédiction, contrôle

1.2.4. Quelques exemples d'applications

En dehors de la recherche fondamentale en intelligence artificielle et de la tentative de modélisation du cerveau, les RNA sont utilisés concrètement dans de nombreux cas différents, dont nous allons voir quelques exemples.

1.2.4.1. La catégorisation de données :

Le problème ici est de rassembler des données similaires en une même catégorie. Les RNA à apprentissage non-supervisé réalisent ce type de tâche et seront donc employés dans la compression de données ou l'exploration d'analyse de données par exemple.

1.2.4.2. L'approximation de fonctions :

Les capacités de classification et de catégorisation des RNA sont encore exploitées ici, puisqu'ils servent parfois à approximer une fonction inconnue (dont on connaît par contre les paires d'entrées/sorties) à une fonction connue. Ce type de difficultés est très fréquent dans le travail de modélisation ou dans le domaine de l'ingénierie.

1.2.4.3. Problèmes de prédiction/prévision :

Les RNA sont également utilisés lorsqu'il faut, à partir de données recueillies à un temps t , prédire les données à un temps $t + 1$. Les RNA intéressent donc les domaines de la science et de l'ingénierie mais aussi l'économie de marchés ou la météorologie.

1.2.4.4. L'optimisation :

Pour faire de l'optimisation, le problème est soit, de maximiser le succès soit, de minimiser l'erreur. Encore une fois, les réseaux à apprentissage non-supervisé sont particulièrement adaptés pour ce type de tâche et vont donc intéresser des domaines aussi différents que les mathématiques, les statistiques, l'ingénierie, la médecine ou l'économie.

1.2.4.5. Mémoire adressable par le contenu :

Les RNA sont capables de rappeler une donnée, à partir d'une donnée d'entrée partielle ou bruitée. Ils représentent donc une mémoire associative (ou à contenu adressable) pour laquelle une information est accessible via son contenu [6]. Les RNA peuvent donc s'avérer utiles dans l'établissement d'une base de données multi-média par exemple.

1.2.4.6. Le contrôle :

Considérons un système dynamique pour lequel on veuille contrôler la sortie. Les RNA vont être capables de comparer la sortie du système avec un modèle contrôle et d'adapter ce système, pour retrouver la bonne sortie.

1.2.4.7. La classification de données :

Un des grands problèmes de la classification de données, est d'associer à un pattern d'entrée, un pattern de sortie, parmi de nombreux patterns pré-spécifiés. Comme nous l'avons vu, les RNA sont capables d'apprendre l'association entrée/sortie et sont donc tout a fait aptes à résoudre ce type de problème. Du fait de cette capacité, les RNA vont donc être utilisés par exemple dans, la reconnaissance de caractères [8], [10], la reconnaissance vocale, la classification de courbes d'électroencéphalogramme, la classification de cellules sanguines ou encore la carte descriptive d'un circuit imprimé.

1.3. Reconnaissance des formes

IL y a plusieurs techniques pour la reconnaissance de forme ou bien d'ordre général la classification, on cite trois méthodes avec un exemple pour bien comprendre ces techniques.

1.3.1. Méthode classique

La première, la plus classique, consiste simplement à mesurer la distance euclidienne entre l'image de test et toutes les images présentes dans la base d'apprentissage afin d'en déterminer la plus faible. L'avantage principal réside dans une programmation extrêmement rapide et un temps d'accès aux informations utiles immédiat [9], [11] (pas de nécessité d'apprentissage). De plus, l'incrémentation est facile puisque pour de nouvelles données ou personnes à ajouter à la base, seules les transformations en image nettes de références sont nécessaires. Par contre l'inconvénient majeur réside dans le temps de calcul nécessaire au parcours de toutes les images de la base pour effectuer le calcul de distance qui dépend de la taille des images et du nombre d'individus.

1.3.2. Méthodes modernes ou par réseau de neurones

La seconde repose sur un réseau de neurone discriminant. La méthode du réseau discriminant repose sur l'utilisation d'un unique réseau de neurones donnant en sortie la classe d'appartenance (dans cet exemple le nom de la personne) d'une image inconnue en entrée. Il est composé d'une couche d'entrée, d'une couche cachée et de plusieurs sorties pour autant de personnes à discriminer.

Il y a donc autant de cellules d'entrées que de pixels composant les images. Ceci implique que toutes les images sont de tailles identiques. Afin de réduire la complexité du réseaux.

La couche cachée est composée d'un nombre plus ou moins important de neurones (entre 10 et 100) dont dépendra le temps nécessaire à l'apprentissage et le résultat.

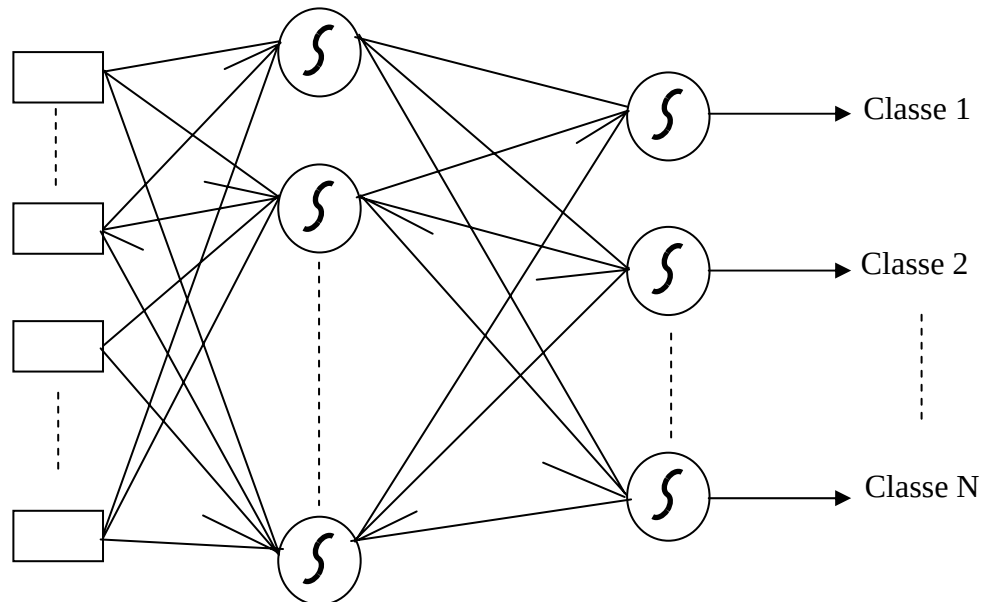


Figure 1.2 structure d'un réseau de classification.

Le principal avantage par rapport au calcul des distances euclidiennes réside dans le temps de calcul nécessaire à l'identification. En effet, celui-ci n'est lié qu'à la complexité du réseau de neurones. Mais même avec de nombreuses cellules sur la couche cachée, la propagation de l'entrée vers la sortie se fait de manière quasi-instantanée (en ms). La taille des données stockées est également fortement réduite, puisque le réseau (en Ko) remplace tout besoin d'images.

Par contre, cette méthode nécessite un très long apprentissage qui dépend du nombre de neurones sur la couche cachée et surtout du nombre de classes en sortie. Avec 33 personnes à discriminer et 120 neurones sur la couche cachée, il faut compter un minimum de 48 heures de calculs avec un P4 pour les réseaux les plus simples !! Ceci est dû à un développement de l'application sous Matlab, ce qui permet d'espérer un gain conséquent en C. On imagine alors la difficulté pour ajouter une nouvelle personne à la base. Dans ce cas, un réapprentissage total du réseau est nécessaire. Un autre type de réseau a donc été testé.

1.3.3. Méthode par série de RN- dite type diabolo-

La troisième méthode est ainsi basée sur une série de réseaux de neurones de type diabolo. Plutôt que de chercher à discriminer chaque classe, une modélisation de chaque individu est ici réalisée. La particularité de ce réseau réside dans un nombre de sorties identique au nombre d'entrées, le but étant alors d'obtenir une sortie la plus proche possible de l'entrée dans le cas d'une image appartenant à la bonne classe. Chaque personne est donc stockée dans un réseau propre (entre 50 et 200 Ko).

Concrètement, chaque nouvelle image à tester est propagée dans les différents réseaux diabolos puis une mesure de distance entre le vecteur d'entrée et de sortie est réalisée. Puisqu'il s'agit d'un réseau modélisateur, plus cette distance sera faible, meilleur sera la probabilité d'appartenance à la classe testée. La mesure la plus faible est alors retenue pour le choix de la personne.

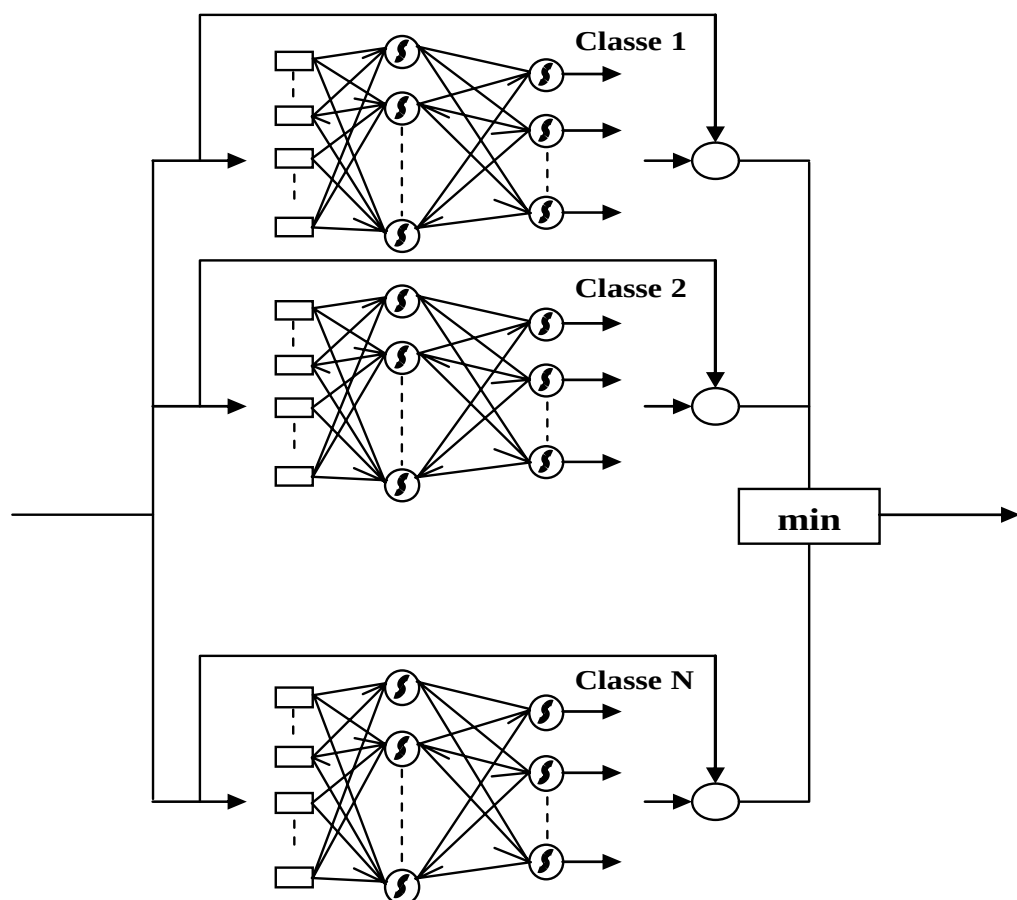


Figure 1.3 Reconnaissance des formes on utilisant un réseau type diabolo

Le temps de calcul nécessaire à l'identification est légèrement plus élevé qu'avec la méthode précédente mais reste tout à fait faible en comparaison de la méthode de calcul des distances euclidiennes à chaque image. Le grand avantage apporté par l'utilisation de réseaux modélisateurs réside dans la simplicité et la rapidité d'incrémentation.

Seul l'apprentissage du réseau de la nouvelle personne est ainsi nécessaire pour agrandir la base. De plus, les réseaux étant indépendants les uns des autres, un apprentissage en parallèle devient possible augmentant la rapidité des tests de configuration, surtout si l'on ajoute à cela une complexité réduite de l'architecture. Enfin, les résultats d'identification sont meilleurs avec cette méthode qu'avec la précédente.

1.4. Conclusion

Après cette présentation générale on a pu montrer que les réseaux de neurones reposent à Présent sur des bases mathématiques solides qui permettent d'envisager des applications dans presque tout les domaines y compris industriel et à grande échelle, notamment dans le domaine de la classification . Si la résolution des Problèmes difficiles nécessite toujours et nécessitera encore très longtemps Beaucoup de travail et un éventail étendu de connaissances en statistiques, traitement du signal, automatique..., etc. il n'est pas douteux que les réseaux de neurones peuvent alléger considérablement la tâche de toute personne travaillant dans ces domaines en permettant une approche efficace et générique des problèmes non linéaires.

Système De Perception Pour Robot Mobile

Système de Perception Pour Robot Mobile

2.1. Introduction

La détection d'obstacle est un des problèmes clef de la robotique, tout robot mobile devant évoluer dans un environnement dynamique doit être capable de détecter et éviter les obstacles de manière autonome, c'est un exemple simple et intéressant vu que le sujet sur le quel nous travaillons touche principalement le domaine de la robotique et plus précisément l'évitement d'obstacle ce dernier nécessite une bonne perception, ce qui nous a poussé à détailler cette notion (Perception) dans ce qui suit

2-2 Perception

La notion de *perception* en robotique mobile est relative à la capacité du système à recueillir, traiter et mettre en forme des informations utiles au robot pour agir et réagir dans le monde qui l'entoure [15], [17]. Alors que pour des tâches de manipulation on peut considérer que l'environnement du robot est relativement structure, ce n'est plus le cas lorsqu'il s'agit de naviguer de manière autonome dans des lieux très partiellement connus. Aussi, pour extraire les informations utiles à l'accomplissement de sa tâche, il est nécessaire que le robot dispose de nombreux capteurs mesurant aussi bien son état interne que l'environnement dans lequel il évolue. Le choix des capteurs dépend bien évidemment de l'application envisagée. Pour se focaliser sur le problème de navigation, nous allons nous restreindre dans ce chapitre aux capteurs utiles à cette tâche. Dans un second temps, nous expliquerons comment utiliser les données issues de ces capteurs pour obtenir une représentation fiable de l'état du système et de l'environnement.

2.2.1. Capteurs en robotique mobile

En robotique mobile, on classe traditionnellement les capteurs en deux catégories selon qu'ils mesurent l'état du robot lui-même ou l'état de son environnement. Dans le premier cas, à l'image de la perception chez les êtres vivants, on parle de *proprioception* et donc de *capteurs proprioceptifs*. On trouve par exemple dans cette catégorie les capteurs de position ou de vitesse des roues et les capteurs de charge de la batterie. Les capteurs renseignant sur l'état de l'environnement, donc de ce qui est extérieur au robot lui-même, sont eux appelés *capteurs extéroceptifs*. Il s'agit de capteurs donnant la distance du robot à l'environnement, la température, signalant la mise en contact du robot avec l'environnement, etc.

L'étude détaillée des capteurs, qui relève à la fois de la physique, de l'électronique et du traitement du signal, ne sera pas vue ici. Nous nous contenterons d'expliquer les principes de fonctionnement des capteurs présents. On tachera simplement de garder à l'esprit que les défauts inhérents aux différents systèmes de mesure utilisés (bruit, erreurs ou échecs de mesures, difficulté de modélisation) influent fortement sur la perception que le robot a de l'environnement.

Nous présentons dans cette section les capteurs les plus couramment utilisés en robotique mobile pour les besoins de la navigation.

2.2.1.1. Les capteurs proprioceptifs

Les capteurs proprioceptifs permettent une mesure de certaines paramètres internes au système lui-même [16]. Dans le cas de la robotique mobile, ces capteurs sont souvent utilisés pour estimer la position courante du robot lors de son déplacement ; c'est ce qui est connu dans la littérature par *odométrie*

➤ L'odométrie

Elle permet d'estimer le déplacement à partir de la mesure de rotation des roues (ou du déplacement des pattes). La mesure de rotation est en général effectuée par un codeur optique disposé sur l'axe de la roue, ou sur le système de transmission (par exemple sur la sortie de la boîte de vitesse). Le problème majeur de cette mesure est que l'estimation du déplacement fournie dépend très fortement de la qualité du contact entre la roue et le sol [16]. Elle peut être relativement correcte pour une plate-forme à deux roues motrices sur un sol plan de qualité uniforme, mais est en général quasiment inutilisable seule pour un robot à chenille par exemple. Notons cependant que l'erreur de ces méthodes se retrouve en général principalement sur l'estimation de la direction du robot, tandis que la mesure de la distance parcourue est souvent de meilleure qualité.

Ces modèles vont donner, en fonction de la mesure réalisée, la probabilité du déplacement réel. Il existe divers types de modèles, mais les plus simples et les plus utilisés sont des modèles de bruit gaussiens sur la direction et la longueur du déplacement, ainsi que sur le changement de direction du robot. Les écarts types de ces différentes gaussiennes dépendent de la valeur de la mesure : l'erreur sur la longueur du déplacement est par exemple proportionnelle à cette longueur (Figure 2.1).

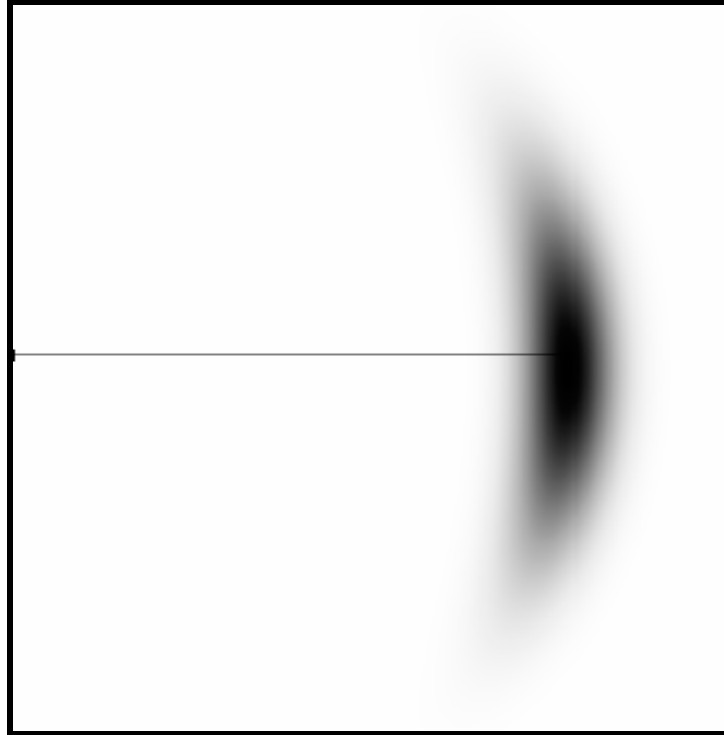


Figure 2.1- Exemple de modèle probabiliste simple de l'odométrie. Le niveau de gris représente la probabilité de la position après un déplacement rectiligne vers la droite. Le modèle utilise un bruit gaussien sur la longueur du déplacement et sur la direction du déplacement.

Il est bien sûr possible d'utiliser des modèles beaucoup plus fins de l'odométrie, par exemple en faisant une hypothèse de bruit gaussien sur la mesure de rotation de chaque roue, et en en déduisant l'erreur de déplacement du robot.

Comme nous le verrons au chapitre sur la localisation, ces modèles probabilistes peuvent être utilisés pour générer des positions possibles du robot selon la distribution de probabilité déduite de la mesure de l'odométrie. L'application successive de plusieurs modèles probabilistes déduits de plusieurs mesures d'odométrie permet de les combiner et d'estimer la distribution de probabilité de position finale du robot (Figure 2.2).

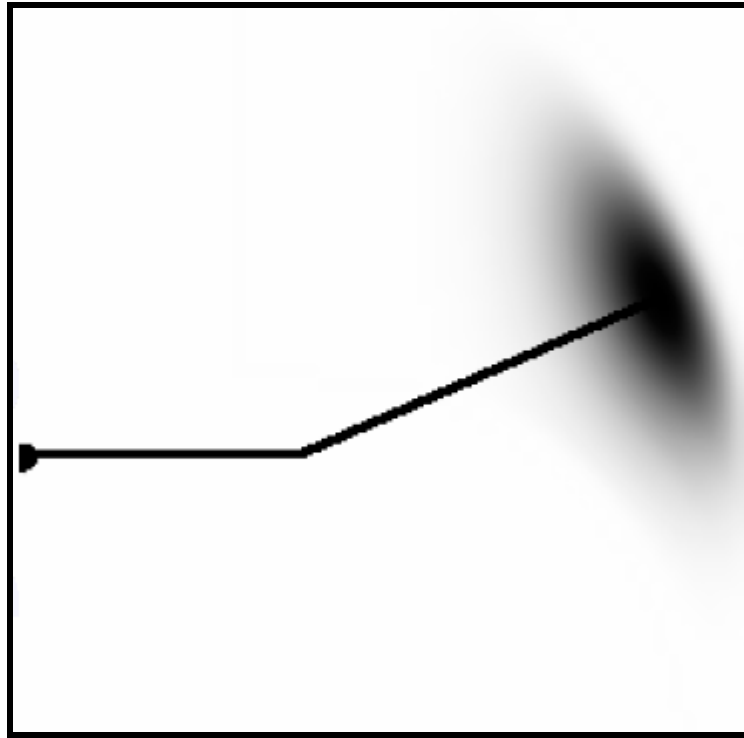


Figure 2.2. *Exemple de combinaison de mesures pour les modèles probabilistes de l'odométrie*

➤ **Les systèmes radar doppler**

Au lieu de mesurer le déplacement par des mesures sur les roues, il est possible d'utiliser un petit radar pointé vers le sol qui permet de mesurer la vitesse du véhicule par effet Doppler. Ce système présente l'avantage d'être beaucoup plus précis que la mesure passant par les roues, et d'être indépendant des dérapages possibles de ces roues [15], mais est en général plus cher et encombrant. Il est de plus très rare sur les petites plates-formes car il ne peut mesurer de faibles vitesses de déplacement.

➤ **Les systèmes inertiels**

La mesure de déplacement potentiellement la plus fiable provient de la mesure des accélérations de la plate-forme par des capteurs inertiels. Cette mesure est potentiellement fiable car elle ne dépend pas de la nature locale de l'environnement,

cependant les capteurs inertiels sont tous entachés de bruit de mesure qui produit une dérive de l'estimation de la position au cours du temps [15].

La qualité des mesures inertielles dépend très fortement du type de capteurs utilisés. Historiquement, les premiers capteurs ont été réalisés à base de systèmes mécaniques et peuvent fournir des mesures extrêmement précises, au prix d'un coût et d'une masse très élevés. Ces dernières années ont vu apparaître de nouvelles technologies de capteurs, notamment basés sur les techniques de microélectronique, qui ont permis la réalisation de capteurs inertiels "bas coût" et l'apparition de ces capteurs dans des produits grand public. La précision de ces capteurs est toutefois de quelques ordres de grandeur plus faible, ce qui rend leur utilisation isolée quasiment impossible. Ces capteurs fournissent toutefois un très bon complément à l'odométrie, notamment pour l'estimation de la direction.

L'accélération en translation de la plate-forme est mesurée par des accéléromètres linéaires. On dispose en général deux accéléromètres pour prendre des mesures dans deux directions perpendiculaires du plan de déplacement du robot.

Un troisième peut être disposé verticalement afin de mesurer la position en trois dimensions.

L'accélération angulaire est mesurée par des gyromètres. On dispose en général un gyromètre selon l'axe vertical, qui permet ainsi de mesurer l'angle de lacet du robot. Deux autres gyromètres peuvent être positionnés selon deux axes du plan de déplacement afin d'estimer la direction en trois dimensions.

Il est également possible de mesurer la rotation du robot par rapport à un axe de référence en utilisant un gyroscope. Cette mesure s'effectue en général par rapport à un axe de référence mis en rotation et isolé mécaniquement le plus possible du

robot, ce qui rend sa direction indépendante de la direction du robot. Cette mesure peut être moins bruitée que l'intégration du signal d'accélération mais dépend très fortement de la qualité de la réalisation mécanique du système, qui dépend très directement du prix du gyroscope.

L'ensemble de ces éléments (3 accéléromètres et 3 gyromètres) peut être réuni pour former une centrale inertielle qui permet d'estimer complètement les six degrés de libertés de la position dans un espace à 3 dimensions. Les centrales inertielles "bas coût" sont cependant aujourd'hui de qualité insuffisante pour une utilisation isolée, tandis que les centrales de qualité correcte restent très chères. Ce domaine est cependant en évolution rapide avec l'arrivée de nouvelles technologies et l'apparition de centrales "bas coût" de qualité correcte devrait se faire dans les prochaines années.

L'utilisation des données fournies par ce type de senseurs passe aussi en général par un modèle probabiliste, qui peut être du type de celui présenté pour l'odométrie.

2.2.1.2. Capteurs extéroceptifs

Quelle que soit la technologie utilisée pour effectuer la mesure telemetrique, le capteur retourne généralement deux informations .La première donne l'angle de gisement, c'est-à-dire la direction dans laquelle a été faite la mesure. La seconde donne la distance au corps ayant réfléchi l'onde émise [17]. Cette technique de mesure permet donc de positionner les objets présents dans la scène par rapport au robot. Elle se prête très bien aux environnements d'intérieur, structures, comportant des formes régulières et statiques comme des murs, qui par ailleurs, possèdent généralement de bonnes propriétés de réflexion.

Il existe différents types de télémètres, qui permettent de mesurer la distance à l'environnement, utilisant divers principes physiques.

a- Télémètres à ultrason

Les télémètres à ultrason sont historiquement les premiers à avoir été utilisés. Ils utilisent la mesure du temps de retour d'une onde sonore réfléchiée par les obstacles pour estimer la distance [16] (Figure 2.3). Ces télémètres sont très simple et peu cher, et sont donc très répandus, mais possèdent de nombreux inconvénients.

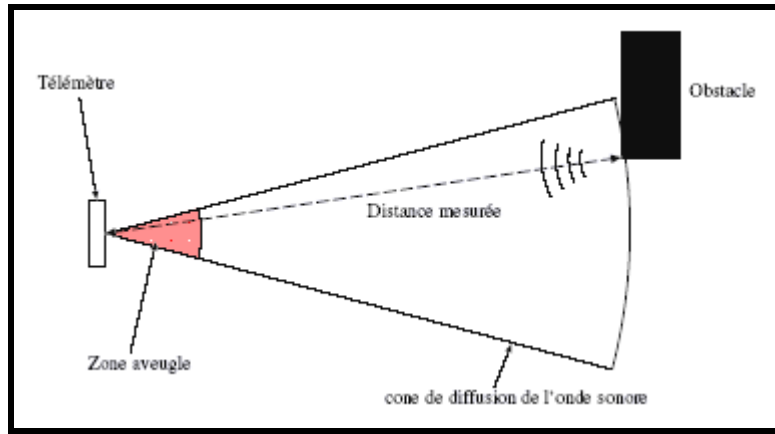


Figure 2.3. Principe du télémètre à ultrasons.

$$P(\text{scan} | \text{Obstacles}) = \prod_{i=1}^M P(\text{mesure}_i | \text{distance obstacles})$$

En premier lieu, deux télémètres voisins ne peuvent être utilisés simultanément, car il est impossible de savoir par lequel des deux télémètres une onde réfléchiée a été émise (phénomène de "crosstalk"). Un robot possédant plusieurs télémètres doit donc les activer l'un après l'autre, ce qui entraîne un taux de rafraîchissement global des mesures relativement faible.

Ces télémètres possèdent une "zone aveugle", de quelques dizaines de centimètres, en dessous de laquelle ils ne peuvent détecter les obstacles. Cette zone est due à une temporisation entre l'émission de l'onde sonore et le début de la détection de l'onde réfléchiée qui est nécessaire pour ne pas perturber cette mesure.

De plus, l'onde réfléchie est très sensible aux conditions environnementales locales. Ainsi, si l'angle entre l'obstacle et la direction de l'onde sonore est trop faible, il n'y aura pas de retour de l'onde sonore et l'obstacle ne sera pas perçu.

L'onde de retour dépend également de la texture de l'obstacle. Un mur couvert de moquette pourra par exemple ne pas être détecté.

Les télémètres ultrason détectent les obstacles se situant dans un cône relativement large (d'angle au sommet d'environ 30 degrés). Cette caractéristique peut être à la fois un avantage et un inconvénient. C'est un inconvénient car un obstacle détecté n'est pas localisé en angle à l'intérieur du cône de détection, et on obtient donc une mesure de la position relativement imprécise. C'est par contre un avantage car des éléments relativement fins (les pieds de table ou de chaise par exemple) sont détectés dans ce cône, alors qu'ils pourraient ne pas être détectés par des télémètres ayant un angle d'ouverture très fin [15].

b- Télémètres à infrarouge

Ces télémètres possèdent l'avantage d'avoir un cône de détection beaucoup plus restreint. Il utilisent une lumière infrarouge au lieu d'une onde sonore pour la détection et peuvent être basés sur différentes techniques qui permettent de recueillir plus ou moins d'information [16]. Il est possible de mesurer simplement le retour ou le non-retour d'une impulsion codée, ce qui permet de détecter la présence ou l'absence d'un obstacle dans une certaine portion de l'espace.

Il est également possible de réaliser une triangulation sur le faisceau de retour de l'onde lumineuse, ce qui permet d'avoir une mesure de la distance de l'obstacle.

Les inconvénients de ces télémètres sont liés à leur portée, en général relativement restreinte, et à leur sensibilité aux fortes sources de lumières qui contiennent un fort rayonnement infrarouge. Un projecteur du type de ceux utilisés

pour la télévision pointé sur le robot, par exemple, sature en général complètement le récepteur et empêche toute détection d'obstacle. Ils sont également très sensibles à la couleur et à la nature de la surface de l'obstacle (par exemple, ils détectent difficilement les vitres).

c- Télémètres laser

Les télémètres les plus utilisés à l'heure actuelle pour des applications de cartographie et de localisation sont les télémètres laser à balayage. Ils utilisent un faisceau laser mis en rotation afin de balayer un plan, en général horizontal, et qui permet de mesurer la distance des objets qui coupent ce plan (Figure 2. 4).

Cette mesure peut être réalisée selon différentes techniques (mesure du temps de retour, interférométrie...).

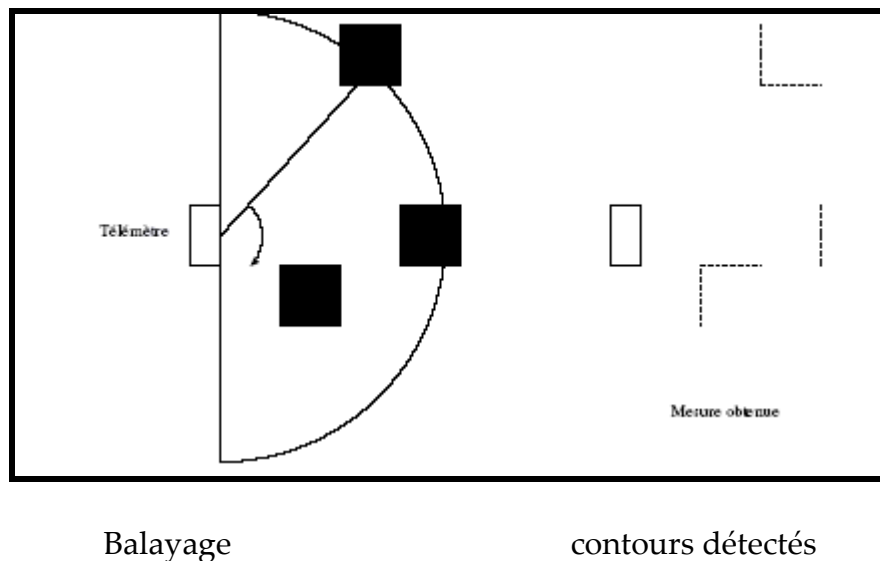


Figure 2.4. Illustration d'un télémètre Laser.

Les télémètres les plus courants ont une bonne résolution angulaire car ils permettent d'obtenir une mesure de distance tous les demi-degrés, sur une zone de 180 ou 360 degrés selon les modèles [15]. La mesure est de plus relativement précise

(avec un bruit de l'ordre de quelques centimètres) à une distance relativement grande (plusieurs dizaines de mètres). La fréquence d'acquisition est en général de l'ordre de la dizaine de Hertz, voire proche de la centaine pour certains modèles. Ces télémètres sont très utilisés en environnement intérieur car ils fournissent des données abondantes et précises sur la position des objets caractéristiques de l'environnement tels que les murs. Ils possèdent toutefois un certain nombre d'inconvénients.

En premier lieu, leur zone de perception est restreinte à un plan et ne permet donc pas de détecter les obstacles situés hors de ce plan (un petit objet posé au sol par exemple). Ils ne peuvent pas non plus détecter les objets ne réfléchissant pas correctement la lumière du laser (en premier lieu les vitres, mais aussi certains objets très réfléchissants, tels que les objets chromés). Pour limiter ces inconvénients, il est possible de les utiliser en conjonction avec des capteurs à ultrason qui ont un cône de détection plus large et qui peuvent détecter les vitres.

Enfin, la plupart des algorithmes de cartographie et de localisation existants supposent que le plan de mesure du télémètre laser reste horizontal et à hauteur

constante, ce qui n'est plus vrai en cas de sol irrégulier ou, dans la majorité des cas, en extérieur.

A l'heure actuelle, des télémètres laser balayant l'espace selon deux axes commencent à apparaître. Ils permettent ainsi d'obtenir une image de distance selon un angle solide de l'ordre de quelques dizaines de degrés dans les deux dimensions.

Ces télémètres restent toutefois chers et fragiles du fait de la mécanique nécessaire au balayage. De plus, la fréquence d'acquisition est relativement faible (de l'ordre de quelques Hertz), ce qui pose problème lorsque le robot est en mouvement.

Des systèmes sans balayage permettant d'obtenir une image de profondeur est également en cours de développement. Il restent aujourd'hui du domaine de la recherche mais sont d'un très grand intérêt potentiel pour la robotique mobile.

d- Modèle probabiliste

Les modèles probabilistes associés à ces capteurs permettent de donner la probabilité de la mesure en fonction de la distance réelle de l'obstacle. Les probabilités sont estimées pour chacune des mesures individuelles prises depuis une position, puis agglomérées par produit :

$$P(\text{scan} \mid \text{Obstacles}) = \prod_{i=1}^M P(\text{mesure}_i \mid \text{distance obstacles})$$

Pour estimer la probabilité d'une mesure individuelle, il est possible d'utiliser une simple loi gaussienne comme modèle probabiliste, mais les modèles sont en général un peu plus évolués et utilisent une combinaison de lois qui modélisent divers phénomènes (Figure 2.5) :

- la mesure effective de l'obstacle visé, modélisé par une gaussienne en général.
- la perception d'un obstacle imprévu, modélisé par une loi décroissante
- la non perception de l'obstacle qui donne une mesure à la distance maximale du télémètre, modélisé par un pic.

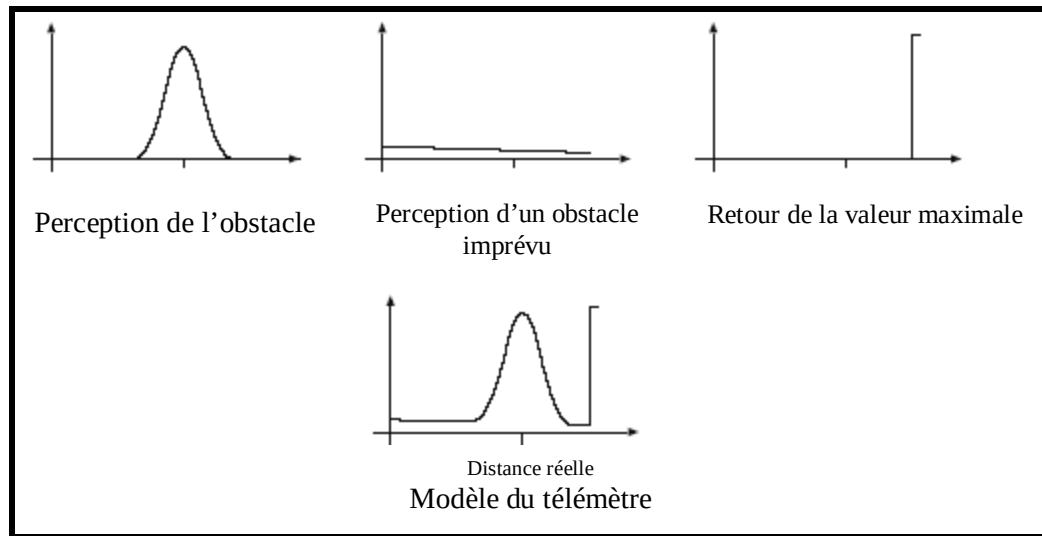


Figure 2.5. *Modèle probabiliste de télémètre.*

Les paramètres de cette combinaison de lois peuvent être réglés manuellement ou estimés à partir d'un ensemble de mesures, par exemple en utilisant un algorithme de maximisation de l'espérance. Ces modèles peuvent être adaptés à tous les types de télémètres.

2.2.1.3. Les caméras

L'utilisation d'une caméra pour percevoir l'environnement est une méthode attractive car elle semble proche des méthodes utilisées par les humains. Le traitement des données volumineuses et complexes fournies par ces capteurs reste cependant difficile à l'heure actuelle, même si cela reste une voie de recherche très explorée.

a- Caméras simples

Une caméra standard peut être utilisée de différentes manières pour la navigation d'un robot mobile. Elle peut être utilisée pour détecter des amers visuels (des points particuliers qui servent de repère, tels que des portes ou des affiches) à partir desquels il sera possible de calculer la position du robot. Si ces amers sont simplement ponctuels, ou de petite taille, il sera en général simplement possible

d'estimer leur direction. Dans le cas où les amers sont des objets connus en 2 ou 3 dimensions, il sera en général possible d'estimer complètement la position du robot par rapport à la leur. Elle peut également être utilisée pour détecter des "guides" de navigation pour le robot, tels que des routes ou des couloirs.

Il est également possible d'utiliser globalement une image pour caractériser une position ou un point de vue dans l'environnement. Il faudra alors comparer cette image aux nouvelles images acquises par le robot pour savoir si le robot est revenu à cette position. Cette comparaison peut faire appel à de très nombreuses techniques, notamment à celles utilisées dans le domaine de l'indexation d'image.

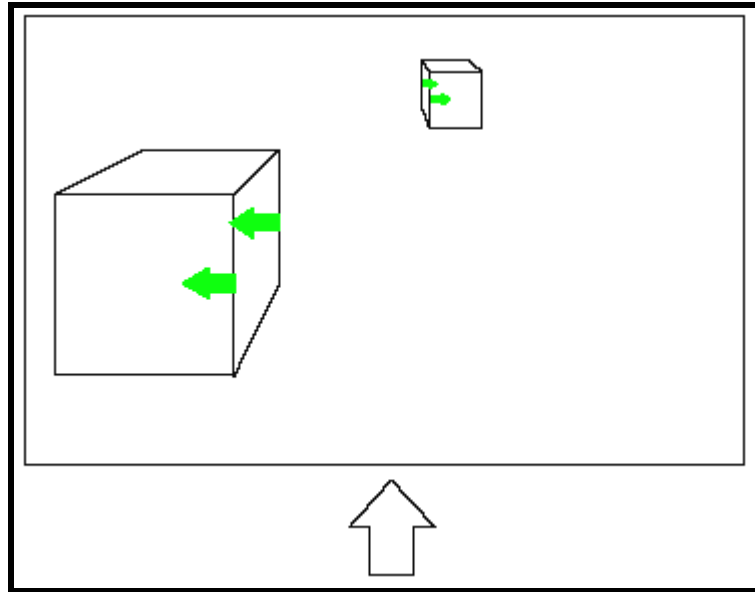


Figure 2.6. Illustration du principe de base du flot optique.

Lorsque le robot est en mouvement, il est également possible de tirer parti du *flot optique* (le mouvement apparent des objets dans l'image, voir (Figure 2.6)), afin d'avoir une estimation de la distance des objets. En effet, les objets les plus proches ont un déplacement apparent plus important que les objets lointains. Cette méthode permet notamment de réaliser un évitement d'obstacles ou de réaliser une

reconstruction tridimensionnelle de l'environnement (par des techniques connues sous le nom de *structure from. motion*).

b- Caméras stéréoscopiques

Lorsque l'on dispose de deux caméras observant la même partie de l'environnement à partir de deux points de vue différents, il est possible d'estimer la distance des objets et d'avoir ainsi une image de profondeur, qui peut être utilisée pour l'évitement d'obstacles ou la cartographie [15]. Cette méthode suppose toutefois un minimum d'éléments saillants dans l'environnement (ou un minimum de texture) et peut être limitée, par exemple dans un environnement dont les murs sont peints de couleurs uniformes. La qualité de la reconstruction risque également de dépendre fortement des conditions de luminosité.

c- Caméras panoramiques

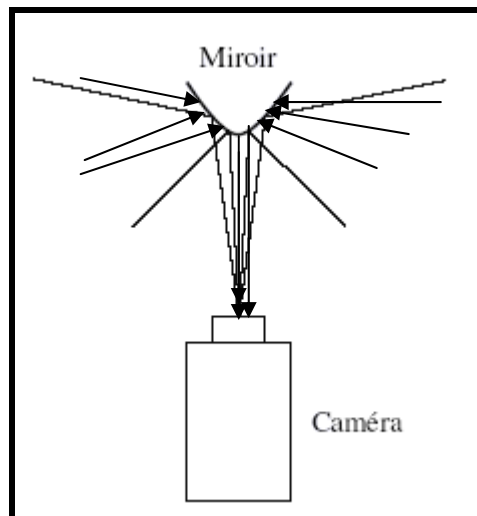


Figure 2.7. Principe des caméras panoramiques avec l'emploi d'un miroir en forme parabolique

Les caméras panoramiques (catadioptriques) sont constituées d'une caméra standard pointant vers un miroir de révolution (par exemple un simple cône, ou un

profil plus complexe qui peut s'adapter à la résolution exacte que l'on veut obtenir sur le panorama) (Figure 2.7). L'image recueillie permet d'avoir une vision de l'environnement sur 360 degrés autour de la camera. Le secteur angulaire vertical observé dépend de la forme du miroir et peut être adapté aux besoins de chaque application.

Ce type de caméra est très pratique pour la navigation car une image prise par une camera panoramique orientée verticalement permet de caractériser une position, indépendamment de la direction du robot. En effet, pour une position donnée et pour deux orientations différentes, la même image sera formée par la caméra, à une rotation autour du centre près, tandis que pour une caméra standard, orientée horizontalement, la scène serait différente.

Ces caméras sont donc très pratiques lorsque l'on caractérise une position de manière globale, mais peuvent aussi être utilisées pour détecter des amers ou pour estimer le flux optique. Dans ce cas, toutefois, comme la géométrie de l'image formée est relativement complexe et comme la résolution obtenue varie énormément selon la direction observée, les algorithmes doivent être adaptés, ce qui pose

Un certain nombre de problèmes. Concernant le flux optique, cependant, les caméras panoramiques possèdent l'avantage de contenir toujours le point d'expansion et le point de contraction dans l'image, ce qui rend l'estimation du mouvement beaucoup plus aisée (Figure 2.8).

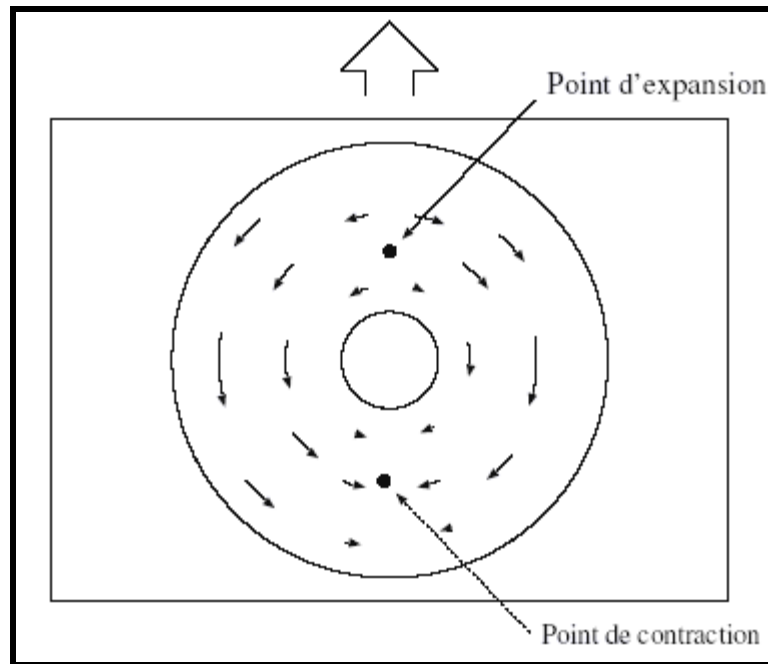


Figure 2.8. *Illustration du flux optique sur une image panoramique dans le cas d'une caméra en translation.*

2.2.1.4. Autres capteurs

a- Les capteurs tactiles

Les robots peuvent être équipés de capteurs tactiles, qui sont le plus souvent utilisés pour des arrêts d'urgence lorsqu'il rencontre un obstacle qui n'avait pas été détecté par le reste du système de perception [16].

.

Ces capteurs peuvent être de simples contacteurs répartis sur le pourtour du robot. Ils ne détectent alors le contact qu'au dernier moment. Il est également possible d'utiliser des petites tiges arquées autour du robot pour servir

d'intermédiaire à ces contacteurs, ce qui permet une détection un peu plus précoce et donne ainsi plus de marge pour arrêter le robot.

b- Les boussoles

Les boussoles permettent, par la mesure du champ magnétique terrestre, de déduire la direction du nord. Ces capteurs peuvent utiliser différentes technologies et ont l'avantage de fournir une direction de référence stable au cours du temps (au contraire des gyroscopes qui dérivent).

Ces capteurs sont toutefois très délicats à utiliser en intérieur car ils sont très sensibles aux masses métalliques présentes dans la structure des bâtiments [17]. En pratique, on les utilise donc principalement en extérieur en apportant le plus grand soin à leur positionnement sur le robot pour éviter les influences des composants du robot, notamment les moteurs électriques.

c- Les balises

Dans certaines applications, il est également possible d'utiliser des balises dont on connaît la position, et qui pourront être facilement détectées par le robot, afin de faciliter sa localisation.

Des techniques très diverses peuvent être utilisées pour ces balises. On peut par exemple utiliser un signal radio, émis de manière omnidirectionnel par la balise.

Le robot sera alors équipé d'une antenne directionnelle qui lui permettra de détecter la direction des différentes balises, afin de déduire sa position par triangulation. On peut également utiliser des codes couleurs ou des codes barres qui pourront être détectés par une caméra.

d- Le GPS

Les besoins de localisation étant omniprésents dans de très nombreux secteurs de la vie actuelle, l'idée d'avoir un système de localisation le plus universel possible a donné lieu à l'apparition du Global Positioning System. C'est un système de balises dont on a placé les balises sur des satellites en orbite terrestre et qui est par conséquent accessible de quasiment partout à la surface du globe. Ce système permet donc d'avoir une mesure de sa position dans un repère global couvrant la terre avec une précision variant de quelques dizaines de mètres à quelques centimètres suivant les équipements.

Ce système est cependant loin de résoudre tous les problèmes de localisation des robots mobiles. Il fonctionne en effet difficilement dans des environnements urbains, et n'est pas utilisable à l'intérieur des bâtiments. Sa précision est de plus souvent trop faible pour qu'un robot terrestre puisse utiliser ces informations seules. En pratique, il est souvent couplé à un système inertiel qui permet de palier aux pertes du signal GPS et il ne remplace de toute façon pas les capteurs du robot qui lui permettent de percevoir son environnement immédiat, qui constitue la source d'information principale pour la navigation à court terme (par exemple l'évitement d'obstacles, par opposition à la navigation à long terme qui consiste à rejoindre un but distant).

2.3. Evitement d'obstacles

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans un environnement dynamique et pour gérer les écarts entre le modèle interne et le monde réel.

Les méthodes utilisées sont efficaces à condition d'avoir une perception correcte de l'environnement. Elles seront par exemple très efficaces avec un télémètre laser, mais donneront des résultats plus bruités avec des sonars. Pour limiter ce problème, il est possible d'appliquer ces méthodes sur une représentation locale (c'est-à-dire de l'environnement proche du robot et centrée sur le robot) de l'environnement qui sera construite en fonction des données de quelques instants précédents. Cette représentation intermédiaire permettra de filtrer une grande partie du bruit des données individuelles (en particulier pour les sonars).

2.4. Conclusion

On a essayé dans ce chapitre de ne pas faire une étude détaillée sur les capteurs, qui relève à la fois de la physique, de l'électronique et du traitement du signal, Nous nous sommes contentés d'expliquer les principes de fonctionnement des capteurs présentés. Et On tâchera simplement de garder à l'esprit que les défauts inhérents aux différents systèmes de mesure utilisés (bruit, erreurs ou échecs de mesures, difficulté de modélisation) influent fortement sur la perception que le robot a de l'environnement et donc peuvent entraîner des perturbations et mauvaises estimations de commande sur les actionneurs du robot mobile.

Description du Réseau de Neurones Développé

Description du Réseau de Neurones Développé

3-1. Introduction

Vu son importance, ce chapitre est scindé en deux parties. La première concerne la présentation détaillée de la structure du réseau de neurones conçu pour notre application. Au sein de cette partie, nous avons présenté la simulation du principe de fonctionnement du système de perception où un télémètre laser a été retenu. En ce qui concerne la deuxième partie, un ensemble de tests montrant l'efficacité du système développé a été montré.

Le travail que nous présentons dans ce chapitre est le fruit de plusieurs étapes de réflexion et de mise en place de solution logicielles. La première étape consiste à sélectionner les réseaux de neurones adéquats (MLP). La seconde s'agit de trouver les bons paramètres d'implémentation pour chacun des réseaux. Ensuite, nous avons élaboré un espace de travail, nous permettant de tester facilement les différents paramètres du problème concernant les ensembles d'apprentissage et le capteur du système de perception utilisé. Ce qui nous a permis de déterminer les paramètres qui donnent des performances optimales au système développé. Une comparaison en termes de performances et d'apprentissage des différents types de structures MLP sera présentée à la fin de ce chapitre.

Comme il peut déduire, la deuxième partie est aussi bien importante que la première. Il s'agit bien de l'algorithme utilisé pour l'apprentissage du réseau de neurones qui sera détaillé dans l'une des sections qui suivent. Le cycle classique de développement des réseaux de neurones peut être séparé en sept étapes structurées suivant l'ordre donné ci-dessous :

● Acquisition Des Données

- ✓ collecte des données
- ✓ analyse des données

- ✓ la mise en forme des données
- ✓ séparation des bases des données
- 1-2 Conception Du Réseau De Neurones
 - ✓ choix d'un réseau de neurones
 - ✓ l'apprentissage du réseau de neurones
 - ✓ validations et test

L'architecture du système conçu peut être schématisée d'une façon simplifiée suivant le synoptique donné par la figure 3-1.

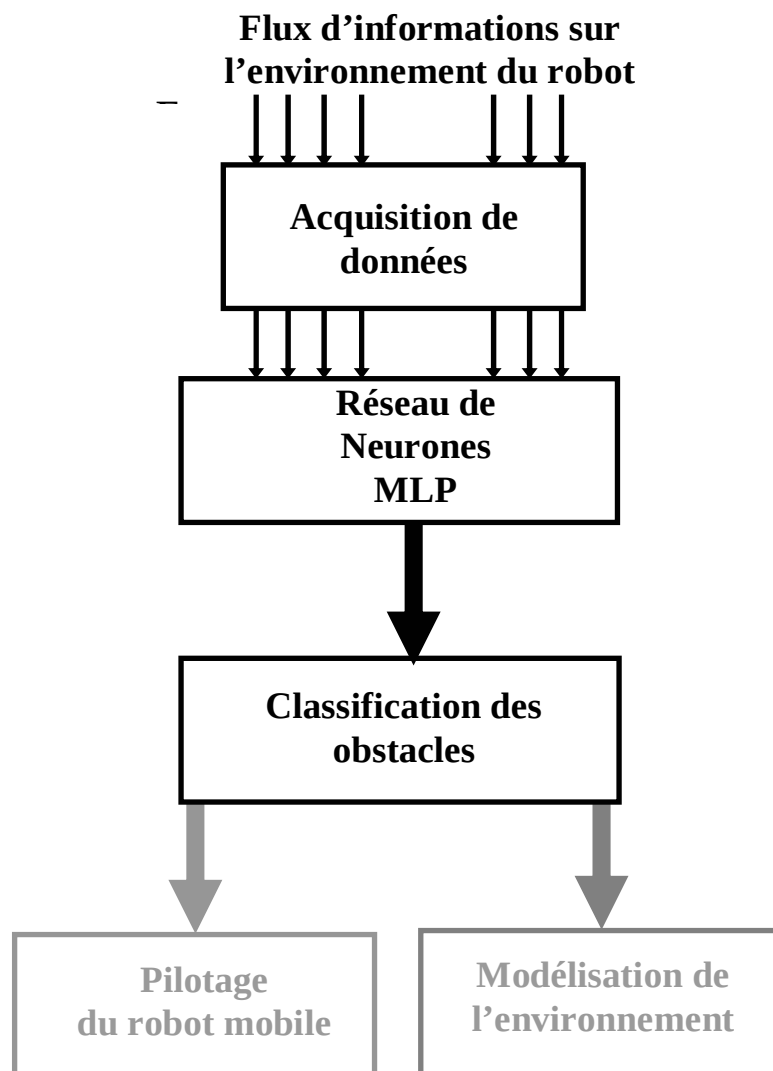


Figure 3.1 Schéma synoptique du système conçu

3-2 Acquisition des données

3-2-1 Collecte des données

L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones et pour le tester. Dans le cas d'applications sur des données réelles, l'objectif est de rassembler un nombre de données suffisant pour constituer une base représentative des données susceptibles d'intervenir en phase d'utilisation du système neuronal. La fonction réalisée résultant d'un calcul statistique, le modèle qu'il constitue n'a de validité que dans le domaine où on l'a ajusté. En d'autres termes, la présentation de données très différentes de celles qui ont été utilisées lors de l'apprentissage peut entraîner une sortie totalement imprévisible. Dans notre application la collecte des données se fait à travers un simple système de perception, qui assure cette collecte, conçu autour d'un capteur laser.

En se basant sur le principe de fonctionnement d'un télémètre laser, un modèle a été simulé. Il s'agit d'une tourelle placée à l'avant du robot mobile. Une rotation de 180° à un pas préfixé permet de collecter un ensemble bien déterminé de mesures. Ces mesures donnent une idée satisfaisante sur une zone de manœuvrabilité dont la surface balayée dépend du robot mobile et de sa nature de mouvement. Pour plus de clarté un ensemble d'exemples a été choisi afin de montrer les valeurs issues du télémètre laser et leur correspondant en mode réduit pour différentes zones utiles de détection.

Exemple -1

Zone de détection : 170 cm

Balayage 180° à un pas de 1°

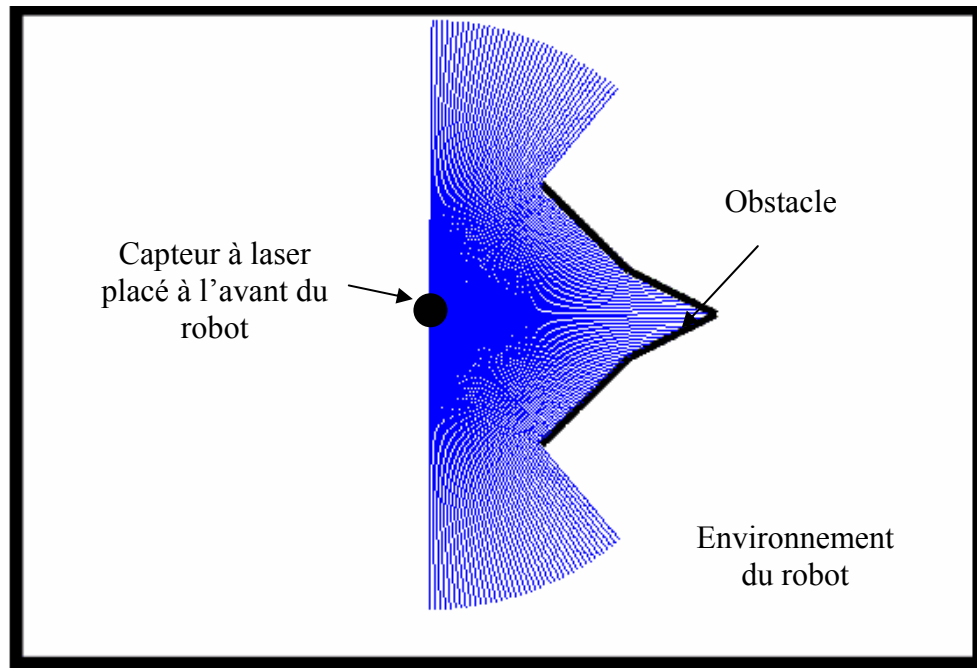


Figure 3-2. Balayage du télémètre laser cas de l'exemple 1

Mesure De Distance Avec Un Pas De Balayage De 1°											
MB	MN	MB	MN	MB	MN	MB	MN	MB	MN	MB	MN
170	1.0000	170	1.0000	100	0.5882	155	0.9118	100	0.5882	170	1.0000
170	1.0000	170	1.0000	101	0.5941	159	0.9353	100	0.5882	170	1.0000
170	1.0000	170	1.0000	101	0.5941	153	0.9000	99.0	0.5824	170	1.0000
170	1.0000	170	1.0000	101	0.5941	149	0.8765	99.0	0.5824	170	1.0000
170	1.0000	170	1.0000	102	0.6000	143	0.8412	99.0	0.5824	170	1.0000
170	1.0000	170	1.0000	103	0.6059	139	0.8176	98.0	0.5765	170	1.0000
170	1.0000	170	1.0000	103	0.6059	135	0.7941	97.0	0.5706	170	1.0000
170	1.0000	170	1.0000	104	0.6118	131	0.7706	97.0	0.5706	170	1.0000
170	1.0000	170	1.0000	105	0.6176	128	0.7529	97.0	0.5706	170	1.0000
170	1.0000	170	1.0000	105	0.6176	126	0.7412	97.0	0.5706	170	1.0000
170	1.0000	170	1.0000	106	0.6235	123	0.7235	97.0	0.5706	170	1.0000
170	1.0000	97.0	0.5706	107	0.6294	118	0.6941	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	108	0.6353	117	0.6882	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	109	0.6412	114	0.6706	97.0	0.5706	170	1.0000
170	1.0000	97.0	0.5706	110	0.6471	114	0.6706	97.0	0.5706	170	1.0000
170	1.0000	97.0	0.5706	111	0.6529	112	0.6588	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	112	0.6588	111	0.6529	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	114	0.6706	110	0.6471	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	114	0.6706	108	0.6353	96.0	0.5647	170	1.0000
170	1.0000	97.0	0.5706	117	0.6882	108	0.6353	97.0	0.5706	170	1.0000
170	1.0000	97.0	0.5706	118	0.6941	106	0.6235	100	0.5882	170	1.0000
170	1.0000	97.0	0.5706	121	0.7118	106	0.6235	170	1.0000	170	1.0000
170	1.0000	98.0	0.5765	124	0.7294	105	0.6176	170	1.0000	170	1.0000
170	1.0000	98.0	0.5765	127	0.7471	104	0.6118	170	1.0000	170	1.0000
170	1.0000	98.0	0.5765	129	0.7588	104	0.6118	170	1.0000	170	1.0000
170	1.0000	98.0	0.5765	134	0.7882	103	0.6059	170	1.0000	170	1.0000
170	1.0000	98.0	0.5765	138	0.8118	102	0.6000	170	1.0000	170	1.0000
170	1.0000	99.0	0.5824	142	0.8353	102	0.6000	170	1.0000	170	1.0000
170	1.0000	99.0	0.5824	145	0.8529	101	0.5941	170	1.0000	170	1.0000
170	1.0000	99.0	0.5824	151	0.8882	100	0.5882	170	1.0000	170	1.0000

Tableau 3.1 mesures et valeurs réduites correspondantes pour exemple 1

MB : valeur brute **MN** : valeur normalisée

Exemple -2

Zone de détection : 190 cm

Balayage 180° à un pas de 2°

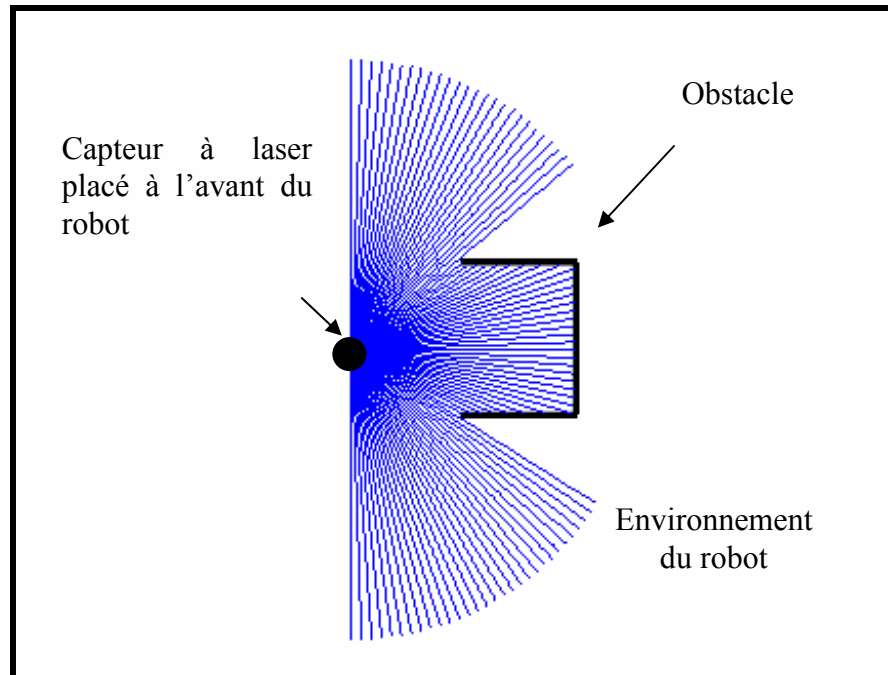


Figure 3.3 balayage du télémètre laser cas de l'exemple 2

Mesure De Distance Avec Un Pas De Balayage De 2°											
MB	MN	MB	MN	MB	MN	MB	MN	MB	MN	MB	MN
190	1.0000	190	1.0000	94.0	0.4947	146	0.7684	92.0	0.4842	190	1.0000
190	1.0000	190	1.0000	101	0.5316	147	0.7737	190	1.0000	190	1.0000
190	1.0000	190	1.0000	110	0.5789	148	0.7789	190	1.0000	190	1.0000
190	1.0000	190	1.0000	120	0.6316	149	0.7842	190	1.0000	190	1.0000
190	1.0000	190	1.0000	133	0.7000	150	0.7895	190	1.0000	190	1.0000
190	1.0000	190	1.0000	149	0.7842	151	0.7947	190	1.0000	190	1.0000
190	1.0000	190	1.0000	150	0.7895	153	0.8053	190	1.0000	190	1.0000
190	1.0000	190	1.0000	149	0.7842	155	0.8158	190	1.0000	190	1.0000
190	1.0000	190	1.0000	148	0.7789	147	0.7737	190	1.0000	190	1.0000
190	1.0000	190	1.0000	147	0.7737	136	0.7158	190	1.0000	190	1.0000
190	1.0000	190	1.0000	146	0.7684	126	0.6632	190	1.0000	190	1.0000
190	1.0000	190	1.0000	146	0.7684	118	0.6211	190	1.0000		
190	1.0000	190	1.0000	146	0.7684	111	0.5842	190	1.0000		
190	1.0000	190	1.0000	146	0.7684	104	0.5474	190	1.0000		
190	1.0000	84.0	0.4421	146	0.7684	99.0	0.5211	190	1.0000		
190	1.0000	88.0	0.4632	146	0.7684	94.0	0.4947	190	1.0000		

Tableau 3.2 mesures et valeurs réduites correspondantes pour exemple 2

On essaie de montrer par les quatre figures allant de 3-4 à 3-7 différentes situations de cas d'obstacles non détectés et de cas de détection d'obstacles avec confusion dans la reconnaissance de formes.

Exemple-3

Cas d'obstacle en dehors de la zone utile de détection choisie suivant les dimensions du robot mobile. Il est à noter que la zone utile de détection peut en générale être différente de la portée du télémètre.

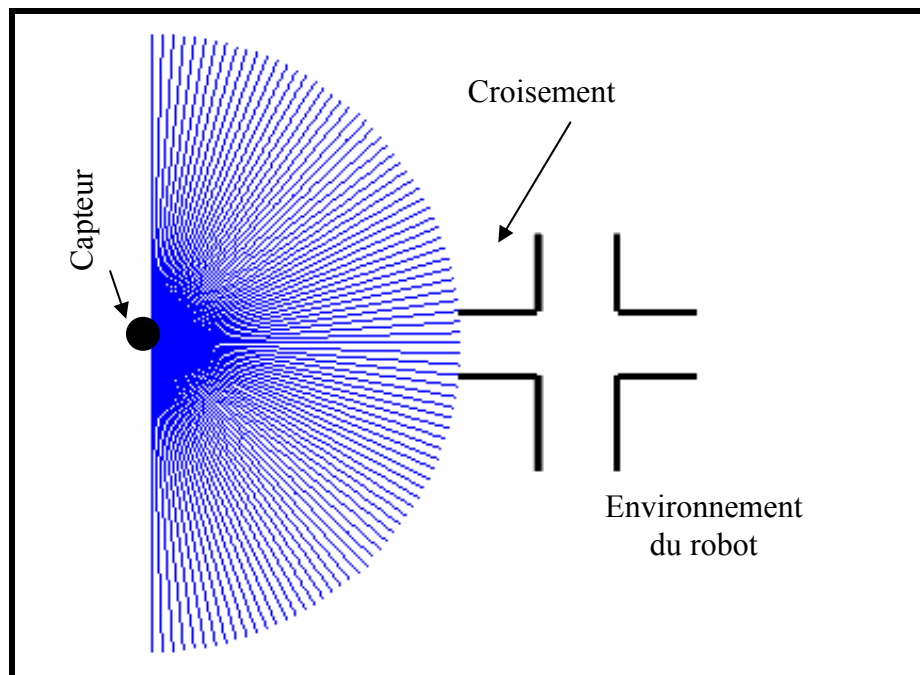


Figure 3.4 balayage du télémètre laser cas de l'exemple 3

Exemple-4

Cas d'obstacle est partiellement détecté dans la zone utile de détection. Dans ce cas précis, il se peut qu'il y ait confusion entre formes d'obstacles d'une même classe.

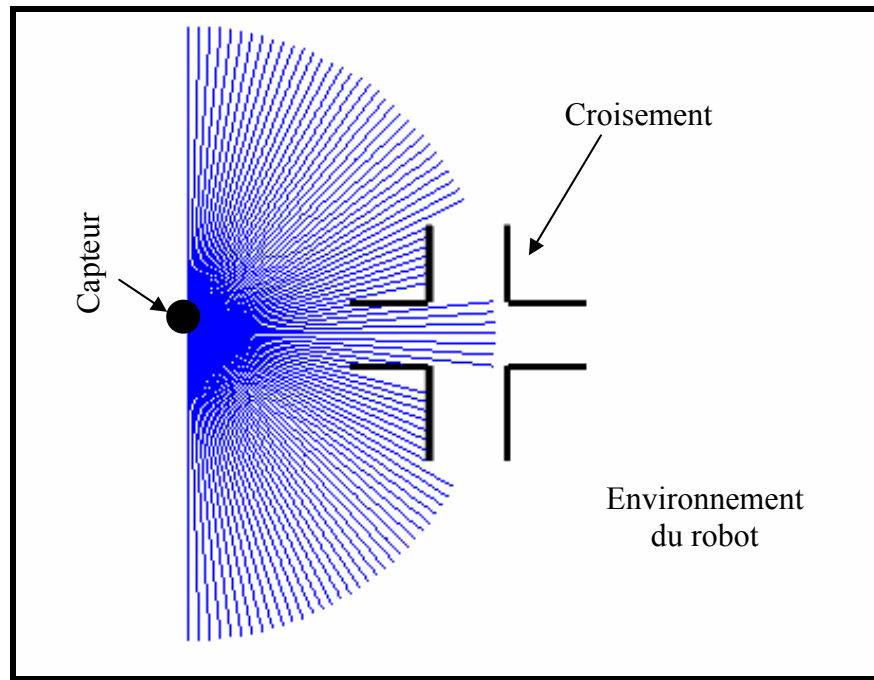


Figure 3.5 balayage du télémètre laser cas de l'exemple 4

Exemple-5

Cas d'obstacle placé entièrement dans la zone utile de détection. Ce cas a montré la grande probabilité avec laquelle la forme de l'obstacle au sein de sa classe puisse être déterminée.

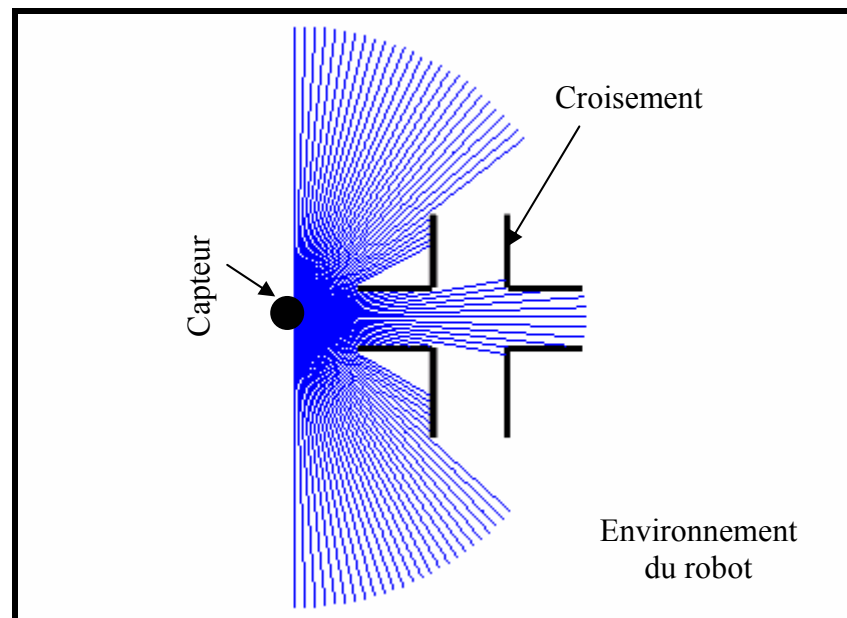


Figure 3.6 balayage du télémètre laser cas de l'exemple 5

Exemple-5

Cas d'obstacle placé tout près du robot mobile, c'est le cas où la forme est entièrement ratée. Cependant, en pratique c'est un cas qui ne peut être résolu qu'en faisant appel aux capteurs tactiles.

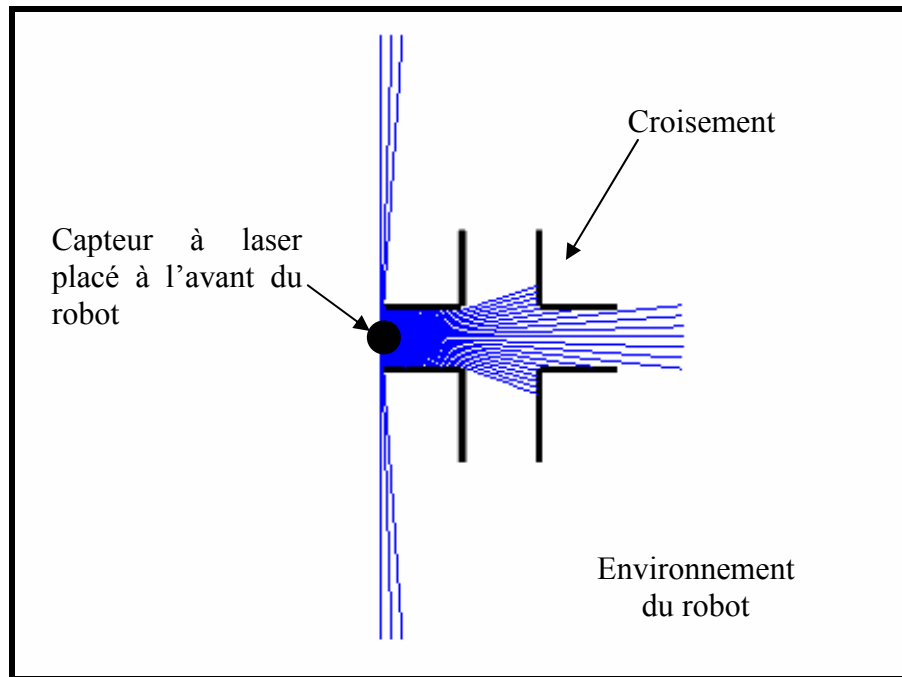


Figure 3.7 balayage du télémètre laser cas de l'exemple 6

Une fois la collecte des données est faite, un prétraitement des données ainsi obtenues doit être effectué afin de présenter à l'entrée du réseau de neurones facilement maniables.

3-2-2 analyse des données

Il est souvent préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier ces données. Ces Caractéristiques constituent l'entrée du réseau de neurones. Notons que cette étude n'est pas spécifique aux réseaux de neurones, quelque soit la méthode de détection ou de classification utilisée.

Il est généralement nécessaire de présenter des caractéristiques représentatives. Cette détermination des caractéristiques a des conséquences à la fois sur la taille du réseau donc le temps de simulation, sur les performances du système pour une capacité de séparation avec un taux de détection et sur le temps d'apprentissage. Une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes.

Dans le cas d'un problème de classification, il appartient à l'expérimentateur de déterminer le nombre de classes auxquelles ses données appartiennent et surtout de déterminer pour chaque donnée sa classe d'appartenance. Dans le cas notre application, les obstacles dans un environnement d'intérieur prennent différentes formes pouvant être réparties sur différentes classes.

Citons couloir, impasse, coin, pièce, mur, angle-gauche, entrée, angle-droit, croisement, croisement en T et sortie comme exemples de formes d'obstacles pouvant obstruer le chemin d'un robot mobile d'intérieur.

Cependant, la modélisation dont on parlée sur les différentes formes d'obstacles n'est pas figée, d'autres formes peuvent facilement être ajoutées ou supprimées. De plus, on peut distinguer les formes qui appartiennent à la même classe soit en utilisant des fonctions telle que l'histogramme pour voir les distributions des formes, soit en faisant un prétraitement après essai de test d'erreur.

Impasses	Coins	passages	Plans

Tableau 3.3 Les différentes classes constituant la base d'apprentissage

3-2-3 la mise en forme des données

De manière générale, pour la mise en forme des données pour un réseau de neurones, les bases de données doivent subir un prétraitement afin d'être adaptées aux entrées et sorties du réseau de neurones. Un prétraitement courant consiste à effectuer une normalisation appropriée, qui tient compte de l'amplitude des valeurs acceptées par le réseau. Nous reviendrons par la suite sur les avantages et inconvénients de la normalisation. Dans notre cas la normalisation se fait par division de toutes les valeurs de distances du vecteur **dist** par la valeur du rayon de la zone utile de détection d'obstacle a fin de ramener toutes les valeurs dans l'intervalle réduit normalisé [0,1].

3-2-4 Séparation des bases de données

Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données : une base pour effectuer l'apprentissage du réseau et une autre pour le tester afin de déterminer ses performances pour le contrôle de sa phase d'apprentissage. Certes, il n'y a pas de règles pour la détermination de ce partage de manière quantitatif. Cependant, il résulte souvent d'un compromis tenant compte du nombre de données dont on dispose et du temps imparti pour effectuer l'apprentissage.

Chaque base doit cependant satisfaire aux contraintes de représentativité de chaque classe de données et elle doit généralement refléter la distribution réelle ; c'est à dire la probabilité d'occurrence des diverses classes.

Avant de détailler ces deux étapes importantes dans la conception d'un réseau de neurones il est tout de même important de passer la conception du réseau.

3-3 Conception du Réseau de Neurones

3-3-1 choix d'un réseau de neurones

Il existe un grand nombre de types de réseaux de neurones, avec pour chacun ses avantages et ses inconvénients. Le choix d'un réseau est étroitement lié aux critères :

- la tâche à effectuer (classification, association, contrôle de processus, séparation aveugle de sources...),
- la nature des données
- d'éventuelles contraintes d'utilisation temps réel (certains types de réseaux de neurones, tels que la 'machine de Boltzmann' [AZENCOTT et al., 1992], nécessitant des tirages aléatoires et un nombre de cycles de calculs indéfini avant stabilisation du résultat en sortie, présentent plus de contraintes que d'autres réseaux pour une utilisation temps réel),

- des différents types de réseaux de neurones disponibles dans le logiciel de simulation que l'on compte utiliser ou le développement de son propre programme.

Ce choix est aussi fonction de la maîtrise ou de la connaissance que l'on a de certains réseaux, ou encore du temps dont on dispose pour tester une architecture prétendue plus performante. Dans notre cas on a utilisé la toolbox du matlab vu qu'elle possède des fonctions et des algorithmes très puissants dans ce domaine réseaux de neurones et plus précisément la classification, ce que nous allons le détailler par la suite

3-3-1-1 Utilisation de la Boîte à outils Matlab Réseaux de Neurones

Il s'agit d'une structure hiérarchique ; certains membres de la structure de base sont eux mêmes des structures relativement complexes. Cette structure peut être visualisée lors de la création du réseaux ou plus tard.

Création d'un Réseau MLP

- ✓ La fonction de création d'un réseau est spécifique au modèle de réseau utilisé (newc, newlvq, etc).
- ✓ Pour les réseaux multicouches, la création du réseau est faite par la commande de la fonction newff :

reseau=newff(PR, [S1 S2.....SN1] , {TF1 TF2.....TFN1}, BTF , BLF , PF) ;

Avec :

PR : Plage des variations des entrées (affichage par minmax(p)) .

Si : Nombre des neurones dans la couche i, pour N1 couches.

TFi : fonction d'activation dans la couche i, par défaut la fonction

d'activation est 'tansig', elle peut être :

Hardlim : Fonctions Heaviside.

hardlims: Fonctions signe.

logsig : Fonction logarithme sigmoïde.

tansig : Fonction tangente sigmoïde.

pureline : fonction linéaire.

satlins : Fonction linéaire a seuil.

BTF : l'algorithme d'apprentissage par paquets du réseau, la fonction BTF peut être:

trainlm : apprentissage par l'algorithme de Levenberg-Marquardt

trainbfg : apprentissage par l'algorithme BFGS.

trainoss : apprentissage par l'algorithme «one-step BFGS Method»

trainbr : version de trainlm avec modération automatique des poids.

trainrp : apprentissage par l'algorithme RPROP.

trainscg : apprentissage par scaled conjugate gradient (SCG)

traincgf : apprentissage par la méthode du gradient conjugué+FLETCHER-REEVES.

traincp : apprentissage par la méthode du gradient conjugué+POLAK-RIBIERE.

BLF : l'algorithme d'apprentissage incrémental du réseau, la fonction BLF peut être :

Learngd : L'algorithme d'apprentissage sera la descente de gradient à taux d'apprentissage fixe.

Learngdm : version de learngd avec moment.

PF : fonction du coût.

mae : erreur absolue moyenne

mse : erreur quadratique moyenne

msereg : version de mse avec modérations des poids

sse : somme des carrés des erreurs

Apprentissage

Il existe 2 types d'apprentissage :

- Incrémental : fonction adapt. Apprentissage incrémental (on line) ; les poids sont modifiés à chaque présentation d'une entrée.
- Par paquets : fonction train. Apprentissage par paquets (*off-line, batch mode*) ; les poids sont modifiés uniquement après présentation de toutes les entrées.

Simulation (ou activation) d'un réseau

$A = \text{sim}(\text{net}, p)$; où net est le pointeur retourné par une fonction de création de réseau.

On va maintenant parler sur l'apprentissage, une propriété très importante dans les réseaux de neurones, après qu'on a opté pour la fonction NEWFF comme structure générale du réseau. La somme des carrés des erreurs SSE qui est utilisée comme fonction du coût, est bien adaptée à notre application de classification. L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux, cependant il existe plusieurs algorithmes utilisés pour faire l'apprentissage des réseaux multicouches. En général, les méthodes du second ordre assurent une convergence plus rapide que celle du premier ordre pour les réseaux dont l'apprentissage est par paquets. Cependant, pour les réseaux dont l'apprentissage est incrémental les méthodes du premier ordre assurent une convergence plus rapide que celle du second ordre. Citons quelques algorithmes d'apprentissage :

- L'algorithme de Levenberg-Marquardt est le plus rapide et il assure la meilleure convergence vers un minimum de l'erreur quadratique pour les problèmes d'approximation des fonctions où le nombre des poids du réseau est inférieur à cents. Quand le nombre de poids augmente l'efficacité de l'algorithme LM diminue, car la taille du Hessien augmente

et nécessite une très grande place dans la mémoire, cet algorithme est moins adapté aux problèmes de classification.

- L'algorithme RPROP est le plus rapide et assure la meilleure convergence vers un minimum de l'erreur quadratique pour les problèmes de classification. Cependant, cet algorithme est mal adapté aux problèmes d'approximations des fonctions ; il nécessite une place mémoire très modeste.
- Les algorithmes du gradient conjugué, en particulier SCG, performant sur des variétés des problèmes, spécifiquement dans le cas où la taille du réseau est grande. L'algorithme SCG est aussi rapide que l'algorithme LM dans les problèmes d'approximation des fonctions par contre il est plus rapide que le LM dans le cas des réseaux de grandes tailles.). Il est aussi rapide que l'algorithme RPROP pour les problèmes de classification. l'algorithme SCG nécessite relativement une place mémoire très modeste. Ce sont ces paramètres qui nous ont encouragé à utiliser ces algorithmes.

3-3-2 l'apprentissage du réseau de neurones

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation. Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références ; il s'agit des réponses souhaitées pour chaque exemple présenté. La technique de validation croisée, qui sera précisée par la suite, permet un arrêt adéquat de l'apprentissage pour obtenir de bonnes performances de généralisation. Certains algorithmes d'apprentissage se chargent de la détermination des paramètres architecturaux du réseau de neurones. Si on n'utilise pas ces techniques, l'obtention des paramètres architecturaux optimaux se fera par comparaison des performances obtenues

pour différentes architectures de réseaux de neurones. Des contraintes dues à l'éventuelle réalisation matérielle du réseau peuvent être introduites lors de l'apprentissage.

3-3-2-1 bases d'apprentissage

Le choix de la base d'apprentissage est très important dans la conception ainsi que dans les résultats obtenus par le réseau. On avait mentionné cette importance dans la partie analyse des données de ce chapitre parce qu'un mauvais choix de cette base peut entraîner beaucoup de problèmes sur les performances du système comme sa capacité de séparation et son taux de détection, et sur le temps de développement (temps d'apprentissage). Une étude statistique sur les données peut permettre d'écarter celles qui sont aberrantes et redondantes dans le cas d'un problème de classification. Pour notre application, on a divisé la base d'apprentissage en quatre classes comme il est illustré dans le (tableau 3.3). Les différentes classes qui constituent les bases d'apprentissages sont :

Classe passage : Couloir, croisement, crois en T, etc....

Classe impasse : formes en U et V...

Classe coin : U et V miroités...

Classe plan : mur horizontal ou incliné

Ce sont ces formes d'obstacles qui sont simulées ; les 91 valeurs issues du télémètre laser lors de leur détection, après traitement et adaptation, forment le vecteur d'entrées de notre réseau de neurones. Toutes les situations exploitées durant l'étape d'apprentissage constituent la base DATA_OBS. Les figures 3-8 et 3-9 montrent deux exemples de simulation de détection d'un mur et d'impasse respectivement et dont les valeurs de mesures correspondantes seront utilisées pour l'apprentissage du réseau.

La portée utile de détection d'obstacle est limitée aux cas des deux exemples à un rayon de 190 cm avec une précision de balayage de 2°.

- Cas de détection d'un mur :

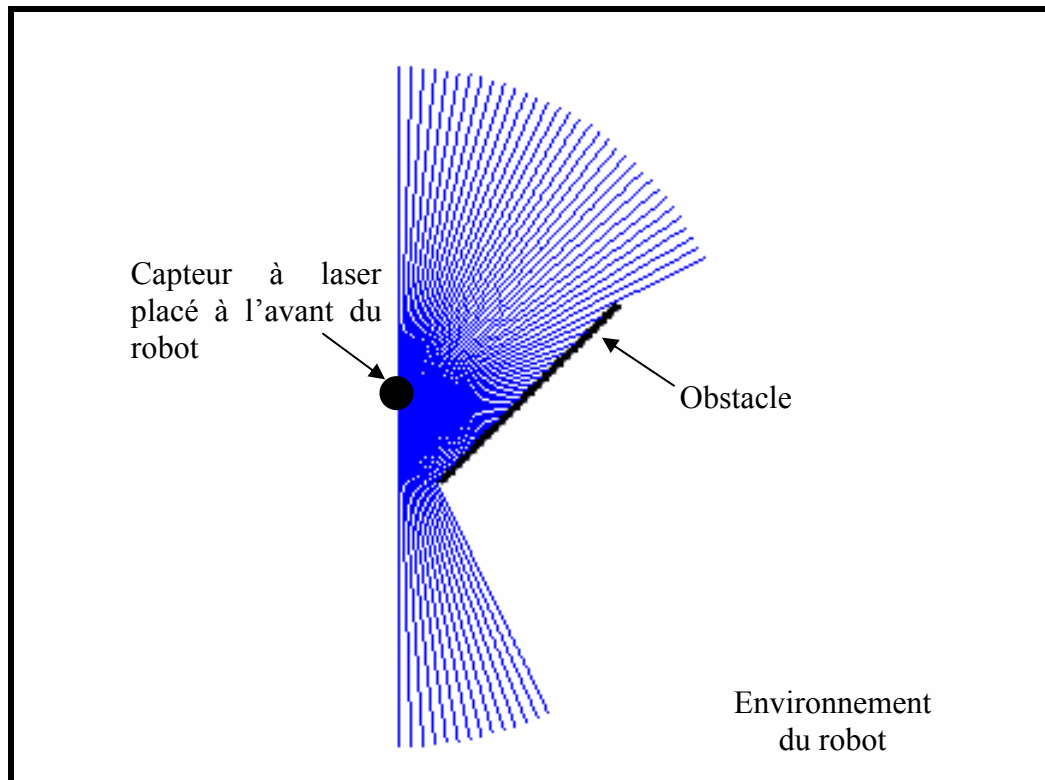


Figure 3.8 cas de détection de mur pour apprentissage

- Cas de détection d'une impasse en forme de coin:

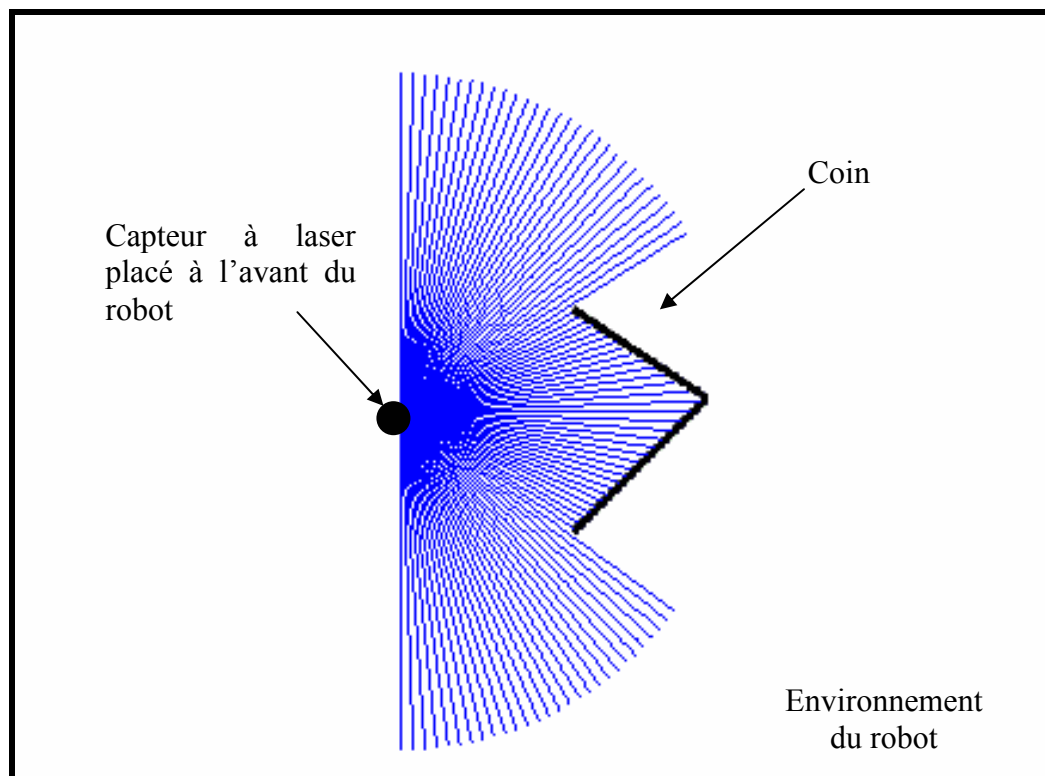


Figure 3.9 cas de détection d'impasse pour apprentissage

Une caractéristique très importante à prendre en considération lors de développement des réseaux de neurones est de voir comment les réponses de ces derniers tendent vers les résultats attendus et surtout avec quelle vitesse ; c'est la propriété convergence.

3-3-2-2 Convergence

Dans la plupart des travaux utilisant les réseaux de neurones type MLP, l'algorithme d'apprentissage le plus utilisé est celui basé sur la règle de rétropropagation du gradient. Son principe de fonctionnement consiste à minimiser la fonction du coût E donnée par l'équation 3-1. C'est ce qui nous a encouragés à tester cet algorithme qui a donné de bons résultats dans le cas de notre application.

$$E = \sum_i (d_k - s_k)^2 \quad (3-1)$$

La gradient intervient dans la modification des poids des cellules suivant l'équation

$$w_{ij}^{t+1} = w_{ij}^t - \eta \frac{\partial E}{\partial w_{ij}} \quad (3-2)$$

Sachant la sortie de chaque cellule au niveau de sa couche est définie comme une fonction d'activation appliquée au potentiel de cette même cellule. Ce potentiel dépend des cellules précédentes avec des coefficients appelés poids comme il est donné par l'équation 3 - 3.

$$s_i = f(p_i) \text{ Avec } p_i = \sum_j w_{ij} x_j \quad (3-3)$$

Par un changement de variables dans l'expression du gradient du coût, on arrive à l'expression donnée par l'équation 3-4.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial p_i} \frac{\partial p_i}{\partial w_{ij}} \quad (3-4)$$

En remplaçant p_i par son expression en (3-3), on obtient l'expression simplifiée donnée par l'équation 3-5.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial p_i} x_j \quad (3-5)$$

L'algorithme utilisé dans notre application est le RPROP. Son principe de fonctionnement consiste à retenir que les valeurs des poids au moment de changement de signe. C'est cette propriété qui a rendu cet algorithme rapide et bien adapté aux applications de la classification qui exigent un espace mémoire important.

Certes il n'y a pas que le choix de l'algorithme qui permet d'améliorer la vitesse d'apprentissage d'un réseau de neurones selon l'application, mais le taux d'apprentissage et le nombre de neurones dans les couches cachées ont aussi un effet sur ce critère. Les courbes données sur les quatre figures 3-10 à 3-13 illustrent comment le nombre de neurones et le taux d'apprentissage agissent sur la fonction coût E.

Le programme d'apprentissage est configuré comme suit :

net = newff(minmax(P),[Nnc Nns],{fcl fs},'trainrp'); avec P : les entrées, Nnc : nombre de neurones de la couche cachée, Nns : nombre de neurones de la couche de sortie, fcl : fonction d'activation pour les neurones de la couche cachée, fs : fonction d'activation pour les neurones de la couche sortie, trainrp : algorithme RPROP d'apprentissage choisi.

net.trainParam.epochs = 5000; Nombre maximal des itérations.

net.trainParam.goal = 0.000001; Seuil de la fonction coût E

net.trainParam.mc = 0.95; mc :moment utilisé

net.trainParam.lr =; η : taux d'apprentissage

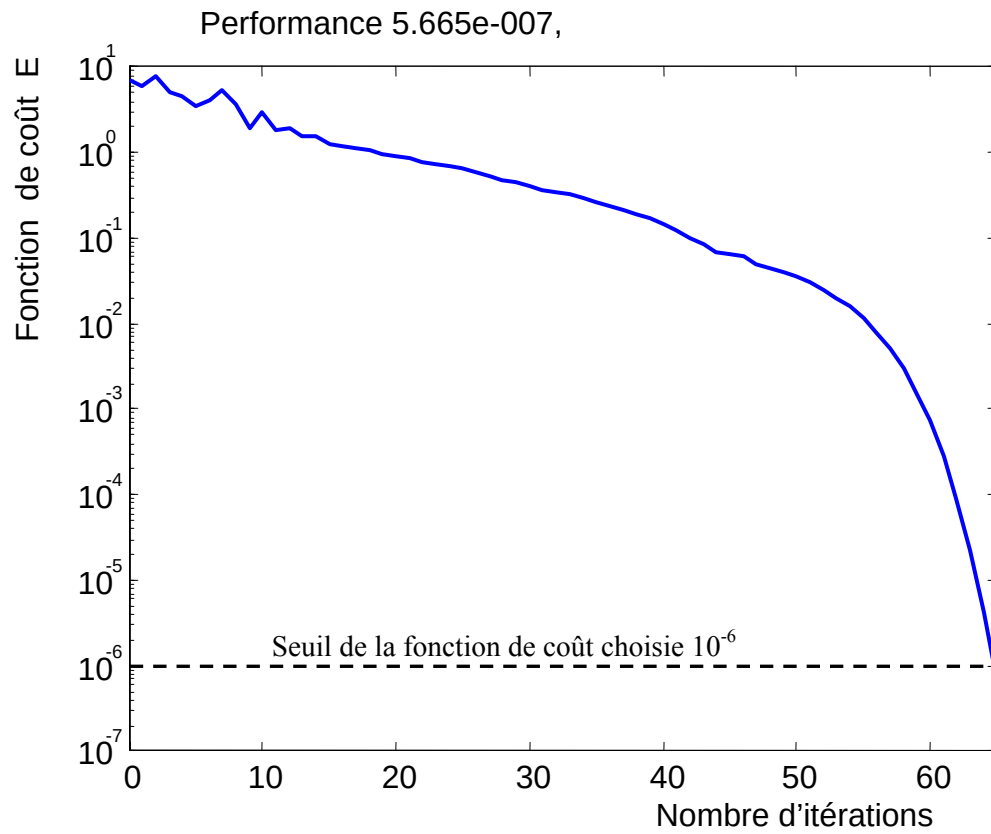


Figure 3.10 Evolution de la fonction coût E,

Nombre de neurones dans la couche cachée= 20

Taux d'apprentissage $\eta=0,35$.

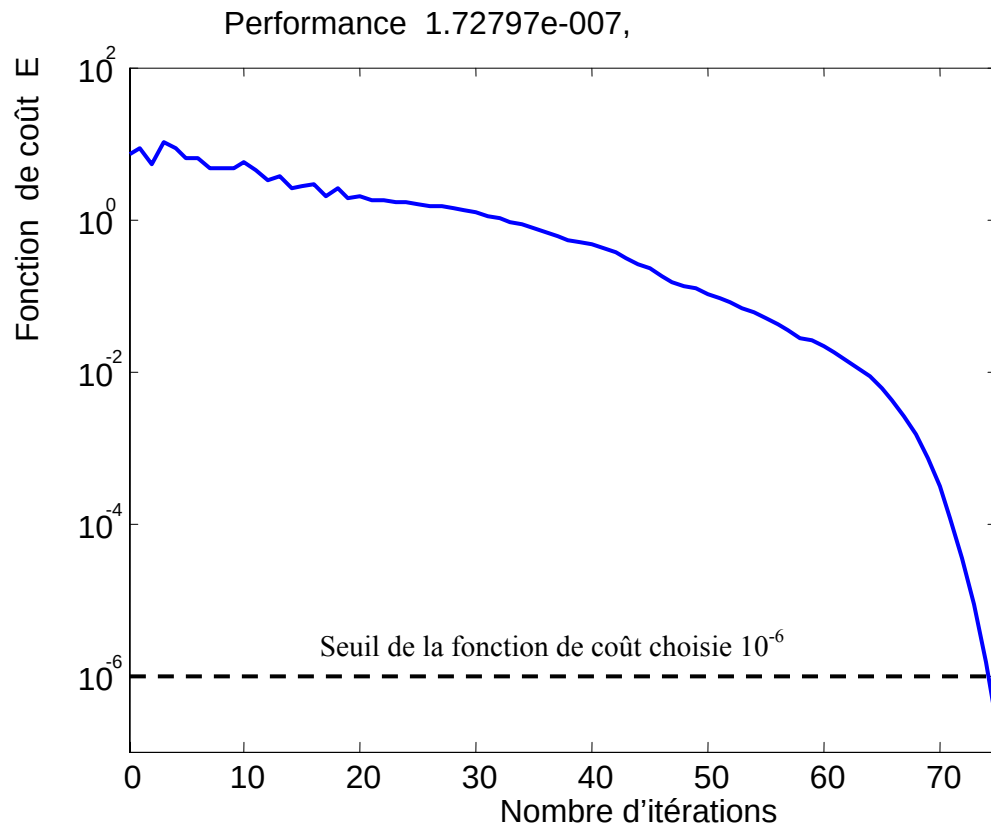


Figure 3.11 Evolution de la fonction coût E,

Nombre de neurones dans la couche cachée= 40

Taux d'apprentissage $\eta=0,35$.

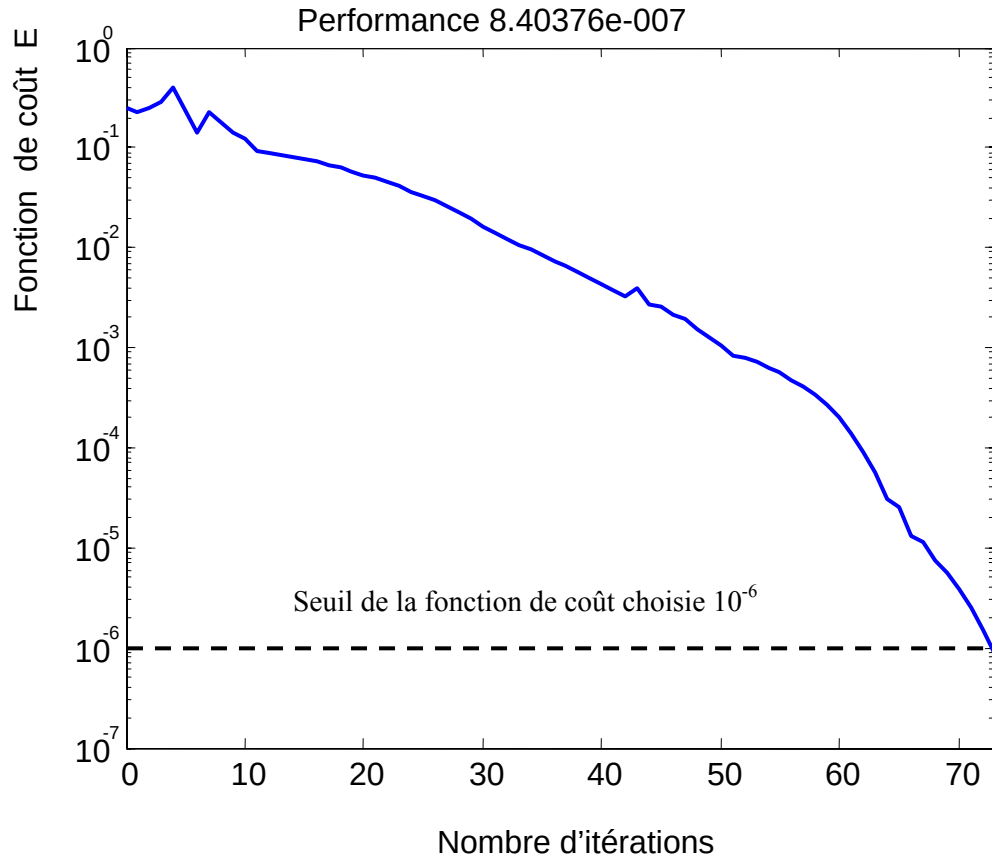


Figure 3.12 Evolution de la fonction coût E,

Nombre de neurones dans la couche cachée= 20

Taux d'apprentissage $\eta=0,95$.

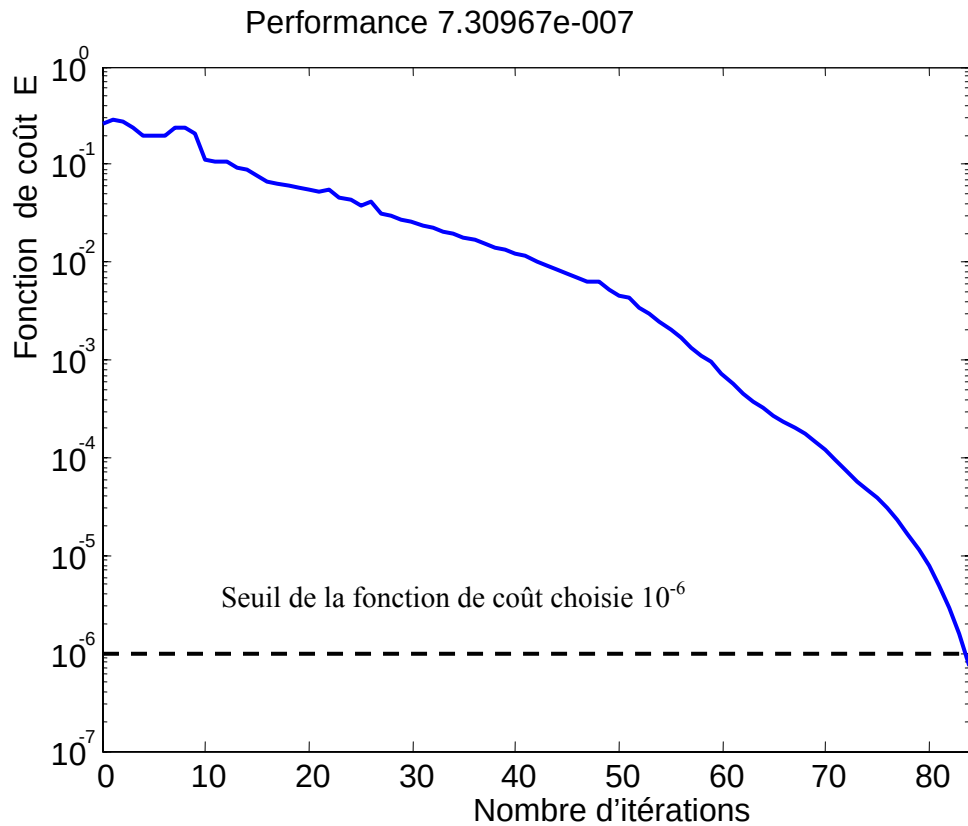


Figure 3.13 Evolution de la fonction coût E,

Nombre de neurones dans la couche cachée= 40

Taux d'apprentissage $\eta=0,95$.

3-3-3 validations et test

3-3-3-1 exemples de base de test

Une fois le réseau de neurones entraîné (après apprentissage), il est nécessaire de le tester sur une base de données différentes de celles utilisées pour l'apprentissage. Ce test permet à la fois d'apprécier les performances du système neuronal et de détecter le type de données qui pose problème. Si les performances ne sont pas satisfaisantes, il faudra soit modifier l'architecture du réseau, soit modifier la base d'apprentissage (caractéristiques discriminantes ou représentativité des données de chaque classe).

Les figures 3-14 et 3-15 montrent deux exemples de test. Le premier exemple est choisi tel qu'il montre une confusion entre plan incliné et couloir. Le second est choisi comme l'association d'un couloir et d'un coin. IL a été remarqué lors de l'opération de test que la confusion au niveau du premier exemple dépend du degré d'inclinaison du mur supérieur du couloir. Le deuxième est dans les cas détecté comme impasse en forme de coin.

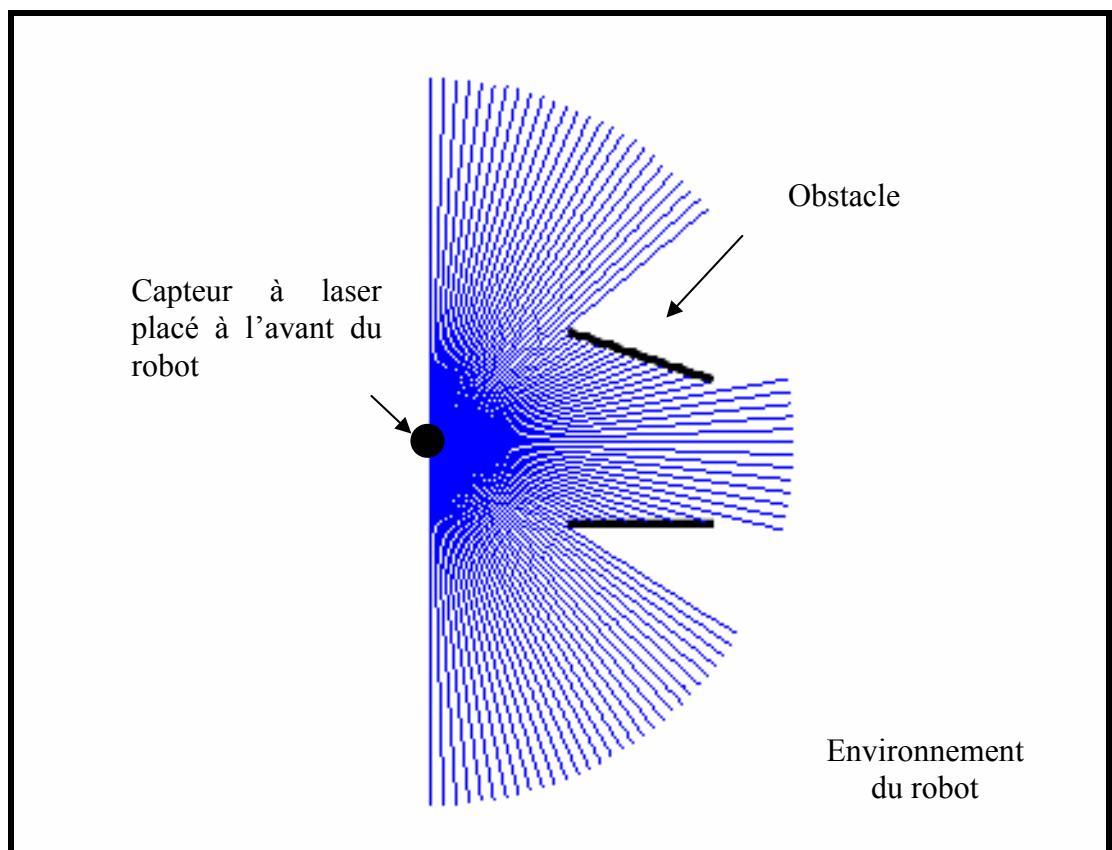


Figure3.14 représente Un obstacle sous forme de Couloir déformé pour le test

ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures
1	1.0000	20	1.0000	39	1.0000	58	0.5842	77	1.0000
2	1.0000	21	1.0000	40	1.0000	59	0.5632	78	1.0000
3	1.0000	22	1.0000	41	1.0000	60	0.5474	79	1.0000
4	1.0000	23	1.0000	42	1.0000	61	0.5316	80	1.0000
5	1.0000	24	1.0000	43	1.0000	62	0.5105	81	1.0000
6	1.0000	25	1.0000	44	1.0000	63	0.5000	82	1.0000
7	1.0000	26	1.0000	45	1.0000	64	0.4842	83	1.0000
8	1.0000	27	1.0000	46	1.0000	65	0.4842	84	1.0000
9	1.0000	28	1.0000	47	1.0000	66	1.0000	85	1.0000
10	1.0000	29	1.0000	48	1.0000	67	1.0000	86	1.0000
11	1.0000	30	1.0000	49	1.0000	68	1.0000	87	1.0000
12	1.0000	31	0.4421	50	1.0000	69	1.0000	88	1.0000
13	1.0000	32	0.4632	51	1.0000	70	1.0000	89	1.0000
14	1.0000	33	0.4947	52	0.7789	71	1.0000	90	1.0000
15	1.0000	34	0.5316	53	0.7368	72	1.0000	91	1.0000
16	1.0000	35	0.5789	54	0.6947	73	1.0000		
17	1.0000	36	0.6316	55	0.6684	74	1.0000		
18	1.0000	37	0.7000	56	0.6316	75	1.0000		
19	1.0000	38	0.7842	57	0.6053	76	1.0000		

Tableau 3-4 Valeur réduites des données issues du télémètre
Cas de la configuration de l'exemple figure 3-14

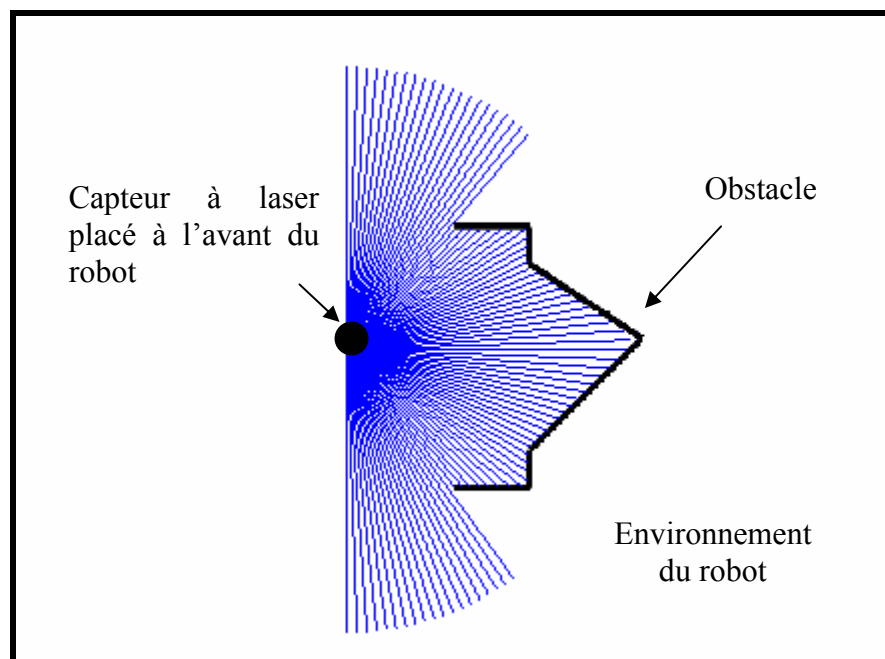


Figure3.15 obstacle de test en forme couloir+ coin

ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures	ordre	Valeur réduite de la mesures
1	1.0000	20	0.6158	39	0.8158	58	0.7000	77	1.0000
2	1.0000	21	0.6263	40	0.8316	59	0.7053	78	1.0000
3	1.0000	22	0.6474	41	0.8526	60	0.7158	79	1.0000
4	1.0000	23	0.6684	42	0.8737	61	0.7316	80	1.0000
5	1.0000	24	0.6895	43	0.8947	62	0.7474	81	1.0000
6	1.0000	25	0.7158	44	0.9263	63	0.7579	82	1.0000
7	1.0000	26	0.7474	45	0.9526	64	0.7211	83	1.0000
8	1.0000	27	0.7789	46	0.9842	65	0.6895	84	1.0000
9	1.0000	28	0.7842	47	1.0000	66	0.6579	85	1.0000
10	1.0000	29	0.7632	48	0.9789	67	0.6316	86	1.0000
11	1.0000	30	0.7474	49	0.9316	68	0.6105	87	1.0000
12	1.0000	31	0.7316	50	0.8947	69	0.5895	88	1.0000
13	1.0000	32	0.7316	51	0.8632	70	0.5684	89	1.0000
14	1.0000	33	0.7421	52	0.8368	71	1.0000	90	1.0000
15	1.0000	34	0.7474	53	0.8105	72	1.0000	91	1.0000
16	1.0000	35	0.7632	54	0.7842	73	1.0000		
17	1.0000	36	0.7684	55	0.7632	74	1.0000		
18	1.0000	37	0.7842	56	0.7421	75	1.0000		
19	1.0000	38	0.7947	57	0.7211	76	1.0000		

Tableau 3-5 Valeur réduites des données issues du télémètre

Cas de la configuration de l'exemple figure 3-15

3-3-3-2 Etapes de validation

Pour la vérification de la robustesse et la performance du réseau de neurones développé, dans le cas de notre application de classification nous avons jugé utile de le faire passer par les étapes de test suivantes :

a- Taux de reconnaissance

Dans cette partie, il s'agit de déterminer le paramètre qui permet de déterminer le pourcentage de classification sans erreur. Ceci est défini comme étant le taux d'erreur dont l'expression :

$$\text{taux d'erreur} = \frac{\text{nombre de cas mal classés}}{\text{nombre total de cas}}$$

Une valeur du rapport égale à l'unité indique un taux de reconnaissance de 100%. Dans le cas de nos essais nous avons essayé de tester le degré

d'identification d'éléments au sein d'une même classe, ce qui a conduit aux résultats du tableau 3-6. On s'est limité à un nombre de 5 formes pour l'apprentissage et 5 pour 5 autres pour la classification.

	Mur	Coin	Impasse
Apprentissage	5	5	5
Généralisation	5	4	5
	Couloir	T	Croisement
Apprentissage	5	5	5
Généralisation	5	4	3
	Pièce	angle _D	angle _G
Apprentissage	5	5	5
Généralisation	3	3	3
	Entrée	Sortie	
Apprentissage	5	5	
Généralisation	4	4	

Tableau 3.6 Taux de reconnaissance des situations

Pour d'autres formes géométriques et toujours dans le même contexte du taux de reconnaissance, nous avons pris 3 éléments pour la base d'apprentissage et 5 autres pour la classification ; les résultats obtenus sont résumés sur le tableau 3-7.

Nous pouvons constater que le taux de reconnaissance n'est pas le même pour toutes les formes. Certaines sont plus difficiles à reconnaître que d'autres.

	Mur	sommet	Impasse
Apprentissage	3	3	3
Généralisation	5	4	5
	Plan incliné droite	Plan incliné gauche	Couloir
Apprentissage	3	3	3
Généralisation	3	3	4

Tableau 3.7 Taux de reconnaissance de quelques formes géométriques
Qui constituent les bases des données

b- Résolution des indécisions

Que doit répondre le réseau lorsqu'il se trouve dans des situations d'indécision? Une solution insiste à inclure ces situations d'indécision dans l'une des classes entre lesquelles le réseau hésite. 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.

c- Tests d'invariance par déformation, translation et homothétie

Le réseau permet jusqu'à un certain point de reconnaître des formes indépendamment de transformations telles que translation, homothétie et déformation. Pour des formes géométriques simples cette invariance se vérifie assez bien.

c-1- Déformation :

Dans ce test, la performance du réseau est obtenue par une déformation des formes qui constituent les bases d'apprentissage (formes bruitées) et c'est ainsi que le réseau peut reconnaître l'objet de la figure 3.16 Comme appartenant à la même classe que l'objet de la figure 3.15

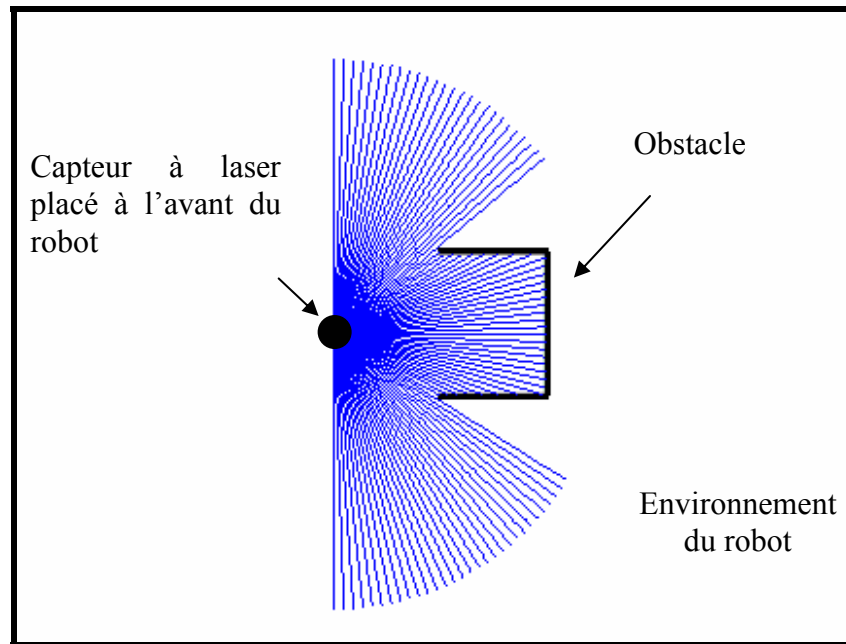


Figure 3.15 Impasse en U durant l'apprentissage

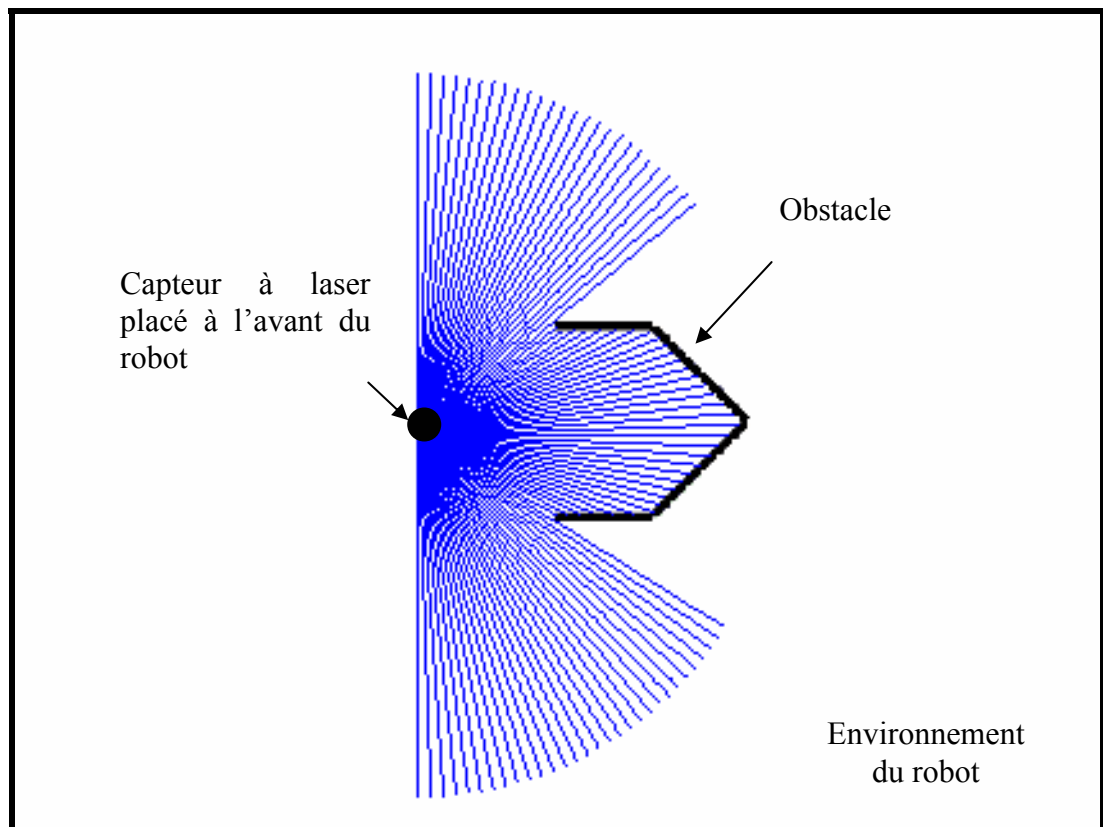


Figure 3.16 Forme reconnue comme un U lors de la validation

c-2- Translation :

Dans ce test, la performance du réseau est vérifiée par tester la reconnaissance lors de déplacement du capteur (robot mobile) vis-à-vis de l'obstacle. C'est ainsi que le réseau peut reconnaître l'objet de la figure 3.18 comme appartenant à la même classe que l'objet de la figure 3.17 pour différentes distances séparant la robot de l'obstacle. Il est à remarquer ici que l'obstacle reste toujours à l'intérieur de la zone critique de reconnaissance de formes..

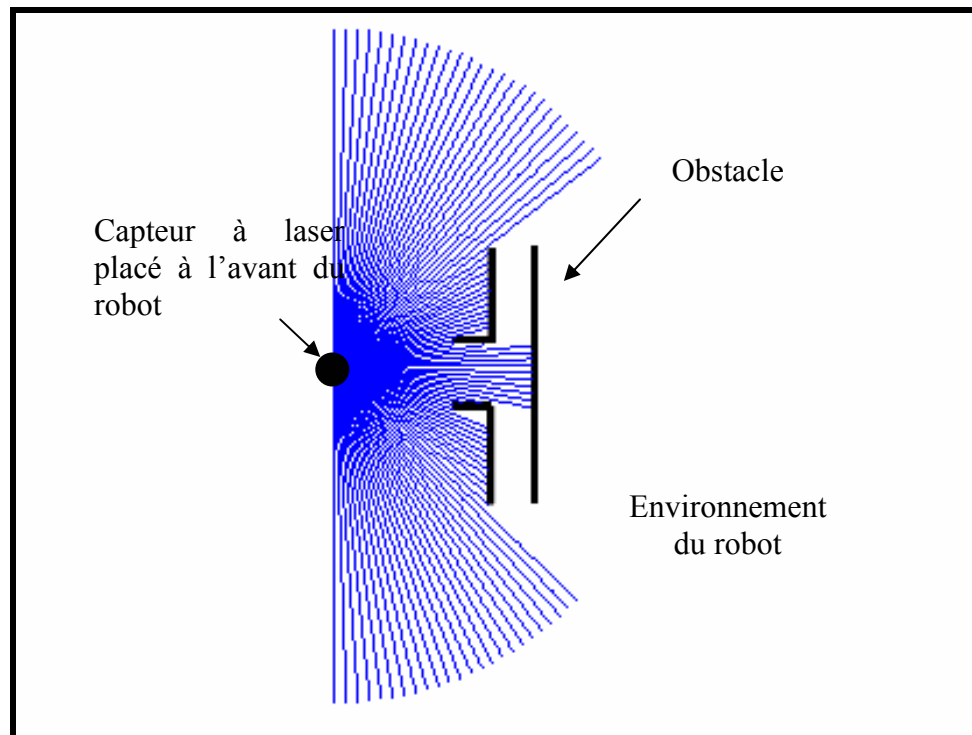


Figure 3.17 *Forme en T durant l'apprentissage*

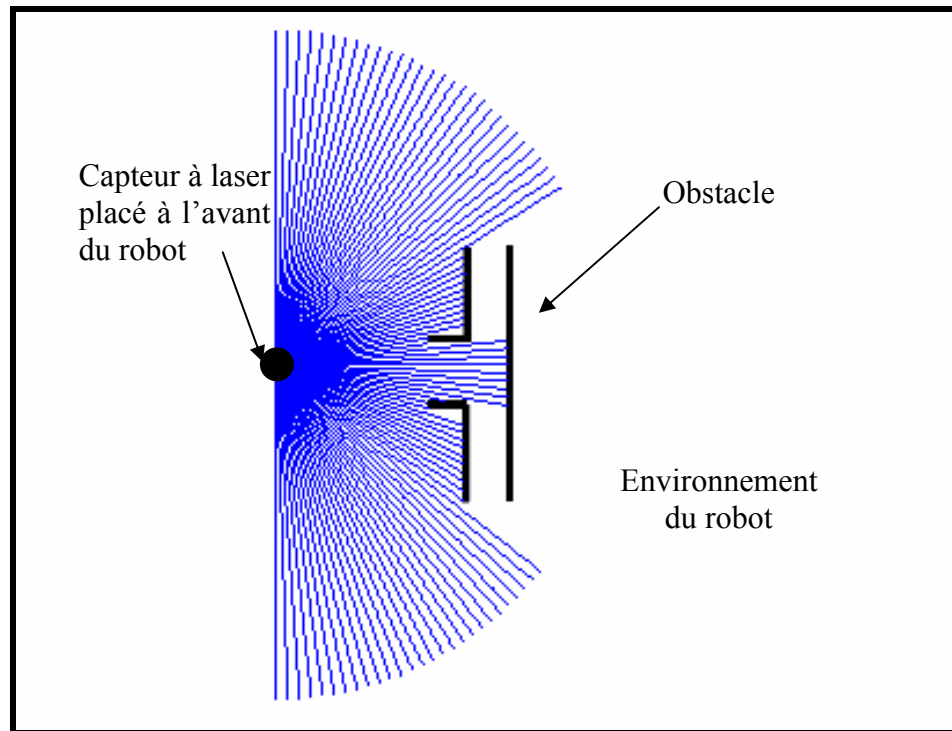


Figure 3.18 reconnaissance d'un T après son éloignement du capteur

c-3- Homothétie :

L'adaptation au changement des tailles de l'environnement est une technique qui nous permet de bien tester la robustesse du réseau de neurones utilisé ainsi que toute la méthodologie suivie pour le test du système de perception mis en œuvre.

Dans ce test la performance du réseau est testée par la multiplication des dimensions des obstacles par des facteurs d'homothétie. Les facteurs retenus pour les tests sont : 0.7, 0.8, 0.9, 1.1 et 1.2 ce qui correspond respectivement à un rétrécissement et à un grossissement des obstacles. Les figures 3.19 et 3.20 montrent les configurations utilisées pour la reconnaissance de la forme de l'obstacle après sa multiplication par les facteurs d'homothétie de 0,7 et de 1,2.

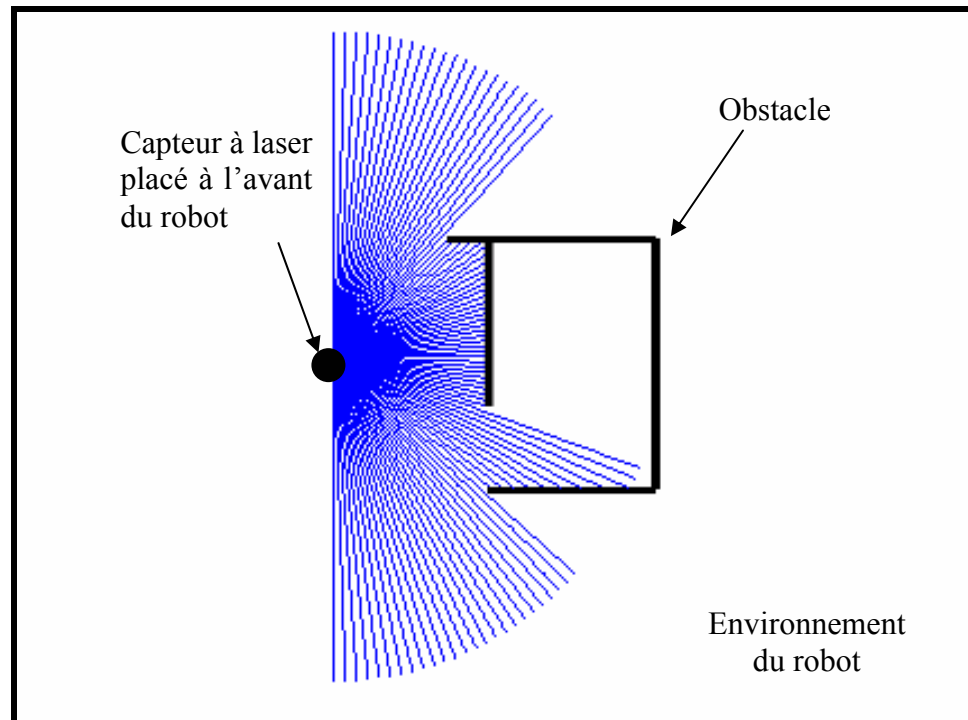


Figure 3.18 *Forme d'obstacle durant l'apprentissage*

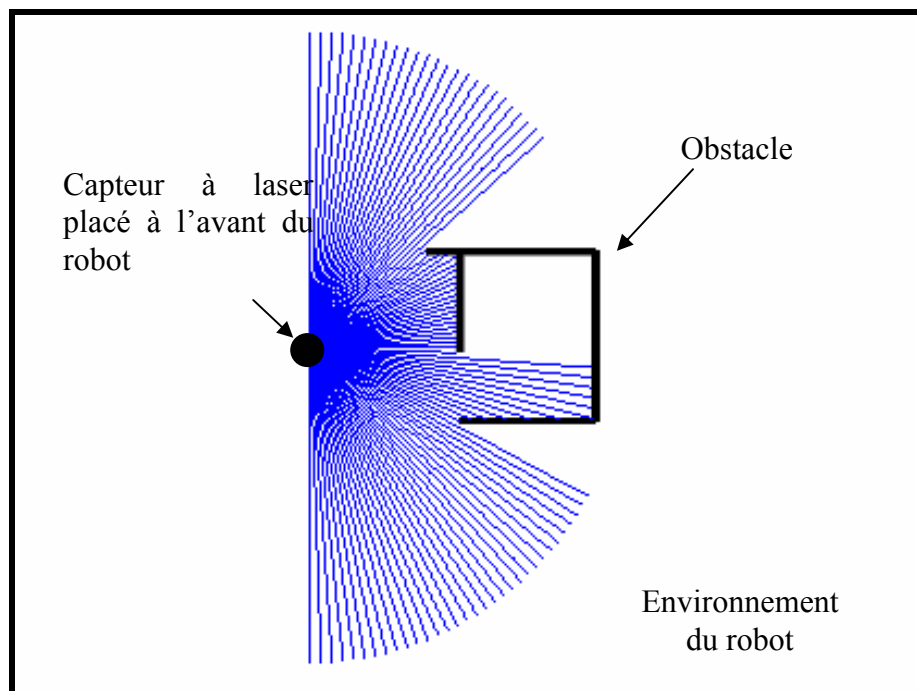


Figure 3.19 *Reconnaissance de la forme d'obstacle de la figure 3-18*

Facteur d'homothétie = 0,7.

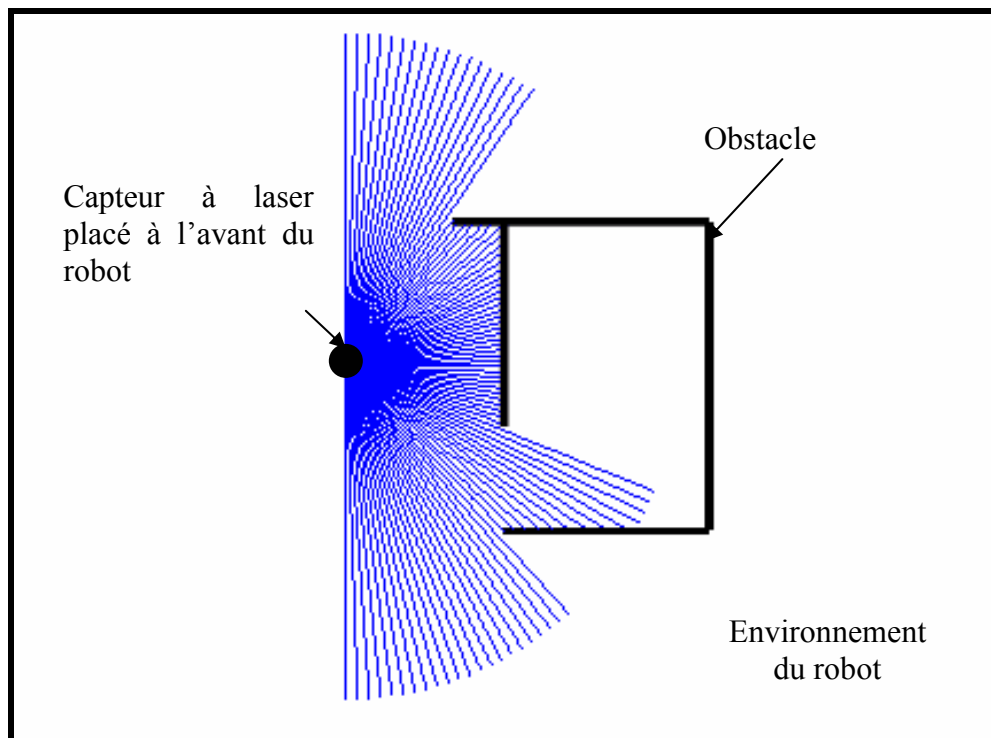


Figure 3.20 Reconnaissance de la forme d'obstacle de la figure 3-18

Facteur d'homothétie =1,2.

Le tableau 3-8 montre le taux de reconnaissance de différentes formes pour 5 valeurs de facteur d'homothétie. Cinq valeurs pour le facteur d'homothétie ont été choisies pour montrer le cas de grossissement et rétrécissement des obstacles.

Facteurs d'homothétie	Couloir	Mur	Impasse
0.8	5	5	5
0.9	5	5	5
1	5	5	5
1.1	5	5	5
1.2	5	5	5

Facteurs d'homothétie	Coin	Croisement	UN T
0.8	5	5	5
0.9	5	5	5
1	5	5	5
1.1	5	5	5
1.2	5	5	5

Facteurs d'homothétie	Pièce	Angle_D	Angle_G
0.8	5	4	4
0.9	5	5	5
1	5	5	5
1.1	5	5	5
1.2	5	4	3

Facteurs d'homothétie	Entrée	Sortie
0.8	5	5
0.9	5	5
1	5	5
1.1	5	5
1.2	5	5

Tableau 3.8 *Taux de reconnaissance d'une même forme d'obstacle pour différentes dimensions*

Les résultats montrés dans le tableau 3-8 montrent des taux de reconnaissance très intéressants. On constate que le réseau de neurones, que nous avons conçu, réagit bien lors de la multiplication par des facteurs d'homothétie qui simulent des changements des dimensions des formes d'obstacles pour le robot.

d- Matrice De Confusion

L'objectif de ce test est de résumer, sous forme d'une matrice, les cas de confusion entre les différentes formes que peuvent prendre les obstacles dans un environnement d'intérieur.

Le but de la matrice de confusion est de voir et sentir le chevauchement et le conflit des situations.

	Couloir	Impasse	Entrée	Pièce	Croisement	UN T	Mur	Coin
Couloir	100%	0%	0%	0%	0%	0%	0%	0%
Impasse	6.66%	93.34%	0%	0%	0%	0%	0%	0%
Entrée	0%	0%	93.34	0%	0%	6.66%	0%	0%
Pièce	0%	0%	0%	100%	0%	0%	0%	0%
Croisement	0%	0%	0%	0%	86.67%	13.33%	0%	0%
UN T	0%	0%	0%	0%	6.66%	93.34%	0%	0%
mur	0%	0%	0%	0%	0%	6.66%	93.34	0%
coin	0%	0%	0%	0%	0%	0%	0%	100%

Tableau 3.9 la matrice de confusion des différentes situations

On remarque que le taux d'erreur peut être considéré comme négligeable pour les 8 formes choisies.

3-5 Conclusion

Après tous ces tests, les résultats obtenus montrent que le système conçu - structure et algorithme d'apprentissage- est robuste face à des situations bruitées ce qui prouve la bonne généralisation du réseau. Nous pouvons conclure que la fiabilité, robustesse et les bonnes performances pour un réseau de neurones dépendent de sa structure et de l'algorithme d'apprentissage utilisé tout en tenant compte du type d'application.

Les résultats que nous avons obtenus et que nous avons présentés dans ce chapitre sont assez satisfaisants du point de vue reconnaissance de forme.

Le développement d'un réseau de neurones se scinde en deux parties. La première partie concerne l'architecture réseau. La deuxième partie est consacrée au dimensionnement du réseau : les entrées, le nombre de couches cachées, le nombre de neurones pour chaque couche cachée ainsi que le nombre de neurones pour la couche de sortie. C'est aussi au niveau de cette deuxième partie que les fonctions d'activation de chaque couche et ainsi que leurs paramètres seront sélectionnées.

Conclusion Générale

Conclusion Générale

L'objectif principal de ce travail est le développement d'une approche qui permet à un robot mobile d'intérieur d'identifier la forme des obstacles se trouvant sur son chemin. Bien que beaucoup de travaux dans ce domaine ont été réalisés, le critère crucial à prendre en considération concerne la navigation en temps réel. Le problème que nous nous sommes posés est de trouver l'outil ou la méthode qui permet de répartir les différentes formes d'obstacles en classe ; le choix de l'application des réseaux de neurones s'est imposé.

Le développement de cette approche a été fondé sur deux étapes. En premier, niveau conceptuel, il s'agit de proposer une architecture pour le réseau de neurones qu'il faut utiliser pour la classification des formes d'obstacles d'un environnement d'intérieur. En second, niveau de mise en œuvre du système, dans cette partie le but est de trouver le moyen permettant le test et la généralisation. Du moment que le travail réalisé s'est limité à un travail de simulation, la simulation d'un télémètre laser, monté en tourelle à l'avant du robot, a été réalisée.

Pour l'opération de classement, nous avons essayé de maintenir la structure proposée la simple possible. En effet, au lieu de réserver par chaque classe une sortie du réseau ce qui augmente considérablement le nombre de neurones dans cette dernière, nous avons limité à ce niveau le nombre de neurones à 4 pouvant décoder 2^4 soit 16 formes possibles. C'est ainsi que le réseau conçu a pu discriminer un nombre important de formes tout en maintenant une structure simple.

Les composantes du vecteur d'entrée, que ce soit pour l'apprentissage ou la généralisation, sont issues des valeurs des distances séparant le robot mobile de

l'obstacle se trouvant dans la zone critique de perception délimitée au préalable suivant les dimensions et la manoeuvrabilité du robot.

Les résultats obtenus montrent, qu'avec un nombre restreint de formes d'obstacles lors de l'apprentissage, une bonne classification au cours des tests de généralisation.

Cependant, la fonction principale d'un robot mobile est d'atteindre son objectif, il est donc nécessaire d'intégrer ce module dans un contrôle pour permettre de générer la commande adéquate pour le contournement de l'obstacle suivant sa forme. Cette dernière idée, nous l'avons ajouté dans cette conclusion comme des perspectives pour des travaux futurs.

Bibliographie

Bibliographie

- [1] **I. Mesbahi a. Benyettou f. Hendel n.e berrached,**'' *Reactive navigation of mobile robot by temporal radial basis function approach*''; 0-7803-8560-8/04/IEEE, 2004
- [2] **Marc parizeau,** ''Réseaux de neurones'', édition 2004
- [3] **Jian-kang wu**''*Neural networks and simulation methods*''; Editeur: CRC Press (Déc 1993) ISBN 0-8247-9181-9(acid-free paper)
- [4] **Simon haykin**'' *Neural networks a comprehensive foundation*'' ISBN0-13-908385-5 prentice hall international, Inc pages: 34-78, second edition, 1999
- [5] **Ben J.A.krose P.patrick van der smagt,** '' *An introduction to neural networks*'' university of amsterdam faculty of mathematics and computer science: pages 33-42, Fifth edition january 1993
- [6] **Shih-chi huang and yih-fang huang,** ''*Learning algorithms for perceptrons using back-propagation with selective update*'' paper, 1990
- [7] **Xiao li, Jeff bilmes and Jonathan malkin,**'' *Maximum margin learning and adaptation of mlp classifiers*'' , edition 2005
- [8] **M.talibi-alaoui,R.touahni et A.sbihi,** ''*Classification des images couleurs par association des transformations morphologiques aux cartes de kohonen*'' ; 2004
- [9] **Bernard Dubuisson,**'' *Diagnostic et reconnaissance des formes*'' Hermès, paris, 1990 ISBN 2-86601-240-2
- [10] **Le cum and bengio,** '*Pattern recognition and neural network*'', 1995
- [11] Publié sous la direction de **Murat Kunt,** *Reconnaissance de formes et analyse de scène ; Presses polytechniques et universitaires romandes*, 2000
- [12] **S. A. Alan,** ''*Recognition of environment and reactive navigation of an autonomous mobile robot by network of neurons*'', Thesis of Doctorate, Evry Val d'Essonne, 1996.

- [13] **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.
- [14] **herve Abdi, Dominique Valentin, et Betty Edelman**, "*neural networks*", juin 1999
- [15] **David FILLIAT**, "*robotique mobile*", cours c10-2 ENSTA pages : 36-48 octobre 2004
- [16] **A. C. Victorino, P. Rives et J.-J. Borrelly**. "*Mobile Robot Navigation Using a Sensor-Based Control Strategy*", In *IEEE International Conference on Robotics and Automation*, may 2001
- [17] **Bernard BAYLE** "*Robotique mobile*" document de l'Ecole Nationale Supérieure de Physique de Strasbourg pages : 21 et pages 31-35, année 2005-2006
- [18] **H.Takahiko**, "*Decision rule for pattern classification by intergrating, interval features values*", *IEEE transaction on pattern analysis and machine intelligence*, vol20, n°4, 1998.
- [19] **G. Von Wichert** ". *Mobile robot localization using a self-organised visual environment representation. Robotics and Autonomous System*" '1998 pages: 185–194
- [20] **R. C. Luo and M. G. Kay**" *Multisensor integration and fusion in intelligent Systems*" *IEEE Transactions on Systems, Man, and Cybernetics*, pages :901–931, année 1989.
- [21] **R. Greiner and R. Isukapalli**." *Learning to select useful landmarks*". *IEEE Transactions on Systems, Man, and Cybernetics-Part B, Special Issue on Learning Autonomous Robots*, 1996.
- [22] **G. Dudek and P. MacKenzie**." *Model-based map construction for robot localization In Proceedings of Vision Interface*"',année 1993
- [23] **Vania ANDRONOVA** "*Utilisation de données météo et des réseaux de neurones pour la prédiction de vitesses de vent*" *Projet de fin d'étude Master 2nd année* Le 18 juillet 2006

