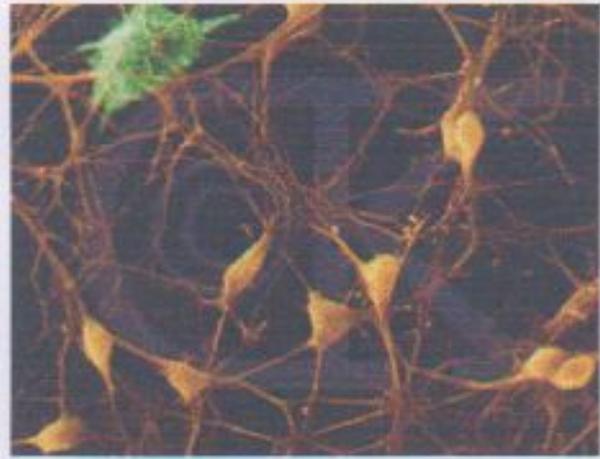
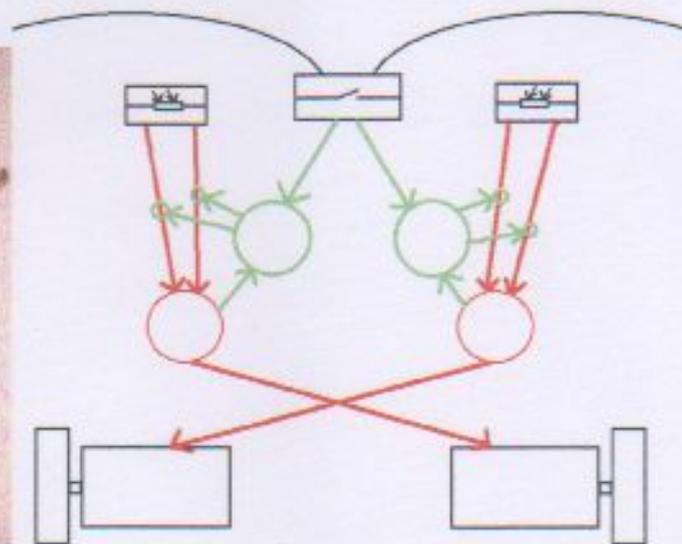


TIPE 2007



**Un réseau neuronal-astrocytaire
pour la robotique autonome**

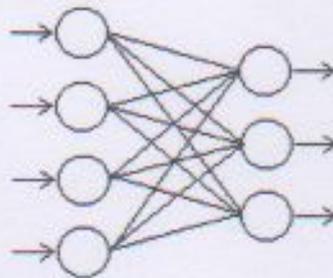


Le réseau neuronal reste actuellement le seul composant informatique capable d'apprendre de l'expérience. Son intérêt réside dans le fait que, bien conditionné, il peut donner une réponse satisfaisante à une situation inconnue, ce qui en fait un outil de choix pour la reconnaissance de l'image ou de l'écriture, et pour gérer un robot autonome.

Historique des réseaux

Ce sont les neurologues **Warren Surgis Mc Cullock** et **Walter Pitts** qui, dans leur article *what the frog's eye tells to the frog's brain*, ont pour la première fois proposé un modèle simplifié au neurone : le **neurone formel**, capable de résoudre des problèmes arithmétiques, mathématiques ou symboliques, mais il faudra attendre 1949 pour que le physiologiste **Donald Hebb** découvre une loi d'apprentissage. La **loi de Hebb**, toujours utilisée de nos jours, stipule que lorsque deux neurones sont activés au même moment, leurs connections se renforcent.

En 1957, **Frank Roseblatt** propose le **perceptron**, réseau simple disposant d'une couche d'entrée (la rétine), et d'une couche de sortie.



En 1968 **Rumelhart** introduit le **perceptron multicouche**, capable de traiter des problèmes non linéaires.

Par la suite, les réseaux neuronaux connaîtront un essor considérable : de nouvelles lois d'apprentissage et de nouvelles formes de réseaux apparaissent.

PLAN :

I Modélisation des neurones

- 1 les neurones : la cellule nerveuse et sa modélisation
- 2 les astrocytes : la cellule gliale et sa modélisation
- 3 modélisation du réseau : représentation et fonctionnement

II Le robot d'étude

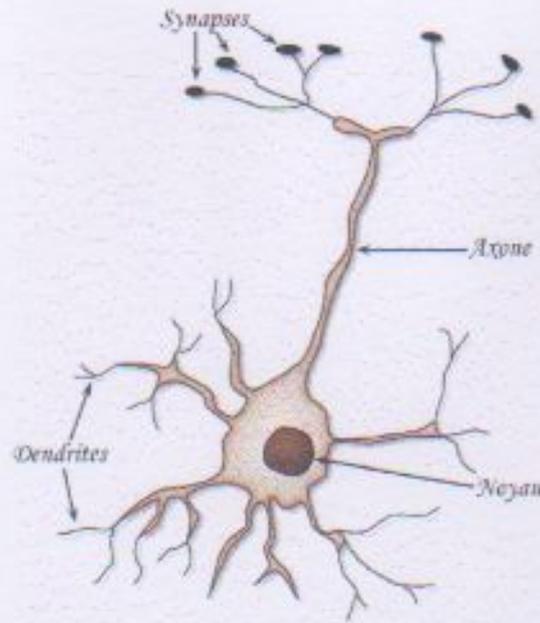
- 1 aspect matériel : l'électronique, la propulsion, les capteurs.
- 2 le réseau du robot
- 3 le programme : langage, fonctionnement global

III Observations

- 1 fonction d'apprentissage : recherche d'une fonction appropriée
- 2 comportement du robot : comportements modélisés, réactions du robot
- 3 durée de l'apprentissage : étude de l'apprentissage dans un environnement.

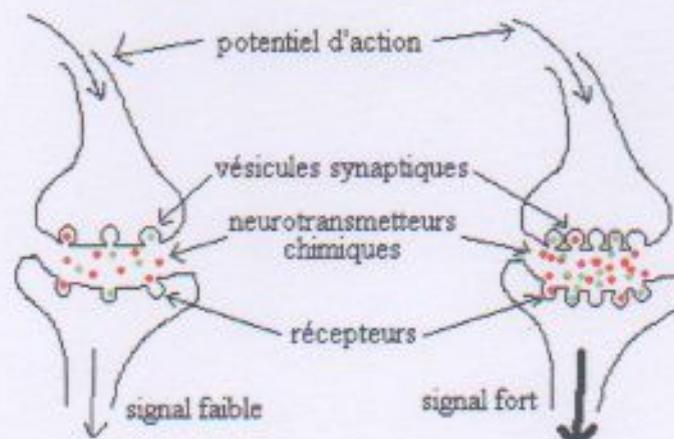
I-1 Le neurone:

Le neurone est une cellule spécialisée dans le traitement de l'information : elle interprète les signaux provenant des neurones en amont par les **dendrites** et transmet le signal par l'**axone** aux neurones en aval. La synapse est la séparation entre l'axone du neurone amont et la dendrite du neurone aval. Dans le cerveau, un neurone peut être lié à plus de 100 000 autres neurones.



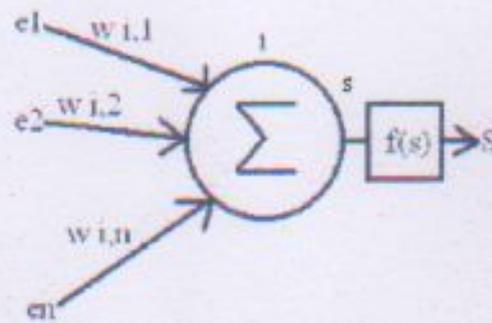
Le principe est le suivant : la **membrane** se comporte comme un ensemble de condensateurs et de résistances en parallèle. Lorsqu'un signal est appliqué sur une dendrite, la membrane se charge. Lorsque celle-ci atteint la valeur seuil de **-30mV**, la membrane se décharge et le neurone envoie, via l'axone, un signal, appelé **potentiel d'action**, de 100 mV.

Au niveau de la synapse, le potentiel d'action est converti en **signal chimique** par des **vésicules synaptiques**, puis reconvertis en signal électrique par des **récepteurs synaptiques**. Le nombre de ces vésicules détermine la valeur du signal appliqué au neurone aval. La modification de ce nombre permet la mémorisation des informations.



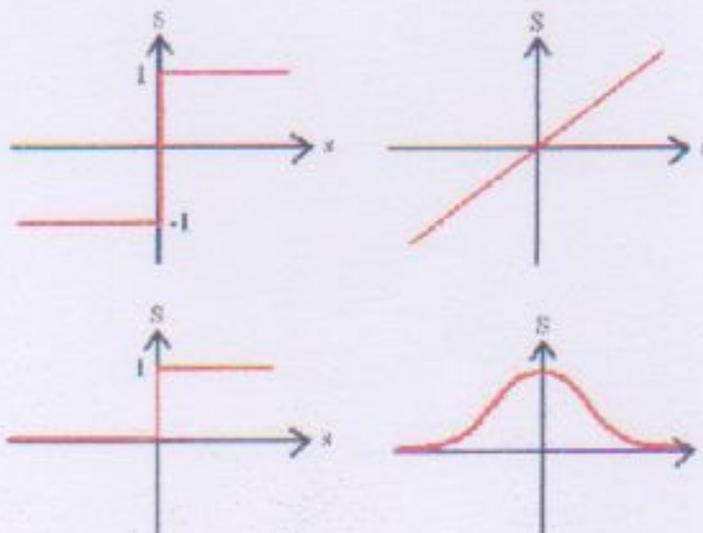
Modélisation : le neurone artificiel est bien différent du neurone naturel par son fonctionnement , mais il s'en rapproche par son principe.

On modélise un neurone par un sommateur , les synapses par des coefficients . Ces coefficients , appelés **poids** , ont la même fonction que la transmission chimique : en multipliant la valeur d'entrée par le poids , on peut amplifier ou réduire celle-ci . Le sommateur fait ensuite la somme des valeurs pondérées. On s'aperçoit que ,comme pour le neurone naturel , les "synapses" jouent un rôle prépondérant dans le traitement de l'information.



$$s = e_1 w_{1,1} + e_2 w_{1,2} + \dots + e_n w_{1,n}$$

On applique ensuite à cette valeur de sortie une fonction dite **fonction de transfert** , qui caractérise le neurone.



Fonction de neurones formels ,

linéaire et gaussien

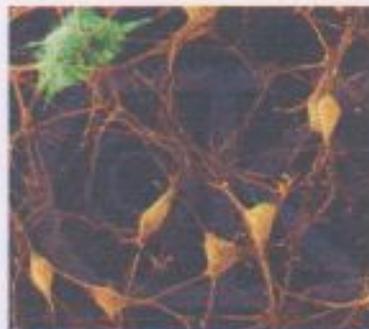
I-2 Les astrocytes

Les astrocytes font parti de la famille des cellules **gliales**, autres constituants du cerveau. Ces cellules sont très nombreuses : chez l'homme, il y a presque 2 astrocytes pour un neurone. On a longtemps pensé que leur fonction était de nourrir les neurones et de leur fournir un support, mais récemment, on a découvert que ces cellules pouvaient communiquer entre elles par **signaux chimiques** (d'où la découverte tardive), en propageant une augmentation du taux de calcium, et forment un réseau aussi complexe que celui des neurones, plus lent que celui-ci, mieux, elles peuvent **intercepter un signal** entre deux neurones et **modifier** ces signaux.



Illustration provenant de Science et Vie n°1058

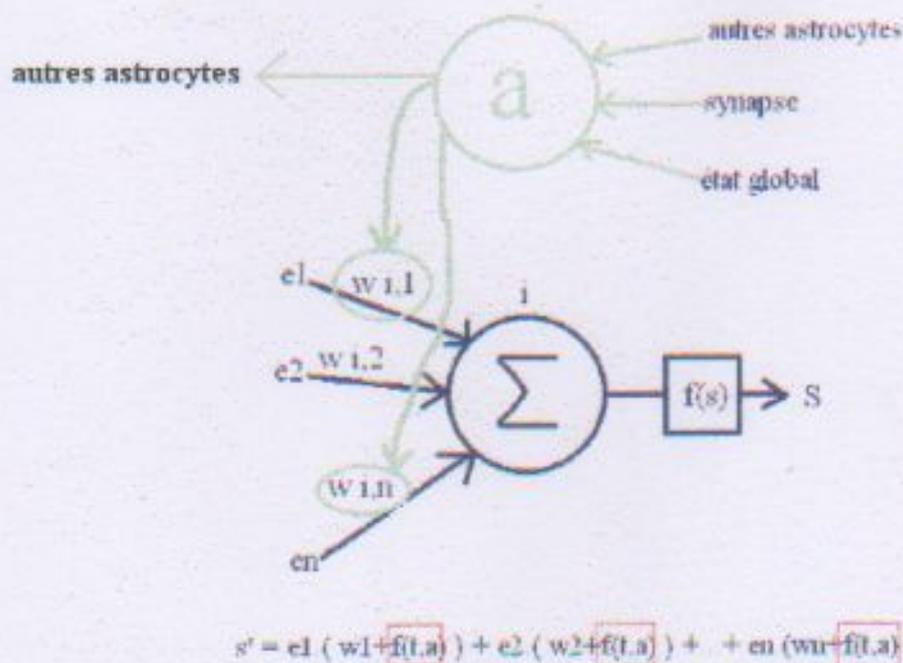
On s'aperçoit donc que le réseau astrocytaire ne forme pas seulement un réseau secondaire, il contrôle le premier, d'où l'importance d'en tenir compte dans un réseau artificiel.



astrocyte (en vert) dans une culture

Modélisation : la modélisation proposée ici se rapproche de celle d'un neurone , à la différence que l'astrocyte artificiel communique plusieurs états à la fois aux autres astrocytes : il transmet le signal d'une **fonction d'apprentissage** , fonction de t à valeur moyenne nulle , qui n'est différente de zéro que sur une période , ainsi qu'un **état de satisfaction** qui correspond à l'efficacité du système. L'astrocyte interprète les signaux provenant des neurones et des astrocytes , et le transmet à d'autres astrocytes ou aux **synapses** auquel il est relié.

Le réseau astrocytaire crée en tout point un état de satisfaction local à partir des astrocytes en amont ainsi que des neurones auxquels il est connecté . les astrocytes transmettent aussi la fonction d'apprentissage : lorsqu'une période de la fonction d'apprentissage a parcouru un astrocyte en amont , elle parcourt l'astrocyte en aval . Cette fonction se propage ainsi de proche en proche dans le réseau comme une onde.



Cette fonction permet l'apprentissage grâce à cette règle:

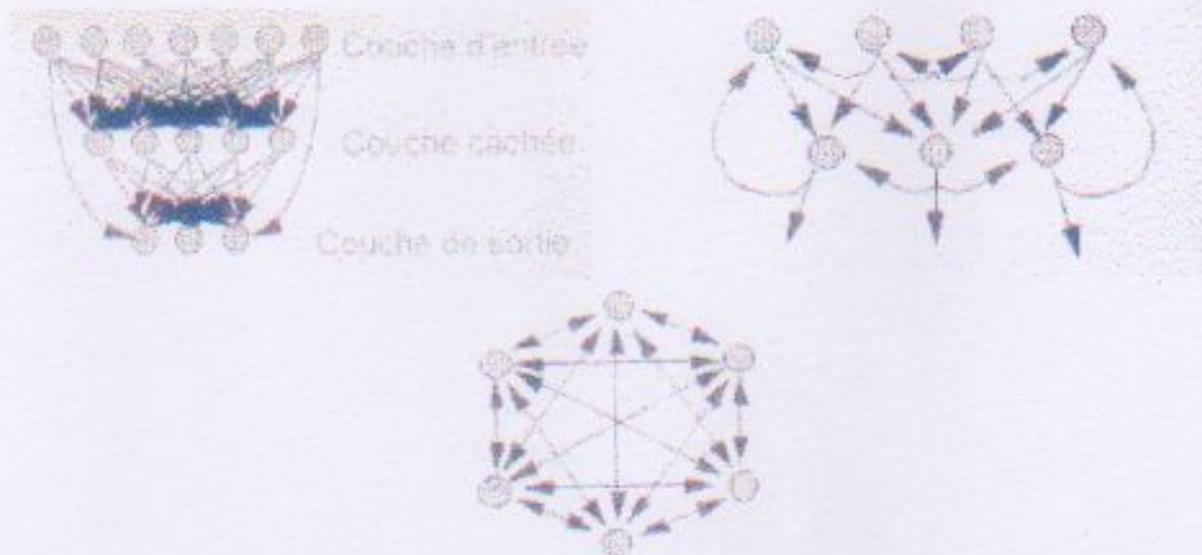
Lorsque cette fonction traverse l'astrocyte , on ajoute f(t) au poids des synapses reliés . Si pendant que f(t) augmente , on constate que l'état de satisfaction local augmente , on augmente la valeur des poids proportionnellement aux entrée correspondantes (donc à leur participation) ,car une augmentation du poids améliore le réseau, on les diminue si il décroît . on effectue l'inverse si f(t) décroît.

Améliorations de cette loi :- l'amplitude et la durée de la période de la fonction d'apprentissage peuvent dépendre de l'état de satisfaction.

- si le nombre de synapse est trop important , les astrocytes se répartissent au hasard, les connections changent de synapse si la modification de l'état de satisfaction est trop faible.

I-3 Modélisation des réseaux:

Les neurones sont regroupés en réseaux , c'est à dire une structure où chaque neurone est connecté à un certain nombre d'autres neurones . La disposition des neurones et la règle de connexions caractérisent le type de réseaux . Parmi les plus utilisés , on trouve le perceptron multicouche , les réseaux à connexions récurrentes (connexion avec des neurones d'une couche en amont) et à connexion totale (chaque neurone est connecté à tous les autres) :



Comme on l'a vu , la capacité de calcul et de mémorisation d'un réseau est dû à la modification des poids des synapses , ce qui nécessite une **loi d'apprentissage** . Il en existe un grand nombre , et leur utilisation dépend de la méthode d'apprentissage : un réseau sera dit **supervisé** si on lui présente un certain nombre d'entrées avec les sorties associés (pour la reconnaissance d'une écriture manuscrite , par exemple) , et **non supervisé** si on laisse le réseau évoluer seul (dans le cas d'un robot ,par exemple)

Les plus utilisés sont la règle de Widrow-Hoff (les poids des connexions actives sont modifiés si la réponse n'est pas celle attendue) , dans le cas des réseaux supervisés , et la loi de Hebb (deux neurones activés au même moment renforcent leur connexion) , utilisé pour les réseaux non supervisés.

La règle d'apprentissage retenue ici sera celle utilisant les astrocytes artificiels.

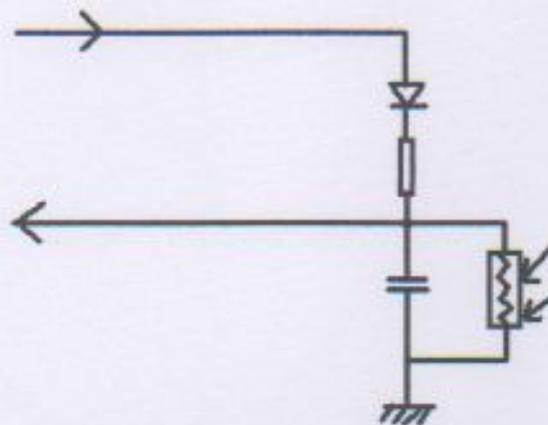
II-1 aspect matériel

L'étude du réseau astrocytaire sera menée sur un petit robot mobile.

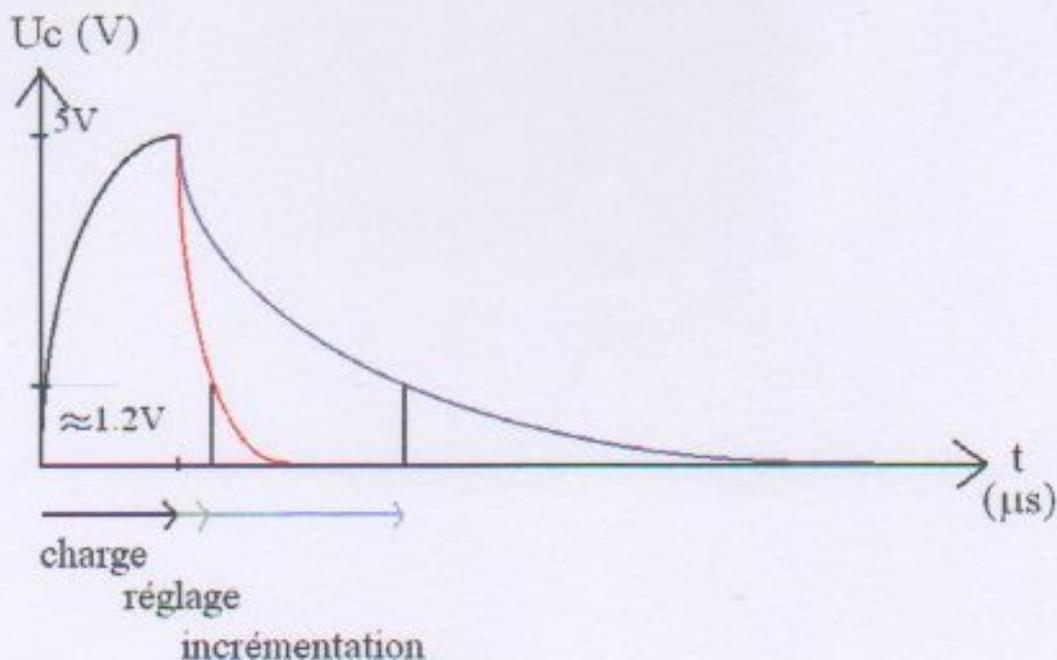
Ce robot est géré par un microcontrôleur 8 bits, le **PIC 16F84**. Doté d'une mémoire de 1 Ko, de 13 entrées/sorties et facile à programmer, il convient parfaitement pour cette application.

Pour la propulsion, j'ai opté pour deux **servocommandes de modélismes modifiées** pour permettre une rotation continue : en effet, les servocommandes ne nécessitent pas d'interface de puissance et l'électronique embarqué permet de les commander en vitesse et en sens de rotation à l'aide d'impulsions de 1 à 2ms à répéter toutes les 18ms.

Le robot dispose de 3 capteurs : un **capteur de contact** type "antennes", et 2 photo résistances faisant office de capteurs de luminosité. Ces derniers fournissant une information analogique, j'ai opté pour une conversion **résistance-temps** pour l'interfaçage.



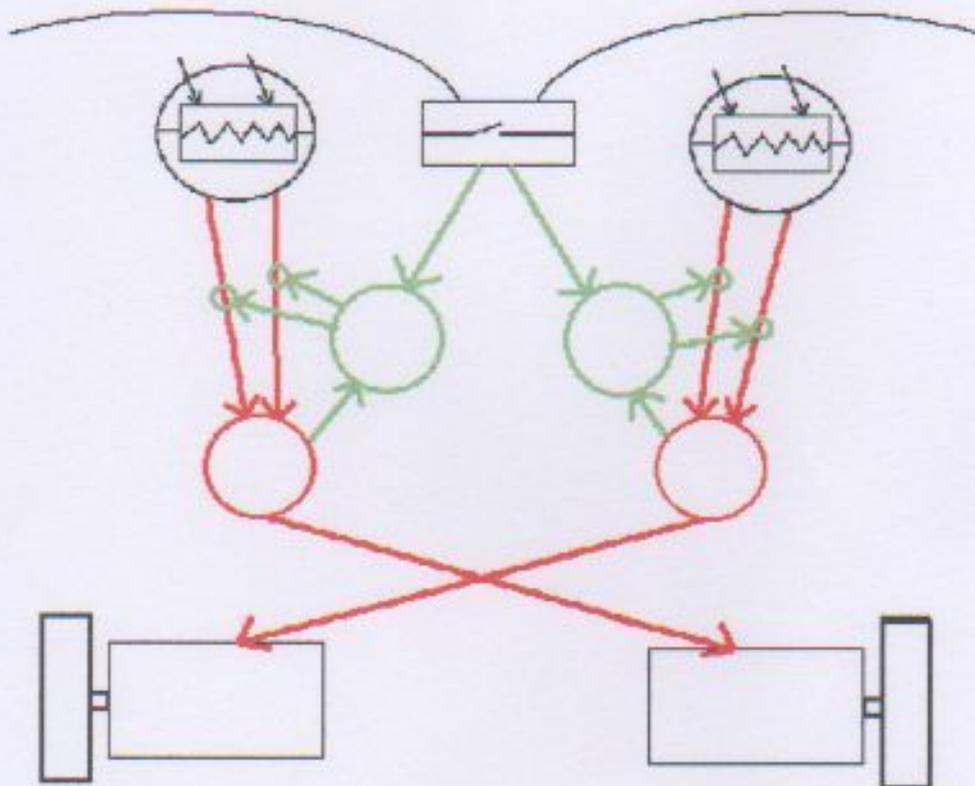
Le principe est le suivant : on charge un condensateur pendant $250\mu\text{s}$, puis on le laisse se décharger à travers la photo résistance. Plus la luminosité sera grande, plus la résistance sera faible, et plus le condensateur se déchargera vite. On incrémente une variable tant que le niveau logique reste à 1 :



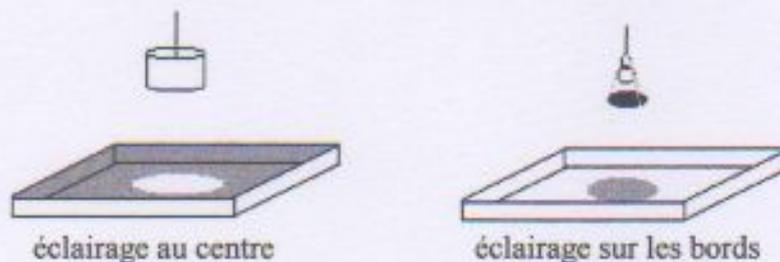
II-2 Le réseau du robot :

Le réseau utilisé est constitué de 5 neurones : 3 en entrée et 2 en sortie , et de deux astrocytes . Chaque capteur de luminosité envoi deux informations à un même neurone : une proportionnelle à la luminosité et l'autre à l'obscurité , chaque connexion disposant de son propre poids. Les neurones envoient ensuite le résultat au moteur du coté opposé au capteur , ce qui permet d'obtenir un comportement photovore ou photophobe .

Les astrocytes sont rebouclé sur eux-même , pour permettre une fluctuation due à la fonction d'apprentissage en continue . L'état de satisfaction tient compte de la vitesse de rotation du moteur associé , fournie par le neurone , et du capteur de contact . Comme ce dernier est un capteur qui fonctionne en tout ou rien , on considère que l'état interne diminue tant que le capteur est fermé.



Le but de ce robot est d'aller le plus vite possible en évitant autant que possible de s'approcher des obstacles . L'environnement du robot est constitué d'une sorte d' "enclos" , avec deux conditions d'éclairage : soit le centre est éclairé , soit les bords sont éclairés . Le robot devra ainsi , dans chaque condition , faire le rapport entre obstacle et éclairage , et adapter son comportement (photovore ou photophobe) , pour s'éloigner des obstacles.



II-3 Le programme

Le programme est édité en **BASIC F84**, un langage de programmation dédié au PIC 16F84, comprenant es instruction en **BASIC** complétées par des instructions en **assembleur**.

La valeur fournie par les capteurs de luminosité s'échelonne sur 6 valeurs (de 1 à 6), les poids sur 14 valeurs. La somme des informations fournies par un capteur de luminosité (luminosité et obscurité) est toujours égale à 6, ainsi, seule la différence entre poids associés détermine le comportement. compte tenu du faible nombre de valeurs disponibles pour les poids, la modification de ceux-ci ne sera pas proportionnelle aux valeurs fournies par les capteurs mais seul le poids de la connexion la plus active sera modifié.

Fonctionnement du programme :

on initialise le programme et on affecte aux variables les conditions initiales

-on lance une boucle de 50 cycles

-on prélève les valeurs fournies par le capteur de luminosité droit

-le neurone droit calcule la sortie

-l'astrocyte ajoute la fonction d'apprentissage

-on prélève les valeurs fournies par le capteur de luminosité gauche

-le neurone gauche calcule la sortie

-l'astrocyte ajoute la fonction d'apprentissage

-on envoi l'impulsion au moteur gauche

-on envoi l'impulsion au moteur droit

-on attend 18 ms

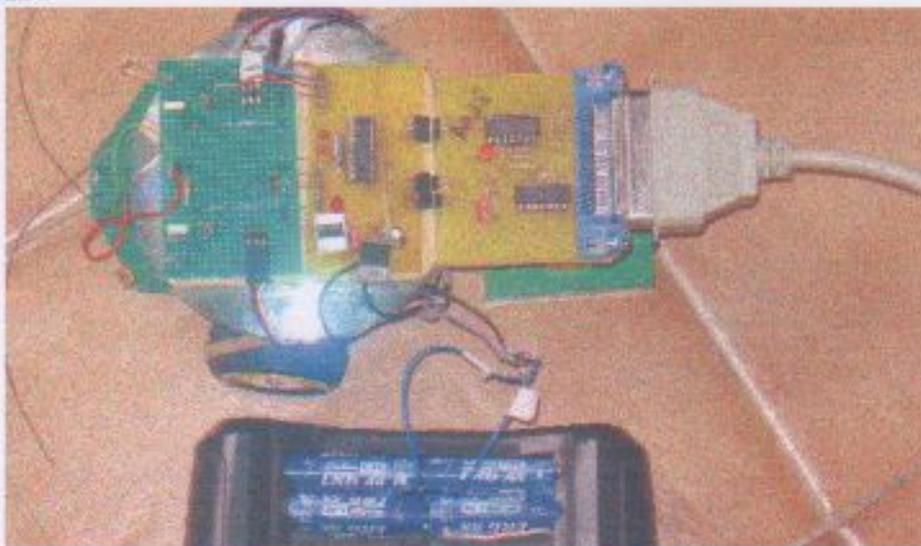
-fin de boucle

-on compare le nouveau et l'ancien état de satisfaction et on modifie en conséquence les poids des connexions les plus actives

-on modifie la fonction d'apprentissage

-on retourne au début du programme

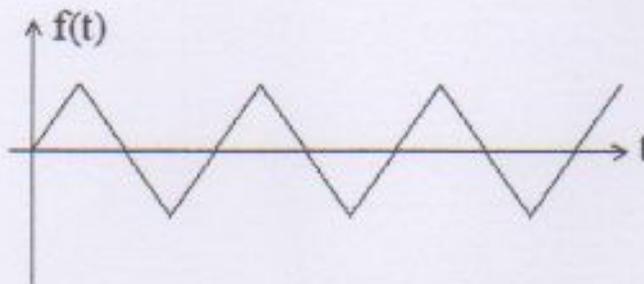
Le programme est téléchargé dans la mémoire du PIC à l'aide d'une carte de programmation relié au port parallèle d'un ordinateur. Il est à noter que pendant le téléchargement, le PIC doit être alimenté sous 12 V. L'alimentation que j'ai utilisé était le coupleur de 8 piles d'une radiocommande :



III-1 Fonction d'apprentissage

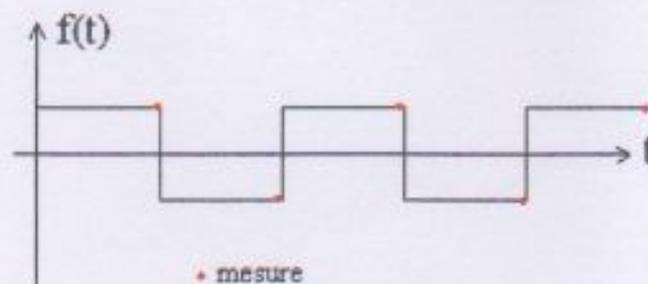
Le PIC m'a permis de tester deux types de fonctions d'apprentissage . Le test a été effectué sans capteur , le robot devait seulement aller aussi vite que possible .

La première fonction testé est la fonction triangulaire : à chaque cycle , la variable $f(t)$ était augmentée ou diminuée de 1 . Comme il y a variation à chaque cycle , on peut mesurer les variations de l'état de satisfaction à chaque cycle . Le robot trouve la vitesse maximale très rapidement (moins de 2 secondes) , trop rapidement pour garder un état en "mémoire" . Cette fonction peut donc convenir pour un système qui doit suivre un but qui varie rapidement , mais elle n'est pas adaptée au robot d'étude , qui doit conserver un comportement plus longtemps.



Fonction triangulaire : on tient compte des modifications à chaque instant

La seconde fonction est un signal carré . on relève l'état de satisfaction avant l'inversion du signe de $f(t)$ pour ne pas tenir compte de l'état transitoire . Cette fois-ci , le robot met plus de temps pour trouver la vitesse maximale , environ 30 à 40 secondes. Cette fonction permet donc de conserver un certain temps les comportements acquis.



fonction carrée : on tient compte des modifications avant le changement de signe

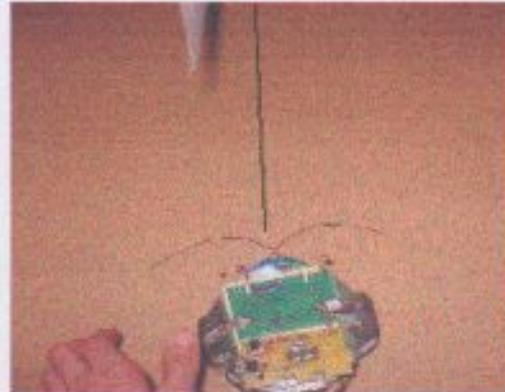
C'est donc le signal carré qui a été choisi comme fonction d'apprentissage pour le robot

III-2 Étude du comportement

1^{ère} expérience : on sépare l'enclos en 2 parties : une éclairée , une dans l'obscurité
on place ensuite le robot dans la partie éclairée et on le laisse évoluer jusqu'à ce qu'il atteigne la vitesse maximale .

On place ensuite le robot sur la frontière entre les deux parties

(note , à cause du faible éclairage , un flash a été nécessaire , les zones d'ombres n'apparaissent pas sur ces photos)



partie éclairée

partie sombre

Observations : le robot se tourne toujours vers la zone éclairée



on effectue l'opération inverse : on laisse évoluer le robot dans l'obscurité . Lorsqu'on le place sur la frontière , le robot se tourne vers l'obscurité

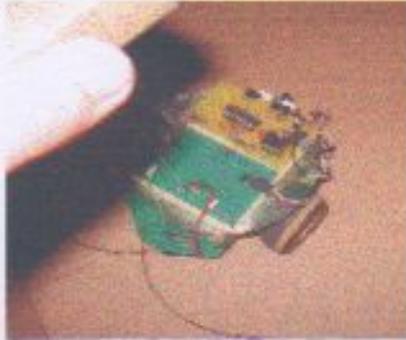
On s'aperçoit donc que le robot préfère toujours les conditions qu'il connaît aux conditions inconnues

2^{ème} expérience : on part des mêmes conditions initiales . Lorsque le robot roule à pleine vitesse , on le place en face de la frontière .

Observations : le robot avance à pleine vitesse , puis lorsqu'il atteint la frontière , il s'arrête puis redémarre lentement , avant d'accélérer progressivement .

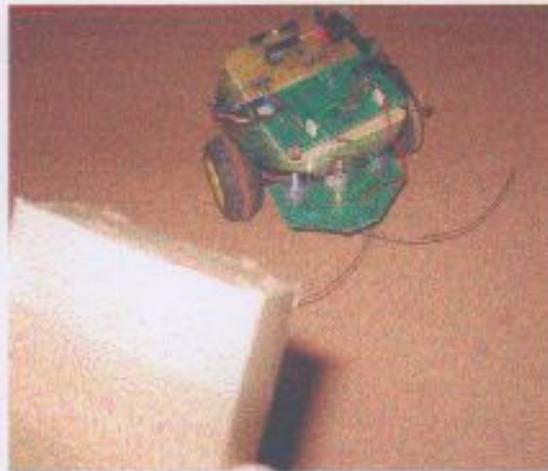
Le robot simule ainsi une appréhension face à l'inconnu.

3^{eme} expérience : on laisse le robot atteindre , sous une lampe , sa vitesse maximale. Puis on le laisse évoluer dans l'ombre . On place le robot face à un obstacle , dans l'obscurité . Le robot le heurte , et après quelques itérations , recule .



Observations : on oriente ensuite la lampe vers le robot : celui-ci se met à la suivre , le robot a adopté un comportement photovore .

Le robot a donc "appris" que les obstacles sont dans l'obscurité et se met à chercher les zones lumineuses.



4^{eme} expérience : les conditions d'éclairage sont inversées par rapport à celles de l'expérience 3 : l'obstacle est dans une zone éclairée .



Observation : lorsqu'on oriente la lampe vers le robot , celui-ci a tendance à l'éviter : il a adopté un comportement photophobe. Le robot a donc "appris" que les obstacles sont dans les zones éclairées et les évites .



5^{eme} expérience : on part de l'expérience 4 :le robot photophobe . On lui fait heurter un obstacle placé dans l'obscurité .

Observations : le robot fini par reculer , puis continu en marche arrière , quelque soient les conditions d'éclairage . On oriente la lampe vers le robot . Celui-ci fini par repartir en marche avant et se met à suivre la lumière , il est devenu photovore .

Le robot montre ainsi une certaine capacité d'adaptation : les obstacles sont désormais dans l'ombre et il peut avancer dans la lumière "sans risque" . Il passe donc de photophobe à photovore pour s'adapter à ces nouvelles conditions.

III-3 Durée d'apprentissage:

6^{eme} expérience : on éclaire cette fois le centre de l'enclos et on laisse le robot évoluer seul .

Observations : le robot démarre lentement , puis accélère . Il atteint sa vitesse maximale au bout de 4 itérations (20 secondes)



Lorsqu'il sort de la zone éclairée , il s'arrête , puis repart . Il finit par heurter un mur .Le robot fait alors varier la vitesse de ses moteurs .



Après quelques itérations , le robot fini par reculer au bout de 20 secondes environ . Le robot recule , et , n'avançant pas droit , heurte un autre mur en marche arrière.



Il fini par repartir en marche avant.

Il passe sous la zone éclairée sans ralentir , puis continu tout droit .

Cette fois , lorsqu'il arrive à la limite de la zone éclairée , il pivote et se met à tourner autours du centre de la zone éclairée . Le robot adopte un comportement photovore pour aller aussi vite que possible sans toucher les obstacles .



Il lui aura fallu 1min 5 pour acquérir ce comportement . 2 minutes plus tard , le robot inverse le sens de sa rotation .



Au total , il restera un peu plus de 5 minutes dans la zone éclairée avant de repartir vers l'obscurité .

Le robot recommence alors le cycle . Mais pendant qu'il tourne autour de la zone éclairée , on éclaire le mur situé devant le robot . Celui-ci fonce sur le mur.



Le robot se met alors à reculer et ce , quelle que soit la condition d'éclairage . Le robot fini par avancer dans l'obscurité et se met à éviter les obstacles éclairés . Le robot est donc devenu photophobe . Il lui aura fallu un peu plus de 3 minutes pour cela .

Lorsqu'on diminue la période de la fonction d'apprentissage , le robot apprend plus vite , mais reste moins longtemps au centre de l'enclos . Au contraire , si on l'augmente , l'apprentissage est plus lent mais le robot restera plus longtemps loin des obstacles

Le robot montre dans cette expérience qu'il est capable de s'adapter à un environnement dont les conditions peuvent changer.

Conclusion :

Même si il peut encore être amélioré (la période de la fonction d'apprentissage peut dépendre de l'état de satisfaction) , le réseau astrocytaire montre qu'il peut permettre un apprentissage non supervisé et une optimisation d'un réseau neuronal .

Ce réseau peut être utilisé dans le domaine de la robotique autonome pour optimiser un réseau neuronal mais pourrait aussi être utilisé sur un robot industriel : en faisant légèrement varier les temporisations ou la position à atteindre à chaque cycle , on peut optimiser un automate et le rendre moins sensible à l'usure de ses composants .