Implémentation VLSI analogique des réseaux de Kohonen

Autor(en): **Heim, Pascal / Arreguit, Xavier / Vittoz, Eric**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **83 (1992)**

Heft 5

PDF erstellt am: **10.07.2024**

Persistenter Link: https://doi.org/10.5169/seals-902805

Implémentation VLSI analogique des réseaux de Kohonen

Pascal Heim, Xavier Arreguit, Eric Vittoz

Cet article décrit les possibilités d'implémentation du réseau de Kohonen au moyen de techniques VLSI analogiques. Il ressort que l'implémentation des fonctions collectives est particulièrement bien adaptée à l'analogique grâce à l'utilisation opportune des caractéristiques non-linéaires des transistors. L'emploi des mémoires analogiques existantes est actuellement le problème le plus délicat, et un effort est encore à faire avant de pouvoir disposer de circuits efficaces.

Der Artikel befasst sich mit den Möglichkeiten der Implementierung von Kohonen-Netzwerken in VLSI-Technik. Es wird gezeigt, dass sich die Analogtechnik für die Realisierung von kollektiven Funktionen speziell gut eignet, vor allem wenn man die nichtlinearen Eigenschaften der Transistoren nutzt. Das grösste Problem bei dieser Technik stellen die heutigen Analogspeicher dar. Es werden noch einige Anstrengungen nötig sein, bevor man über leistungsfähige Schaltkreise verfügen wird.

Adresse des auteurs

Pascal Heim, ing. dipl. EPFL, Dr. *Xavier Arreguit*, ing. dipl. EPFL, Prof. Dr. *Eric Vittoz*, EPFL et CSEM, EPFL-LEG, ELB-Ecublens, 1015 Lausanne et CSEM S.A., Maladière 71, 2007 Neuchâtel.

Le réseau de Kohonen [1] permet de projeter un ensemble de vecteurs appartenant à un espace de dimension n sur son propre espace de dimension m. Le réseau exécute un processus non supervisé d'auto-organisation qui a la propriété de conserver autant que possible les relations topologiques des vecteurs d'entrée. De plus cette organisation attribue une surface plus importante du réseau aux régions de l'espace d'entrée qui sont présentées plus fréquemment. Ces propriétés de projection sont très similaires à celles que l'on trouve dans certaines parties du cerveau.

La simulation sur ordinateur permet d'implémenter pratiquement toute les variantes imaginables du réseau de Kohonen: il n'existe pas de limites dans le choix de l'algorithme ou de la topologie du réseau. En particulier, il est facile de simuler des réseaux à structure évolutive, dans lesquels des neurones sont ajoutés au réseau pendant la phase d'apprentissage. Dans un premier temps, les implémentations en circuits analogiques n'offriront pas une telle souplesse et seront dédiées à des applications particulières. Cependant, leur portabilité et la vitesse inhérente à leur parallélisme total apporteront des avantages décisifs par rapport aux simulations.

En essayant d'exploiter le plus possible les caractéristiques physiques des transistors dans les circuits analogiques, le nombre de composants par fonctions peut être fortement réduit par rapport à une implémentation numérique, ce qui permet d'augmenter le nombre de tâches effectuées sur la même puce. L'approche analogique n'est possible que si une grande précision n'est pas nécessaire, comme par exemple dans les tâches de reconnaissance, pour autant que le réseau travaille de manière collective et continue, ce qui est faisable au moyen de techniques analogiques.

Implémentation analogique

La figure 1 montre le schéma-bloc d'un réseau de Kohonen unidimensionnel ($m = 1$). Le vecteur d'entrée $\boldsymbol{x}$ est distribué parallèlement à tous les neurones. Une rangée de M neurones fournit M signaux de sortie Y_i, fonc-

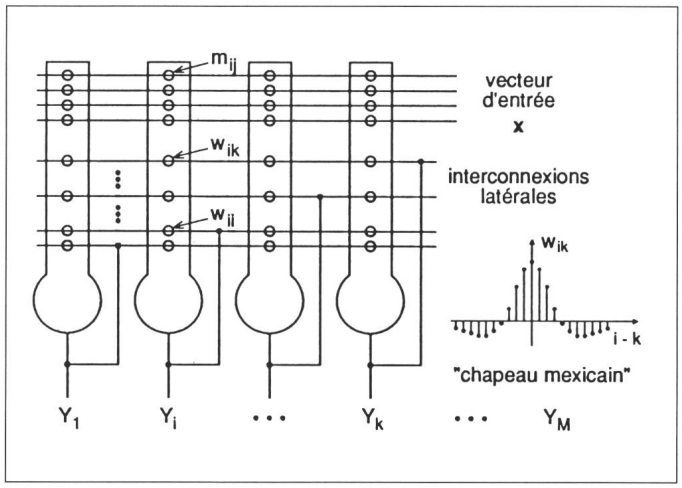

Figure 1
Schéma de principe du réseau de Kohonen

tions du vecteur d'entrée $\boldsymbol{x}$, des poids synaptiques $\boldsymbol{m}_i$ associés à chaque neurone et des contre-réactions internes gouvernant le comportement collectif du réseau. Tous les neurones sont interconnectés entre eux de manière à exciter leur voisinage proche et inhiber le reste du réseau au travers d'une fonction de pondération ayant la forme caractéristique d'un chapeau mexicain (fig. 1). Ces couplages latéraux nécessitent M connexions synaptiques fixes par neurone, c'est-à-dire M^2 interconnexions pour le réseau complet. Il est possible de réaliser cette fonction collective avec beaucoup moins de connexions à l'aide de circuits analogiques simples. L'effet collectif résultant va concentrer l'activité du réseau en une «bulle» localisée à l'endroit où les entrées ont le plus de poids. Le processus d'apprentissage nécessite la formation d'un voisinage qui peut s'apparenter à cette bulle. D'autre part, la plasticité des synapses, c'est-à-dire la vitesse à laquelle elles peuvent s'adapter en réponse à un stimuli, doit être contrôlée lors de l'apprentissage.

L'algorithme d'apprentissage généralement utilisé pour la simulation du réseau est le suivant: partant d'un réseau totalement désordonné, on lui présente successivement des vecteurs d'entrée $\boldsymbol{x}$ choisis au hasard dans une base de données. A chaque itération k, on détermine par une mesure appropriée le neurone «gagnant» c dont le vecteur poids synaptique $\boldsymbol{m}_c$ est le plus proche du vecteur d'entrée. Ensuite, on définit un voisinage Nc autour de ce neurone puis on adapte les poids synaptiques des neurones sélectionnés dans le sens d'un rapprochement avec le vecteur présenté $\boldsymbol{x}$, ce qui, en terme de composantes (indice j), s'exprime par:

$$m_{ij}(k+1) = m_{ij}(k) + \alpha\,(x_j(k) - m_{ij}(k))$$
$$\text{si } i \in N_c(k)$$

$$m_{ij}(k+1) = m_{ij}(k)$$
$$\text{si } i \notin N_c(k) \qquad (1)$$

Au début de l'organisation, le voisinage d'apprentissage recouvre jusqu'à la moitié de la surface du réseau et le gain d'adaptation α est proche de l'unité. A mesure que les vecteurs d'entrée sont présentés, le voisinage diminue progressivement jusqu'à atteindre la taille d'une cellule avec ses plus proches voisins. Simultanément, le gain diminue jusqu'à une valeur de l'ordre de 0,01: le processus entre alors dans une phase de convergence pendant laquelle la surface du réseau tendra à reproduire au mieux la distribution topologique et statistique de la base de données.

La dimension du réseau de Kohonen peut être quelconque, cependant les technologies VLSI actuelles limitent pratiquement les implémentations aux dimensions 1 et 2. En ce qui concerne les implémentations purement analogiques, deux approches différentes ont été étudiées: la première est basée sur la fonction d'excitation-inhibition en forme de chapeau mexicain et la seconde est basée sur un circuit de type Winner-Take-All (WTA) [2] pour la sélection du neurone gagnant et un réseau non-linéaire pour la génération de la bulle.

Réalisation du réseau avec poids fixes

Par définition, le réseau de Kohonen est adaptatif. Cependant, on peut imaginer des applications pour lesquelles on pourrait se contenter de réseaux à poids fixes, préalablement simulés sur ordinateur, tels des ROMs analogiques remplaçant par exemple les tables (look-up tables) utilisées dans le contrôle de certains processus non-linéaires. L'information mémorisée est beaucoup plus riche du fait que les relations topologiques entre les éléments de l'espace d'entrée sont conservés. De plus le centre de gravité de la bulle peut se déplacer de manière localement continue, l'interpolation étant assurée par le réseau d'excitation-inhibition collectif.

La figure 2 montre un circuit implémentant un réseau de Kohonen à poids fixes de 12 × 12 neurones [3], basé sur la fonction d'excitation-inhibition en forme de chapeau mexicain. Afin de réduire le nombre des interconnexions, nous avons utilisé un réseau de résistances-conductances RG [4]. Ce réseau consiste en un maillage orthogonal de résistances R, complété à chaque nœud d'une conductance G à la terre. Chaque neurone est rattaché à un nœud qui définit sa position dans le réseau. Lorsqu'un neurone injecte un courant en un nœud, il produit un gradient de tension décroissant de la forme $\exp(-r/L)$, où $L = (RG)^{-1/2}$ est la longueur caractéristique du réseau et r la distance au point d'injection. Si on soustrait à cette fonction

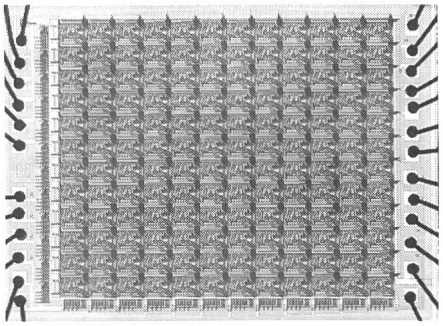

Figure 2 Circuit réalisant un réseau de Kohonen à poids fixes de 12 × 12 neurones
Dimensions 3,2 × 2,4 mm²

une tension constante, on obtient une fonction d'excitation-inhibition dont les propriétés sont très voisines de celles de la fonction chapeau mexicain. Le réseau RG étant linéaire, la même fonction est accessible par tous les neurones par le principe de superposition. Afin de s'affranchir des terminaisons, le réseau RG a été rebouclé dans les deux directions, donnant ainsi au réseau de Kohonen une topologie torique. Cette topologie est structurellement compatible avec des vecteurs d'entrée de dimension supérieure ou égale à 3. La figure 3 montre le résultat pour un des vecteurs d'entrée ($n = 4$). Le réglage de la taille de la bulle nécessite un choix judicieux des différents paramètres (valeurs de R et G, valeur de saturation de l'activité des neurones et poids relatifs des contre-réactions). Par conséquent, cette méthode n'est pas assez souple pour assurer la variation du voisinage

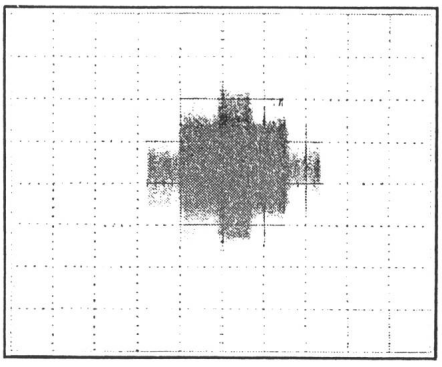

Figure 3 Bulle générée par le circuit de la figure 2 pour un vecteur d'entrée donné

lors de l'apprentissage. En revanche, la densité d'intégration satisfait les exigences de la réalisation de ROMs analogiques: la cellule complète ne mesure que 200 × 180 μm en technologie Sacmos 3 μm et comprend un vecteur synaptique de dimension 4.

Réalisation du réseau avec apprentissage sur puce

Si l'on veut ajouter l'apprentissage sur le chip, il faut disposer de mémoires analogiques et d'un moyen simple d'implémenter la génération de la bulle et l'adaptation des poids synaptiques. La figure 4 montre le schéma-bloc d'une implémentation possible. La génération du voisinage s'effectue à l'aide d'un circuit de type Winner-Take-All pour la sélection du neurone gagnant et d'un réseau non-linéaire [5] dont la structure est la même que celle du réseau *RG* précédemment cité et ne nécessite que trois transistors par cellule. Les avantages de cette solution sont la très grande souplesse de réglage de la taille de la bulle (rapport de 1000 à 1 réalisable) ainsi que sa forme conique (fig. 5) qui

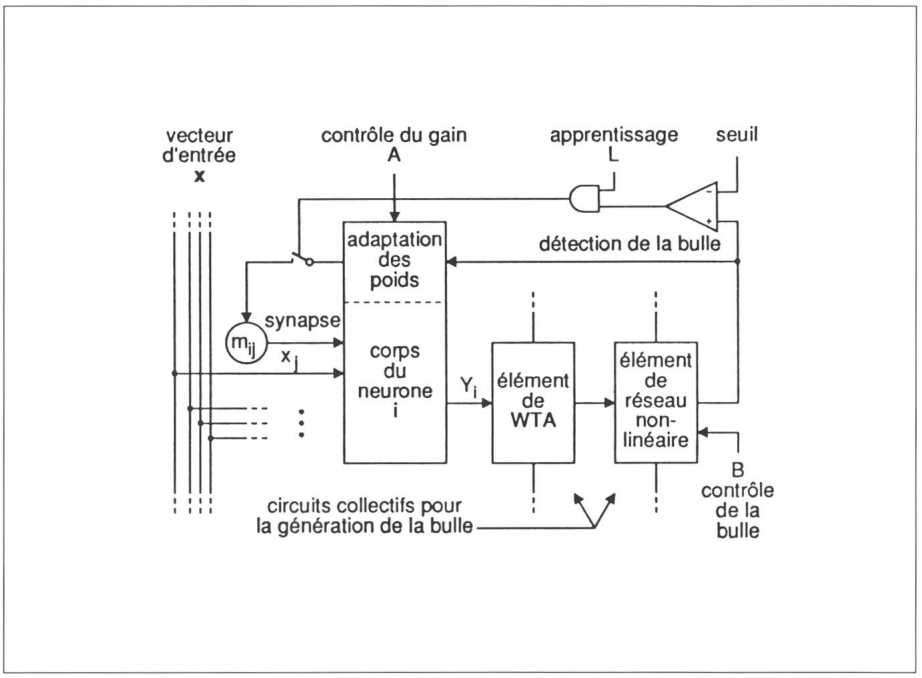

Figure 4 Schéma-bloc d'un neurone avec apprentissage et éléments des circuits collectifs associés

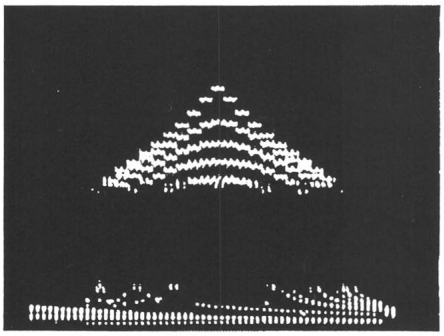

Figure 5 Profil d'une bulle générée par le réseau non-linéaire
5 coupes de la même bulle contenant environ 500 cellules

peut être exploitée pour rendre le gain d'adaptation α fonction du rayon. Des simulations ont montré en effet que si α est maximum au centre de la bulle et décroît avec le rayon, la phase d'organisation est nettement plus courte. La synapse est une mémoire modifiable. Le corps du neurone se charge de contrôler cette modification en tenant compte des paramètres de commande A et B, du profil de la bulle et des grandeurs x_j et m_{ij}. De plus, le corps du neurone effectue la mesure de proximité nécessaire au Winner-Take-All pour sélectionner le gagnant. Finalement, le signal de commande L permet de cadencer l'apprentissage au rythme où sont présentés les vecteurs.

La mémorisation des valeurs analogiques est actuellement le problème le plus difficile auquel nous devons faire face pour implémenter les réseaux de neurones analogiques avec apprentissage sur la puce. Pour effectuer des opérations mathématiques dans les circuits analogiques, on exploite la relation non-linéaire entre la tension de grille des transistors MOS et leur courant de drain (exponentielle en faible inversion, quadratique en forte inversion). Dans la plupart des cas, le transistor est utilisé comme une source de courant commandée ou comme une résistance dont les valeurs respectives dépendent de la tension de grille. Or, comme aucun courant n'est consommé par la grille des transistors MOS, le condensateur est l'élément le plus naturel pour mémoriser une tension dans les circuits et donc pour coder la valeur du poids synaptique. Les technologies CMOS actuelles ont un diélectrique de très bonne qualité et par conséquent le temps de rétention des charges sur les électrodes des condensateurs n'est limité que par les courants de fuite des jonctions des transistors d'accès et de la capacité (typiquement de l'ordre de la ms à la seconde).

De manière à augmenter le temps de rétention, plusieurs implémentations ont été envisagées et testées:

– mémoires à rafraîchissement périodique: la tension aux bornes du condensateur est rafraîchie à l'aide d'un circuit approprié [6; 7] par comparaison périodique avec des niveaux quantifiés prédéfinis. Pour un temps de mémorisation donné, le nombre de niveaux possibles dépend d'une part de la valeur de courant de fuite à compenser (le rafraîchissement doit se produire avant que la tension ne soit dégradée de plus d'un niveau), et d'autre part du niveau de bruit (probabilité non nulle que le bruit $1/f$ soit plus grand à un instant donné que le pas de quantification) et de l'injection de charges. On peut utiliser ces mémoires pour des temps de rétention de l'ordre de quelques minutes à quelques heures.
– mémoires non-volatiles (EEPROM) [7]: une charge électrique est emprisonnée dans la grille flottante d'un transistor MOS. Par effet Fowler-Nordheim ou par effet d'avalanche, on injecte des paquets de charges de valeur contrôlée en utilisant des impulsions de tension de durée et d'amplitude variables, et on peut ainsi incrémenter la valeur de la tension par pas prédéfinis. Un mécanisme similaire permet d'enlever des paquets de charges de la grille flottante. Le temps de rétention de la charge est de plusieurs années. L'inconvénient de cette méthode réside dans la précision avec laquelle on peut fixer la tension mémorisée (relaxation de la valeur) et dans l'utilisation de tensions élevées pour les impulsions (risque de canaux parasites, couplages capacitifs).
– mémoires programmées à la lumière ultraviolette [7]: l'isolant devient légèrement conducteur sous l'effet de la lumière UV. Sa résistance diminue ($R_{eq} = 10^{15}\ \Omega$) et un faible courant a travers l'isolant permet de

charger ou de décharger la capacité de grille. Lorsque l'on coupe la lumière, la charge reste emprisonnée et la valeur de tension correspondante mémorisée. Les inconvénients de cette méthode sont de faire appel à une lumière UV suffisamment puissante pour des temps de programmation raisonnables (quelques dizaines de secondes) et d'autre part de devoir cacher à l'aide d'un écran (deuxième métal par exemple) les parties du circuit qui ne font pas partie du circuit de mémorisation.

Bien que fonctionnelles, ces trois méthodes de mémorisation posent toutes des problèmes auquels le concepteur doit faire face par des techniques de circuits appropriées. La recherche dans ce domaine reste très active.

Effets des imperfections dans les circuits analogiques

Le comportement collectif des réseaux de neurones est sensé les protéger contre certaines imperfections ou même la défaillance totale de certaines de leurs cellules. Si la deuxième affirmation est vraie dans certains cas pour le réseau de Kohonen, il n'en est pas de même pour la première. L'algorithme décrit ci-dessus s'avère très sensible aux erreurs et c'est précisément son aspect collectif qui est problématique. Ces erreurs proviennent surtout de l'implémentation des poids synaptiques. Dans le cas des mémoires à court terme utilisant des condensateurs, les fuites sont inévitables et l'injection de charges due au transistor d'accès altère la valeur mémorisée après chaque modification. L'utilisation de mémoires à rafraîchissement permet de compenser les fuites mais introduit une troncature de la valeur mémorisée à chaque cycle de rafraîchissement (quantification). D'autres imperfections sont par exemple un mauvais choix du neurone gagnant ou une différence du gain d'adaptation α par rapport à $\boldsymbol{x}$ et $\boldsymbol{m}$ (mauvais appariement des éléments). Les conséquences de ces imperfections sont difficiles à analyser, d'autant plus qu'elles dépendent de nombreux paramètres tels que la valeur du gain α lors de la phase de convergence, la taille du réseau, la dimension du vecteur d'entrée et plus problématique encore, la base de donnée. On peut réécrire l'algorithme en y incluant un certain nombre de ces défauts de la manière suivante:

$$m_{ij}(k+1) = \mathrm{Trunc}[m_{ij}(k) + \alpha\,(x_j(k) - (1+\varepsilon_a)m_{ij}(k)) + \varepsilon_f + \varepsilon_i\,]$$

dans la bulle

$$m_{ij}(k+1) = \mathrm{Trunc}[m_{ij}(k) + \varepsilon_f\,]$$

hors de la bulle (2)

Dans cet algorithme altéré, ε_a représente une différence de gain, ε_f une fuite supposée constante, ε_i l'injection de charges supposée constante, et finalement la fonction de troncature si des mémoires rafraîchies sont utilisées. L'effet des défauts est en général visible lorsque le gain n'est plus en mesure de les compenser. Par exemple l'erreur sur le gain ou l'injection de charges peuvent amener le réseau à diverger partiellement en dehors de l'ensemble de définition de la base de donnée. Ces défauts sont actuellement simulés séparément afin d'en connaître qualitativement et quantitativement les effets et concevoir les circuits en conséquence.

Applications et choix du type de vecteur d'entrée

Le réseau de Kohonen s'auto-organise de manière à projeter les vecteurs d'un espace donné sur un plan formé d'un nombre limité de neurones tout en conservant au mieux la topologie de l'espace. Le nombre de classes que l'on peut discriminer est étroitement lié au nombre de neurones implémentés et à la mise en œuvre de l'apprentissage. Une fois l'apprentissage terminé, le réseau peut être utilisé pour associer des vecteurs d'entrée à l'une des classes apprises (position du vecteur gagnant dans le réseau) ou pour caractériser l'espace observé (topologie) en analysant les valeurs des poids synaptiques dans le réseau. Il faut toutefois rester prudent quant à l'interprétation des résultats. En effet, du fait de la projection, deux classes adjacentes sur le réseau ne le sont pas forcément dans l'espace d'origine. La réciproque est aussi vraie.

Le choix du vecteur d'entrée (nombre et type de composantes) est déterminant pour obtenir une bonne classification. Un bon prétraitement de l'information devrait extraire les caractéristiques pertinentes des objets à classifier. Ces caractéristiques peuvent alors être utilisées comme composantes du vecteur d'entrée au réseau de Kohonen. On peut ainsi diminuer la dimension du vecteur d'entrée tout en améliorant les performances de classification du réseau. A la limite, si le nombre de caractéristiques correspond à la dimension de l'espace et qu'elles forment une base orthogonale, le réseau de Kohonen peut être remplacé par un circuit du type Winner-Take-All dans la phase de reconnaissance des données.

Souvent, le problème est de déterminer quelles sont les caractéristiques pertinentes qu'il faut extraire. Par exemple, l'utilisation du réseau de Kohonen a été proposé pour classifier des données spectrales brutes [8] dans un système de reconnaissance de phonèmes. Le vecteur d'entrée n'était composé que de l'information fréquentielle obtenue à partir du résultat d'une FFT sur le signal d'entrée et le taux de reconnaissance obtenu est relativement bas par rapport à celui d'un être humain. De manière générale, les systèmes artificiels de reconnaissance de la parole souffrent d'un manque de ressources de calcul en comparaison avec le système auditif biologique. On peut donc s'inspirer du traitement du signal effectué par la cochlée et par les différentes couches du cerveau pour extraire différentes caractéristiques du son (modulations FM, nombre et type de sources sonores, pitch, perception binaurale [9]). Un gros effort est mis en ce moment à la détermination et à l'extraction de ces caractéristiques.

Un point important à considérer pour l'utilisation du réseau de Kohonen est sa propriété d'auto-organisation non supervisée. La topologie du réseau après organisation (localisation des classes dans le réseau) dépend de l'ordre et de la fréquence avec lesquels on présente les vecteurs d'entrée pendant l'apprentissage. Lorsque l'on veut associer une action à chacune des classes, il est nécessaire d'étiqueter chacune des régions correspondantes du réseau. Ceci ne peut se faire qu'à l'aide d'une deuxième couche avec un mécanisme de supervision. Par exemple, dans l'application du robot mobile [10], le réseau de Kohonen permet d'obtenir une représentation interne de l'environnement (couloir, obstacles, etc.). Une deuxième couche de neurones est utilisée pour effectuer des actions (tourner à droite ou à gauche, avancer). L'association entre le réseau de Kohonen et la deuxième couche est obtenue par un apprentissage supervisé utilisant le principe de récompense-punition.

Conclusion

Actuellement, les efforts consentis dans le domaine des réseaux de neurones sont axés sur les simulations et les circuits digitaux. Les réseaux analogiques ont déjà fait leurs preuves dans des implémentations de prétraitement bas-niveau d'images et de sons (rétines et cochlées artificielles). Dans ce contexte, l'implémentation analogique se révèle imbattable pour la réalisation des fonctions collectives dans les réseaux. La plupart des blocs analogiques nécessaires à l'implémentation VLSI analogique d'un réseau de Kohonen avec apprentissage sur puce ont été développés et intégrés. Néanmoins, avant de pouvoir disposer d'un système complet, les performances des mémoires analogiques devront être améliorées. Les progrès apportés aux implémentations du réseau de Kohonen pourront certainement être appliqués à d'autre types de réseaux.

Bibliographie

[1] *Kohonen T.*: Self-organization and associative memory. Springer-Verlag, Berlin, 1988, pp. 119–157.

[2] *Lazzaro J., Ryckebusch S., Mahowald M. A.* and *Mead C. A.*: Winner-take-all networks of order *N* complexity. Proc. 1988 IEEE Conf. on Neural Information Processing-Natural and Synthetic, Denver, 1988, pp. 703–711.

[3] *Vittoz E. et al.*: Analog VLSI implementation of a Kohonen map. EPFL, Journées d'électronique 1989, pp. 291–301.

[4] *Mead C. A.*: Analog VLSI and neural systems. Addison-Wesley, Reading, 1989, pp. 107–116, 339–351.

[5] *Heim P., Hochet B. Vittoz E.*: Generation of learning neighbourhood in Kohonen feature maps by means of simple nonlinear network. Electronics Letters 27(1991)3, pp. 275–277.

[6] *Hochet B.*: Multivalued MOS memory for variable synapse neural network. Electronics Letters, 25(1989)10

[7] *Vittoz E. et al.*: Analog storage of adjustable synaptic weights. ITG/IEEE Workshop on Microelectronics for Neural Networks, Dortmund, Germany, June 25–26, 1990.

[8] *Kohonen T.*: The «neural» phonetic typewriter. IEEE Computer, March 1988, pp. 11-22.

[9] *Mead C. A., Arreguit X., Lazzaro J.*: Analog VLSI model of Binaural hearing. IEEE Transactions on Neural Networks, 2(1991)2, pp. 230 to 236.

[10] *Sorouchyari E.*: Mobile robot navigation: a neural network approach. EPFL, Journées d'électronique 1989, pp. 159–175.