

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEINEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE BATNA
FACULTE DES SCIENCES DE L'INGENIEUR

MEMOIRE

Présenté au

DEPARTEMENT D'ELECTRONIQUE

Pour l'obtention du diplôme de

MAGISTER EN MICROELECTRONIQUE

Option : IC Design

Par

Souhil KOUDA

Ingénieur, Département d'Electronique-Université de Batna

Intitulé

Conception D'un Capteur D'Humidité Intelligent

Devant le jury constitué de :

Dr. BENHAYA Abdelhamid	M.C. U. Batna	Président
Dr. DIBI Zohir	M.C. U. Batna	Rapporteur
Dr. AYAD Fayçal	M.C. U. Jijel	Examineur
Dr. OUNISSI Abdelhamid	M.C. U. Batna	Examineur
Dr. MAHAMDI Ramdan	M.C. U. Batna	Examineur

Je dédie ce travail à :

Ma mère,

Mon père,

Mes frères et mes sœurs,

Tous mes amis sans exception.

Remerciements

Je souhaiterais en premier lieu adresser mes remerciements à mon encadreur Mr **DIBI Zohir**, directeur de l'institut de l'électronique et Maître de conférence à l'université de Batna, pour l'encadrement actif dont j'ai bénéficié durant cette année. Je lui dois les remerciements les plus sincères.

Je remercie le Professeur **Nour Eddine BOUGUECHAL**, doyen de la faculté des sciences de l'ingénieur à l'université de Batna. Je lui suis très reconnaissant pour ses qualités rares tant au niveau humain que scientifique que j'ai pu entrevoir dans les déplacements que nous avons été amenés à effectuer ensemble.

Je remercie Mr **BENHAYA Abdelhamid**, Maître de conférence à l'université de Batna, qui ma fait l'honneur de s'intéresser à ce travail et de présider son jury.

Mes remerciements vont également à Mr **AIAD Fayçal**, Maître de conférence à l'université de Jijel, Mr **MAHAMDI Ramdan**, Maître de conférence à l'université de Batna, et à Mr **OUNISSI Abdelhamid**, Maître de conférence à l'université de Batna, pour avoir accepté d'être les examinateurs de ce mémoire.

J'adresse mes remerciements à tous mes amis pour leur soutien.

Enfin, j'exprime toute mon amitié à mes camarades grâce à qui je garderai d'excellents souvenirs.

Symbole	Description	Unité
MEMS	Micro Electro Mechanical Systems	
CHS	Capacitive Humidity Sensor	
ANN	Artificial Neural Network	
ANN-model	Modèle à base des réseaux de neurones	
INV-ANN	Modèle inverse à base des réseaux de neurones	
MLP	multilayer Perceptron	
H	Humidité	%RH
RH	Relative Humidity	
S	Sensibilité	nF/%R.H
T	Température	°C
NL	Non Linéarité	
C	Capacité	nF
CVC	Capacity to Voltage Converter	
V_{s1}	Tension de sortie de CVC	mV
V_s	Tension de sortie d'ANN-INV	mV

Sommaire

Introduction générale	5
Chapitre I : Capteur d'humidité	8
Introduction	9
I.1. Capteur d'humidité.....	9
I.1.1. Capteur d'humidité capacitif.....	10
I.1.2. La détermination de l'humidité.....	10
I.1.3. Capacité pour la diffusion dans un corps rectangulaire.....	11
I.2. Capteur d'humidité MEMS	16
I.3. Sensibilité à l'humidité	17
I.4. Effet thermique sur la sensibilité à l'humidité	18
I.5. Etude du modèle analytique	19
I.6. La non linéarité.....	20
Conclusion	21
Chapitre II : Réseaux de neurones et linéarisation	22
Introduction	23
II.1.1. Modèle d'un neurone.....	23
II.1.2. Fonctions de transfert	26
II.1.3. Architecture de réseau	29
II.1.4. Le perceptron multicouches	33
II.1.5. L'apprentissage (supervisé)	33
II.1.6. La rétro-propagation.....	34
II.2.1. Erreurs dans la courbe de transfert de capteur.....	35
II.2.2. Fonction de transfert de capteurs.....	36
II.2.3. Linéarisation	38
II.2.4. Linéarisation de caractéristique de capteur	38
II.2.5. Linéarisation basée sur l'ajustement de courbe « Curve-fitting »	40
Conclusion	41
Chapitre III : Modélisation du capteur d'humidité par les ANNs	42
Introduction	43
III.1. Modélisation du CHS	43
III.1.1. Choix de la base de données.....	43
III.1.2. Création d'une base de validation	44
III.1.3. L'apprentissage du réseau de neurone.....	45
III.1.4. Mesure de la performance du modèle obtenu.....	49
III.2. Modèle ANN du CHS sur PSPICE	51
III.3. Confirmation du comportement capacitif du modèle CHS	54
III.4. Convertisseur Capacité / Tension (CVC).....	55

Conclusion	56
Chapitre IV : Modèle intelligent (INV-ANN)	57
Introduction	58
IV.1. Modélisation du INV-ANN	58
IV.1.1. Linéarisation de la sortie du modèle CHS	58
IV.1.2. Création des bases de données et de tests	59
IV.1.3. L'apprentissage du réseau de neurone	60
IV.1.4. Mesure de la performance du modèle obtenu	61
IV.2. Modèle INV-ANN du CHS sur Pspice	63
IV.3. Modélisation du capteur intelligent	64
IV.4. Test du capteur intelligent	64
IV.5. Test pour un cycle d'humidité	65
IV.6. Test pour un cycle de température	67
Conclusion	67
Conclusion générale	68
Bibliographie	70

Introduction générale

Introduction générale

L'humidité est un paramètre aussi important que la pression ou la température. Elle modifie les caractéristiques électriques des matériaux et agit sur les réponses des systèmes réalisés. Elle peut être exprimée en un certain nombre de manières, et le nombre de méthodes pour la mesurer est encore plus grand. La surveillance et le contrôle de l'humidité paraît d'une grande importance pour la fiabilité des dispositifs.

Les trois types de capteurs d'humidité les plus utilisés sont:

- Capteur d'humidité capacitif
- Capteur d'humidité résistif
- Capteur d'humidité à conductivité thermique

Le capteur utilisé dans ce travail est de type **MEMS** « **M**icro **E**lectro **M**echanical **S**ystems» micro électro mécaniques systèmes. Nos recherches se sont orientées vers un capteur de type capacitif puisque ce type de capteur peut donner de meilleurs résultats tout en restant d'une relative simplicité. Les premiers fabricants de capteurs d'humidité de type capacitif **CHS** « **C**apacitive **H**umidity **S**ensor » utilisaient comme diélectrique des matériaux poreux. Le but recherché était d'avoir une capacité importante car l'électronique d'alors ne permettait pas de travailler avec des capteurs de faible sensibilité.

Les capteurs classiques montrent une caractéristique de transfert non linéaire. De plus l'influence de la température sur le modèle physique est notable dans un milieu de travail assez complexe. Pour ce fait on a adopté les réseaux de neurone comme méthode générale de linéarisation à fin d'obtenir un capteur intelligent. Ce principe, qui tend à se développer, fait appel au concept de capteur 'Intelligent'.

Il s'agit dans ce mémoire d'associer au capteur une structure électronique permettant d'effectuer des corrections, afin d'obtenir une information précise sous forme numérique, donc directement exploitable et pouvant être par exemple facilement raccordé à un réseau industriel.

L'environnement Matlab est utilisé pour reproduire les résultats expérimentaux du CHS, la programmation sur Matlab permet d'ajuster les courbes de réponse du CHS et de créer une base de données sur notre modèle et une autre pour la vérification. L'apprentissage du réseau de neurone est également un programme écrit en Matlab qui est élaboré pour obtenir un réseau de neurone similaire au CHS.

Le simulateur PSPICE est utilisé pour concevoir un modèle ANN simulé à partir du modèle mathématique obtenu par Matlab. Un réseau de boîtes ABM de la bibliothèque de PSPICE nous produit un réseau de neurone semblable au CHS. Puisque notre capteur est de type capacitif, le modèle obtenu par les ABM traduit la variation de l'humidité par une variation d'une capacité, cette dernière est une grandeur passive, qu'il est nécessaire de convertir à une grandeur active. C'est le rôle du circuit convertisseur capacité tension **CVC** « **Capacity to Voltage Converter** » réalisé. Des simulations, à l'aide de PSPICE, sont effectuées pour confirmer la reproduction fidèle du comportement de CHS.

Une linéarisation, par un programme Matlab, est appliquée à la réponse du CHS dont le but est de créer une base de données d'un élément de correction, permettent de corriger la réponse non linéaire du CHS. L'apprentissage pour cette nouvelle base de données nous donne le modèle inverse INV-ANN. Notons que ce composant est totalement différent de l'ANN qui reproduit le comportement du CHS par contre le INV-ANN joue le rôle d'un correcteur de réponse. Les trois blocs, le modèle ANN, le convertisseur capacité tension CVC et le modèle inverse INV-ANN constituent le capteur d'humidité intelligent.

L'objectif de ce travail étant la mesure précise de l'humidité relative de l'air dans un milieu complexe. Nous avons résumé dans ce mémoire l'essentiel des travaux réalisés.

Après une introduction montrant l'importance des capteurs dans une chaîne de mesure et l'intérêt de la mesure du taux d'humidité, on aborde la relation entre la température et l'humidité absolue ou relative de l'air.

Dans le chapitre 1, on s'intéressera à la méthode de mesure de l'humidité ensuite une présentation du capteur MEMS capacitif sera exposé dans ce travail.

Le chapitre II est consacré à la présentation du réseau de neurone. Ensuite, on décrira quelque type d'erreurs de la réponse des capteurs ainsi que les méthodes de linéarisation de ces erreurs.

Le chapitre III décrit le modèle ANN adopté pour le capteur utilisé ainsi que les simulations effectuées par Orcad PSPICE.

Enfin, le dernier chapitre présentera les différentes étapes de la conception du modèle inverse INV-ANN et son implantation sur le simulateur PSPICE ainsi que les différentes simulations effectués.

Chapitre I : Capteur
d'humidité

Introduction

Qu'on le désire ou non, l'eau et la vapeur d'eau coexistent ensemble. En raison de la distribution asymétrique de leur charge électrique, des molécules d'eau sont facilement absorbées sur presque n'importe quelle surface, où elles sont présentées comme couche mono ou multimoléculaire. La vapeur d'eau en air ou n'importe quel autre gaz s'appelle généralement l'humidité ; en liquides et solides, elle est habituellement indiquée comme humidité. La détermination de l'humidité, comme dans la prévision des inondations, brouillard, conditions pour l'apparence des maladies des plantes, etc., est de grande importance économique. Les produits alimentaires ou les matières premières stockées peuvent sécher à basse humidité ou devenir moisissés à l'humidité élevée. Dans beaucoup de processus industriels, la mesure de l'humidité est importante pour l'entretien des conditions optimal à la fabrication. L'humidité peut être exprimée en un certain nombre de manières, et le nombre de méthodes pour les mesurer est encore plus grand. Un ingénieur dont le souci principal doit éviter la condensation n'importe où à son système doit être intéressé par le point de condensation de l'écoulement de gaz. Un chimiste peut être intéressé par la seule quantité de vapeur d'eau, tandis que dans une imprimerie ou une chambre de stockage, l'humidité relative est plus qu'importante [1].

I.1. Capteurs d'humidité

Le besoin de protection de l'environnement a mené à l'expansion dans le développement de capteur. Les capteurs d'humidité ont attiré l'attention dans les domaines industriels et médicaux [2]. La mesure et la commande de l'humidité est importante dans beaucoup de secteurs comprenant l'industrie (papier, électroniques), l'environnement domestique (climatisation), la médecine (équipement respiratoire), etc.... Différentes méthodes sont utilisées pour les mesures d'humidité, par exemple, changements des propriétés mécaniques, optiques, et électriques des mélanges de vapeur d'eau de gaz [3].

Trois types de capteurs d'humidité existent :

- Capteur d'humidité capacitif,
- Capteur d'humidité résistif,
- Capteur à conductivité thermique.

I.1.1. Capteur d'humidité capacitif

Des capteurs capacitifs d'humidité relative sont largement répandus dans des applications industrielles, commerciales, et la télémétrie de météorologie. Ils se composent d'un substrat sur lequel une couche mince d'oxyde de polymère ou de métal est déposée entre deux électrodes conductrices. La surface de sensation est enduite d'une électrode poreuse en métal pour la protéger contre la contamination et l'exposition à la condensation. Le substrat est en général fabriqué de verre, en céramique, ou silicium. Le changement par accroissement de la constante diélectrique d'un capteur d'humidité capacitif est presque directement proportionnel à l'humidité relative (RH) de l'environnement. Le changement de la capacité est en général de 0.2 à 0.5 pF pour un changement de 1% RH, alors que la capacité de bulk est entre 100 et 500 pF à RH 50% et 25°C. Les capteurs capacitifs sont caractérisés par un coefficient de basse température, capacité de fonctionner à des températures élevées (jusqu'à 200°C), plein rétablissement de condensation, et une résistance raisonnable aux vapeurs chimiques. Le temps de réponse s'étend de 30 à 60 s pour un pas de changement de 63% RH [4].

Les techniques de l'état de l'art pour produire le capteur capacitif tirent profit de plusieurs principes employés à la fabrication de semi-conducteur pour rapporter des sondes avec la dérive et l'hystérésis minimales [5].

I.1.2. La détermination de l'humidité

La détermination de l'humidité est basée sur la quantité de vapeur d'eau par unité de la masse de l'atmosphère. Comme tous les gaz dans l'atmosphère, la vapeur d'eau constitue une partie finie de toute la pression atmosphérique. Cette pression partielle de vapeur d'eau est proportionnelle au contenu d'humidité atmosphérique et fournit ainsi une mesure de la quantité absolue d'humidité dans l'air. Si un échantillon d'air est saturé de molécules d'eau à une température donnée, il atteint par la suite un état d'équilibre dans lequel le taux des molécules d'eau sortants du liquide est le même que le taux des entrants au liquide. En conséquence, le contenu de vapeur d'eau dans l'air et la pression de vapeur d'eau devient constant. La pression de vapeur dans cet état s'appelle la pression de vapeur d'eau de saturation et elle augmente avec l'augmentation de la température [6].

L'humidité absolue est définie comme un rapport de la masse de vapeur d'eau par unité de volume d'air, qui peut être exprimée comme :

$$\text{Humidité absolue} \left(\frac{\text{g}}{\text{m}^3} \right) = \frac{\text{la masse de la vapeur d'eau}}{\text{volume d'air}} \quad (\text{I-1})$$

L'humidité absolue change avec l'expansion de volume d'air de sorte qu'elle ne donne pas une représentation fiable de l'humidité globale dans l'air. L'humidité spécifique est le rapport de la masse de vapeur d'eau par unité de la masse de mélange d'air-eau-vapeur.

$$\text{Humidité spécifique} = \frac{\text{la masse de la vapeur d'eau}}{\text{unité masse de mélange d'air-eau-vapeur}} \quad (\text{I-2})$$

Le rapport de mélange est défini comme la masse de vapeur d'eau par unité de la masse d'air sec :

$$\text{rapport de mélange} \left(\frac{\text{g}}{\text{m}^3} \right) = \frac{\text{la masse de la vapeur d'eau}}{\text{masse d'air sec}} \quad (\text{I-3})$$

L'humidité relative est le rapport du contenu de vapeur d'eau dans l'air à la quantité maximale de vapeur d'eau que l'air peut maintenir à une température donnée :

$$\text{Humidité relative} = \frac{\text{quantité de vapeur d'eau dans l'air}}{\text{quantité de vapeur d'eau que l'air peut tenir}} \quad (\text{I-4})$$

Puisque la pression de vapeur de saturation est une fonction de la température, l'humidité relative change non seulement avec la quantité de vapeur d'eau dans l'air, mais également avec la température. L'humidité relative est importante parce que c'est un paramètre sans dimensions et il est associé à la sécheresse du matériel.

I.1.3. Capacité pour la diffusion dans un corps rectangulaire

La capacité transitoire a été dérivée pour la diffusion de quatre-côtés dans un corps rectangulaire. Le schéma I.1 illustre la diffusion d'humidité dans un corps rectangulaire, qui a une longueur de $2a$, une largeur de $2b$, et une hauteur de d .

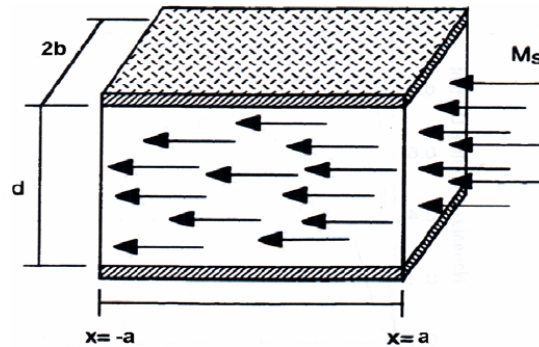


Fig.I.1 - La géométrie d'un solide rectangulaire où la diffusion dans le corps a lieu des quatre-côtés. La concentration en humidité sur toutes les surfaces est fixe à la M_s

La théorie mathématique de diffusion dans les substances isotropes est basée sur l'hypothèse que le taux de transfert d'une substance à travers l'unité surface d'une section est proportionnel à la normale mesurée par le gradient de concentration à la section, c.-à-d.

$$F = -D \frac{\partial M}{\partial x}, \quad (I-5)$$

Où F est le taux de transfert par unité de surface de la section, M est la concentration de la substance diffusée, x est la normale mesurée par coordonnée de l'espace à la section, et D s'appelle le coefficient de diffusion. L'équation fondamentale de la diffusion dans un milieu isotrope est la dérivée d'Eq (I-5).

Considérer un élément du volume sous forme parallélépipède dans lequel ces côtés sont parallèles aux axes des coordonnées et ces longueurs sont $2dx$, $2dy$, $2dz$. Soit le centre de l'élément est à $P(x, y, z)$, où la concentration diffusée sur la substance est M . soit $ABCD$ et $A'B'C'D'$ sont les faces perpendiculaires à l'axe des x comme dans la figure I.2.

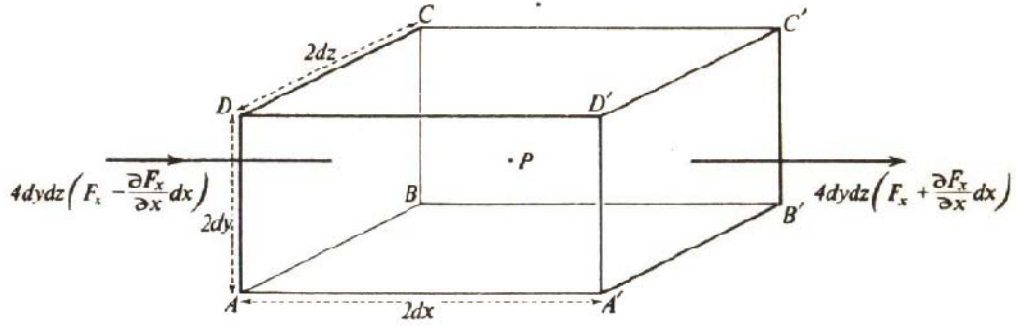


Fig.I.2 - Élément de volume [7]

Alors le taux auquel la diffusion de la substance entre l'élément par la face ABCD dans le plan x dx est donné par :

$$4dydz \left(Fx - \frac{\partial Fx}{\partial x} dx \right) \quad (I-6)$$

Où Fx est le taux de transfert par l'unité de superficie du plan correspondant par P. De façon similaire, le taux de perte de diffusion par A'B'C'D' est donné par

$$4dydz \left(Fx + \frac{\partial Fx}{\partial x} dx \right) \quad (I-7)$$

La contribution du taux de diffusion de la substance dans l'élément de ces deux faces est ainsi égale à :

$$-8dx dy dz \left(\frac{\partial Fx}{\partial x} \right) \quad (I-8)$$

Après le même procédé, à partir des autres faces nous obtenons

$$-8dx dy dz \left(\frac{\partial Fy}{\partial x} \right) \text{ et } -8dx dy dz \left(\frac{\partial Fz}{\partial x} \right) \quad (I-9)$$

Mais le taux auquel la quantité de diffusion de la substance dans l'élément augmente est donné près :

$$-8dx dy dz \left(\frac{\partial M}{\partial t} \right) \quad (I-10)$$

Par conséquent nous avons :

$$\frac{\partial M}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial x} + \frac{\partial F_z}{\partial x} = 0 \quad (\text{I-11})$$

Si le coefficient de diffusion est constant, F_x , F_y , F_z sont donnés par (I-5), et (I-11) devient

$$\frac{\partial M}{\partial t} = D \left(\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2} + \frac{\partial^2 M}{\partial z^2} \right) \quad (\text{I-12})$$

L'équation I.12 peut être résolue en indiquant les conditions aux limites et des états initiaux pour la concentration. Les surfaces supérieures servent d'électrodes à l'extraction de la capacité.

Puis, il n'y a aucun gradient de concentration le long de la direction Z.

$$\frac{\partial M}{\partial t} = D \left(\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2} \right) \quad (\text{I-13})$$

Premièrement, on assume que la condition initial est une valeur constante (c.-à-d., au temps initial, il y a une concentration constante en vapeur d'eau dans la couche sensible). Puisque la couche sensible est abruptement exposée à un environnement humide jusqu'à ce qu'elle atteigne l'état d'équilibre, on assume que les conditions de limite sont en correspondance constante à l'humidité relative spécifique de l'environnement. Ainsi, la solution pour cette équation peut être séparée en deux parties,

$$M = M_1 + M_2 \quad (\text{I-14})$$

Où M_1 et M_2 sont les solutions selon les deux équations ci-dessous dans la limite de champ spécifique et les conditions initiales [7]:

$$\frac{\partial M_1}{\partial t} = D \left(\frac{\partial^2 M_1}{\partial x^2} + \frac{\partial^2 M_1}{\partial y^2} \right) \quad (\text{I-15})$$

$$M_1(x, y, 0) = M_0 \quad (\text{I-16})$$

$$M_1(x, y, t) = 0 \quad (\text{I-17})$$

Et:

$$\frac{\partial M_2}{\partial t} = D \left(\frac{\partial^2 M_2}{\partial x^2} + \frac{\partial^2 M_2}{\partial y^2} \right) \quad (\text{I-18})$$

$$M_2(x, y, 0) = 0 \quad (\text{I-19})$$

$$M_2(x, y, t) = M_s \quad (\text{I-20})$$

Les solutions aux équations (I-15) et (I-18) sont [7] :

$$M_1(x, y, t) = M_0^2 \frac{16}{\pi^2} \left[\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n} \exp(-\alpha_{m,n} t)}{(2n+1)(2m+1)} \times \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right) \right] \quad (\text{I-21})$$

Et

$$M_2(x, y, t) = M_s - M_s \frac{16}{\pi^2} \left[\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n} \exp(-\alpha_{m,n} t)}{(2n+1)(2m+1)} \times \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right) \right] \quad (\text{I-22})$$

Respectivement, où

$$\alpha_{m,n} = \frac{D\pi^2}{4} \left[\frac{(2m+1)^2}{a^2} + \frac{(2n+1)^2}{b^2} \right] \quad (\text{I-23})$$

Puisque la solution d'Eq. (I-13) est la superposition des solutions des Eqs (I-15) et (I-18), donc

$$M(x, y, t) = M_s + (M_0^2) \frac{16}{\pi^2} \left[\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n} \exp(-\alpha_{m,n} t)}{(2n+1)(2m+1)} \times \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right) \right] \quad (\text{I-24})$$

La constante diélectrique ϵ_r change linéairement avec l'absorption d'humidité de sorte qu'on peut l'écrire comme :

$$\epsilon_r(x, y, t) = K_1 M(x, y, t) + K_2 \quad (\text{I-25})$$

Où K_1 et K_2 sont des constantes déterminées par la vapeur liquide spécifique sous la condensation. La capacité du corps, C , est calculée par :

$$C = \int_{-b}^b \int_{-a}^a \frac{\epsilon_0 \epsilon_r(x, y, t)}{d} dx dy \quad (\text{I-26})$$

où a , b , d , ε_0 , et ε_r sont la demi longueur, demi largeur, épaisseur, constante diélectrique, et la constante diélectrique relative, respectivement. Par la substitution d'Eq. (I-25) dans Eq. (I-26) et la simplification, la capacité obtenue est :

$$C = \frac{4\varepsilon_0 K_2 ab}{d} + \frac{\varepsilon_0 K_1}{d} \int_{-b}^b \int_{-a}^a M(x, y, t) dx dy \quad (I-27)$$

Où

$$C = \frac{4\varepsilon_0 K_2 ab}{d} + \frac{\varepsilon_0 K_1}{d} \int_{-b}^b \int_{-a}^a M_s + (M_0^2 - M_s) \frac{16}{\pi^2} \times \left[\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n} \exp(-\alpha_{m,n} t)}{(2n+1)(2m+1)} \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right) \right] dx dy \quad (I-28)$$

Afin de simplifier, la capacité est normalisée avec la considération de la valeur finale de capacité d'état d'équilibre, c-à-d :

$$C_{nor} = \frac{C - C_0}{C_f - C_0} \quad (I-29)$$

Où C_0 , et C_f sont les capacités initiale et finale, respectivement. Par substitution d'Eq (I-27) dans Eq (I-28), l'équation finale pour C_{nor} est donnée par :

$$C_{nor} = \frac{\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n}}{(2n+1)(2m+1)} \int_{-a}^a \int_{-b}^b \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right) - \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n}}{(2n+1)(2m+1)} \exp(-\alpha_{m,n} t) \int_{-a}^a \int_{-b}^b \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right)}{\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{(-1)^{m+n}}{(2n+1)(2m+1)} \int_{-a}^a \int_{-b}^b \cos\left(\frac{(2m+1)\pi x}{2a}\right) \cos\left(\frac{(2n+1)\pi y}{2b}\right)} \quad (I-30)$$

I.2. Capteur d'humidité MEMS

La technologie MEMS et les techniques de microusinage ont eu des impacts important dans la miniaturisation des capteurs. La nouvelle instrumentation utilisée actuellement a été développée à base de ces technologies. Beaucoup d'études ont indiquée le développement et l'application des technologies MEMS à la fabrication des capteurs d'humidité a micro échelle, y compris le type résistif, le type capacitif, le type hygrométrique, le type gravimétrique, et le type optique, une face du capteur d'humidité capacitif est couverte par une couche absorbante d'eau. Il a montré une sensibilité, une linéarité et une stabilité élevées. En conditions humides, l'eau est absorbée par cette couche ce qui cause un changement de la masse d'une électrode mobile. L'humidité peut alors être déterminée. Les propriétés électriques et physiques « la masse » de

substrat fixe et de l'électrode changent pendant que la température change, et c'est ces changements, qui influent sur la valeur de l'humidité relative [8].

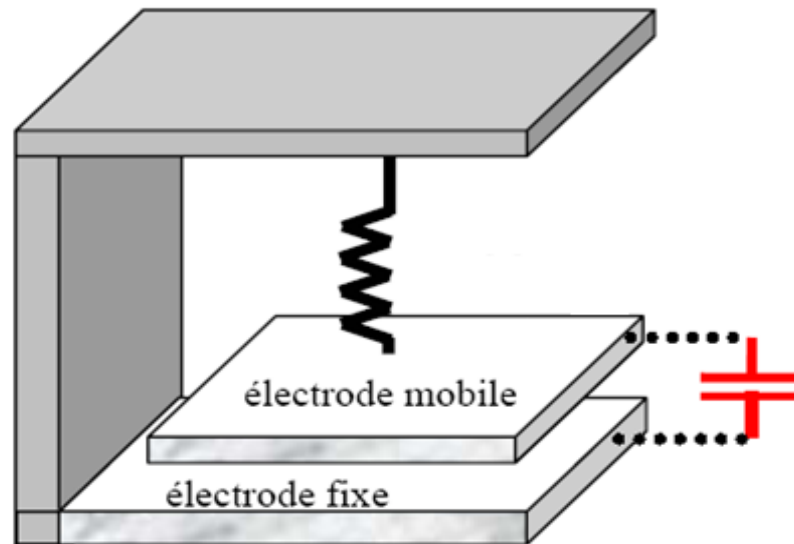


Fig.I.3 - Représentation du capteur d'humidité MEMS

Le capteur d'humidité capacitif comporte une électrode mobile couverte par une matière absorbante d'eau une fois soumis aux conditions humides. L'absorption des molécules d'eau fait augmenter la masse de la couche supérieure de l'électrode mobile, et ceci induit un effort sur cette dernière. En conséquence, un changement de la capacité entre la structure mobile et le substrat est provoqué par le déplacement de l'électrode mobile [6]. La figure-I.3 illustre le fonctionnement du capteur MEMS à détection capacitif. Le « corps d'épreuve » (terme désignant la partie du capteur sensible à l'humidité, le terme « masse sismique » est aussi employé) est une électrode mobile. Une capacité est formée entre le corps d'épreuve et une partie fixe du système. La détection consiste à évaluer les variations de capacité lorsque le corps d'épreuve s'éloigne ou se rapproche de l'élément fixe [9].

I.3. Sensibilité à l'humidité

Afin de déterminer la sensibilité à l'humidité du capteur d'humidité, la capacité a été relevée à mesure que l'humidité était augmentée de 45 %RH à 95 %RH avec un pas de 2.5 %RH aux températures constantes de 25°C, de 50°C et de 75°C. Les résultats correspondants sont montrés

par la figure I.4. La sensibilité d'humidité est 2.03 nF/%RH à 25°C, 3.76 nF/%RH à 50°C et 7.74 nF/%RH à 75°C. On note également que la sensibilité à l'humidité augmente pendant que la température s'élève [8].

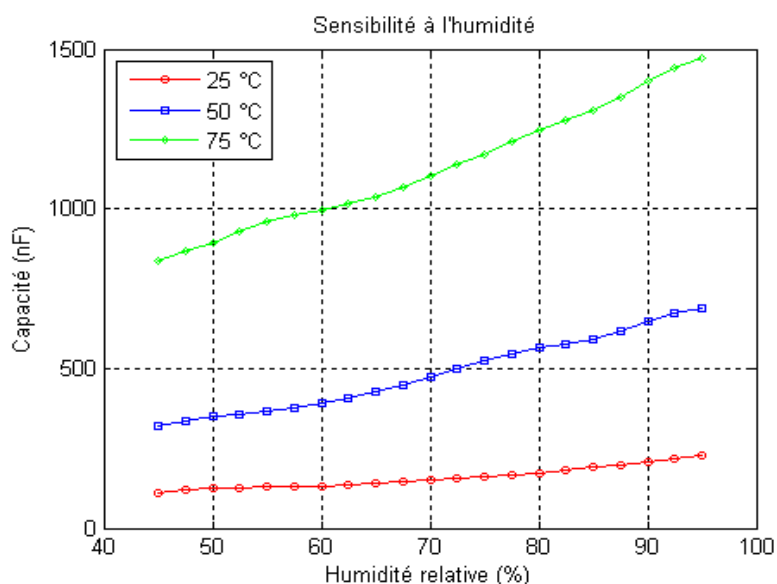


Fig I.4 - Sensibilité à l'humidité à 25°C 50°C et 75°C [8]

I.4. Effet thermique sur la sensibilité d'humidité

La figure-I.5 montre le rapport entre la sensibilité à l'humidité et la température pour un capteur d'humidité. La sensibilité à capteur augmente avec la température selon une relation du deuxième ordre. Une identification de la réponse montre une variation en fonction de la température telle que [8]:

$$S = 0.0016 T^2 - 0.00557 T + 2.3729 \quad (\text{I-31})$$

Où S est la sensibilité (nF/%R.H.) et T est la température. L'Eq (I-31) est utile lors de la compensation de la dérive du signal du capteur d'humidité due à la variation de la température.

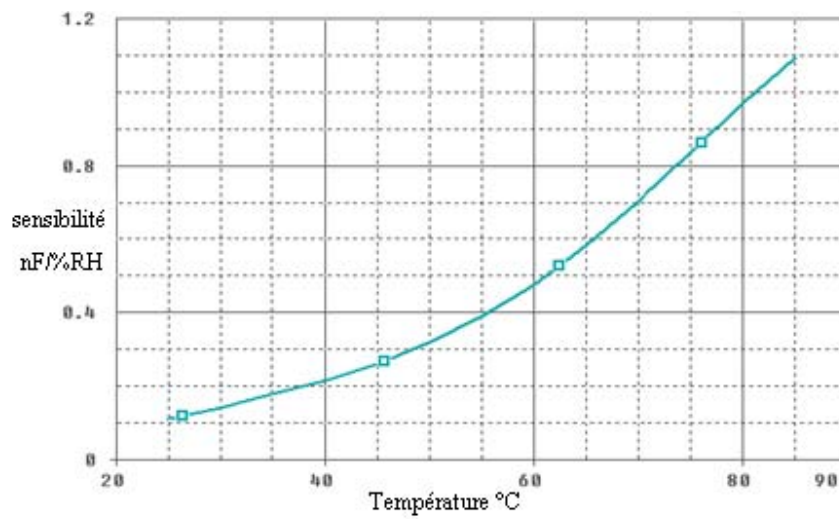


Fig I.5 - Effet de température sur la sensibilité de l'humidité du capteur d'humidité [8]

I.5. Etude du modèle analytique

En examinant les résultats expérimentaux du CHS figure I.5, sa réponse peut être modélisée par une droite à laquelle on ajoute un terme de non linéarité (figure I.6).

Donc la réponse du CHS peut être mise sous la forme :

$$C(H, T) = C_0(T) + S(T) \cdot H + NL(H, T) \quad (I.32)$$

Où :

- $NL(H, T)$: exprime le terme de non linéarité qui dépend de l'humidité appliquée et de la température.
- $C_0(T)$: Capacité d'offset qui dépend de la température.
- $S(T)$: Sensibilité à l'humidité du CHS et qui dépend de la température.

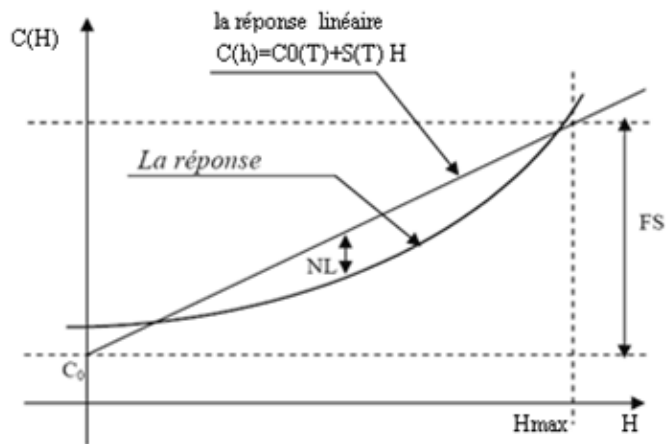


Figure I.6 - Définition graphique du modèle analytique

I.6. La non linéarité

La non linéarité de la réponse à l'humidité à une température constante est calculée en faisant la différence entre les points de mesure $C_m(H)$ et la droite des moindres carrés de ses mêmes points $C_L(H)$. Cette différence est exprimée en pourcentage par rapport à la réponse pleine échelle (FS) [10].

$$NL(\%FS) = 100 \frac{C_m - C_L}{FS} \tag{I-33}$$

La figure I.7 illustre la variation de non linéarité à température ambiante pour un cycle d'humidité de 45% à 95%.

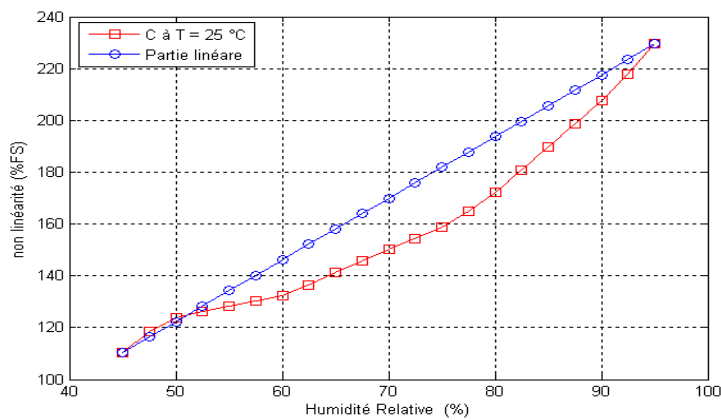


Fig I.7 - La non linéarité en fonction d'un cycle d'humidité à température ambiante

En pratique la température a une faible implication sur la non linéarité par rapport à l'humidité, ce qui est traduit par (I.34) :

$$\left. \frac{\Delta NL}{\Delta T} \right|_{H=cst} \ll \left. \frac{\Delta NL}{\Delta H} \right|_{T=cst} \quad (I.34)$$

La non linéarité se réduit à une fonction de l'humidité, nous avons alors modélisé la non linéarité par une fonction polynomiale d'ordre 2, donné par l'expression (I.35).

$$NL(H) = 0.04 H^2 - 5.6 H + 174 \quad (I.35)$$

Conclusion

La non linéarité de notre modèle analytique est due à l'influence de la température sur la variation de leur capacité de sortie, qui devient une fonction complexe multi variable de l'humidité « H » et de la température « T ». Cependant, l'utilisation de ce capteur dans un milieu dynamique, montre son incapacité à suivre la variation de l'humidité.

Ce modèle analytique pourra néanmoins être amélioré, afin d'obtenir un capteur intelligent, dont le but est d'éliminer la non linéarité et l'effet de la température. Développer un capteur avec des caractéristiques « faible hystérésis, bonne linéarité, sensibilité élevée et un temps de réponse rapide » est notre défi. Nous allons nous baser sur les réseaux de neurones artificiels (ANNs) et les résultats expérimentaux pour modéliser puis corriger toutes ces imperfections.

Chapitre II : Réseaux de
neurones et linéarisation

Introduction

Dans la première partie de ce chapitre, nous présentons le modèle mathématique que nous emploierons dans les chapitres suivants pour décrire, d'une part, un neurone artificiel et, d'autre part, un réseau de neurones complet, c'est-à-dire un ensemble de neurones reliés en réseau. Le modèle que nous présentons dans ce chapitre est celui de base, commun à beaucoup d'architectures. Il n'est cependant pas universel, nous présenterons dans les chapitres subséquents les différentes variantes au fur et à mesure qu'il sera nécessaire de le faire. Pour avoir une idée sur les différents inconvénients des capteurs, la deuxième partie de ce chapitre est consacrée au type d'erreur et de linéarisation des réponses des capteurs.

II.1.1. Modèle d'un neurone

Le neurone artificiel (ou cellule) est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones appartenant à un niveau situé en amont (on parlera de neurones "amonts"). A chacune des entrées est associé un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones appartenant à un niveau situé en aval (on parlera de neurones "avals"). A chaque connexion est associée un poids [11]. (figure II.1)

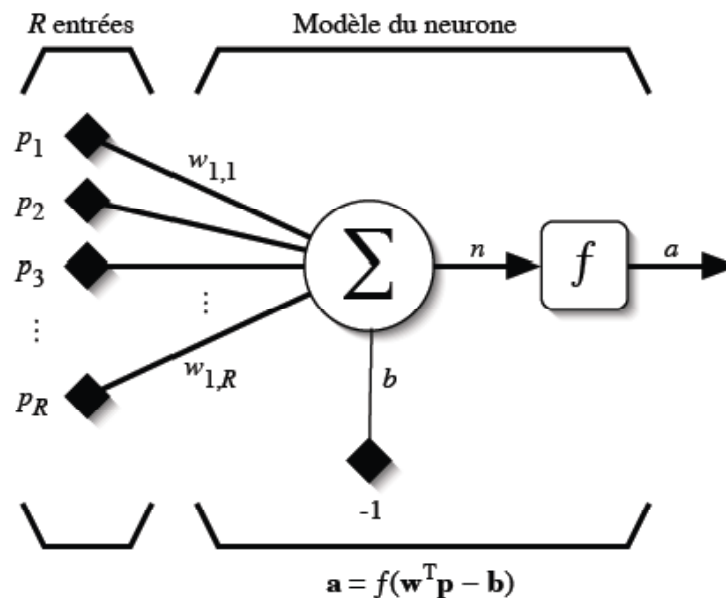


Fig. II.1 - Modèle d'un neurone artificiel

En suivant les notations présentées à la section précédente, les R entrées du neurone correspondent au vecteur $p = [p_1 p_2 \dots p_R]^T$, alors que $\omega = w [\omega_{1,1} \omega_{1,2} \dots \omega_{1,R}]^T$ représente le vecteur des poids du neurone. La sortie n de l'intégrateur est donnée par l'équation suivante :

$$n = \sum_{j=1}^R \omega_{1,j} p_j - b = \omega_{1,1} p_1 + \omega_{1,2} p_2 + \dots + \omega_{1,R} p_{1R} - b, \quad (\text{II-1})$$

Que l'on peut aussi écrire sous forme matricielle :

$$n = W^T p - b \quad (\text{II-2})$$

Cette sortie correspond à une somme pondérée des poids et des entrées moins ce qu'on nomme le biais b du neurone. Le résultat n de la somme pondérée s'appelle le niveau d'activation du neurone. Le biais b s'appelle aussi le seuil d'activation du neurone. Lorsque le niveau d'activation atteint ou dépasse le seuil b , alors l'argument de f devient positif (ou nul). Sinon, il est négatif.

On peut faire un parallèle entre ce modèle mathématique et certaines informations que l'on connaît (ou que l'on croit connaître) à propos du neurone biologique. Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone.

Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie [11]. (figure II.2)

La structure d'un neurone se compose de trois parties :

- La somma : ou cellule d'activité nerveuse, au centre du neurone.
- L'axone : attaché au somma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.
- Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones.

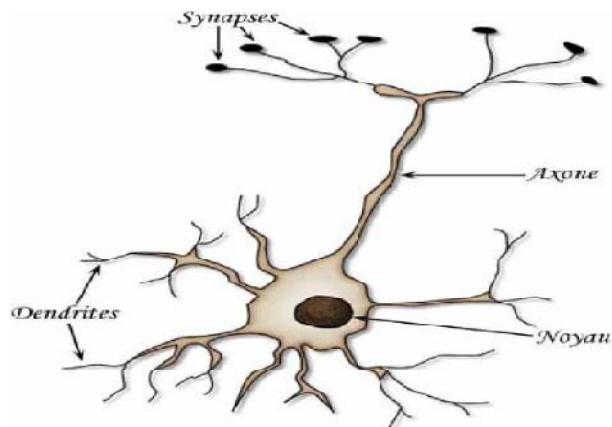


Fig. II.2 - Schéma d'un neurone biologique

Un poids d'un neurone artificiel représente donc l'efficacité d'une connexion synaptique. Un poids négatif vient inhiber une entrée, alors qu'un poids positif vient l'accentuer. Il importe de retenir que ceci est une grossière approximation d'une véritable synapse qui résulte en fait d'un processus chimique très complexe et dépendant de nombreux facteurs extérieurs encore mal connus. Il faut bien comprendre que notre neurone artificiel est un modèle pragmatique qui, comme nous le verrons plus loin, nous permettra d'accomplir des tâches intéressantes. La vraisemblance biologique de ce modèle ne nous importe peu. Ce qui compte est le résultat que ce modèle nous permettra d'atteindre [12].

Un autre facteur limitatif dans le modèle que nous nous sommes donnés concerne son caractère discret. En effet, pour pouvoir simuler un réseau de neurones, nous allons rendre le temps discret dans nos équations. Autrement dit, nous allons supposer que tous les neurones sont synchrones, c'est-à-dire qu'à chaque temps t , ils vont simultanément calculer leur somme pondérée et produire une sortie $a(t) = f(n(t))$. Dans les réseaux biologiques, tous les neurones sont en fait asynchrones.

Revenons donc à notre modèle tel que formulé par l'équation II.2 et ajoutons la fonction d'activation f pour obtenir la sortie du neurone :

$$a = f(n) = f(\mathbf{w}^T \mathbf{p} - b). \quad (\text{II-3})$$

En remplaçant \mathbf{w}^T par une matrice $\mathbf{W} = \mathbf{w}^T$ d'une seule ligne, on obtient une forme générale que nous adopterons tout au long de cet ouvrage :

$$a = f(\mathbf{W}\mathbf{p} - b) \quad (\text{II-4})$$

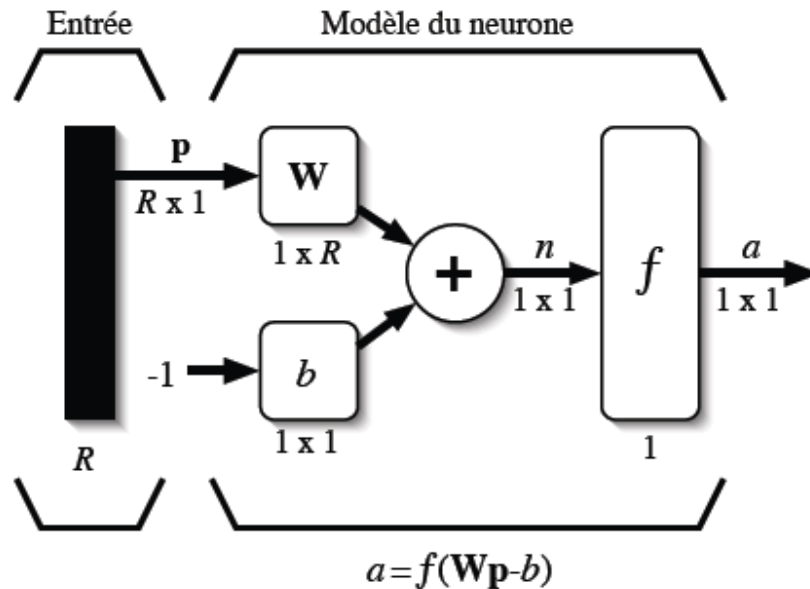


Fig. II.3 - Représentation matricielle du modèle d'un neurone artificiel

L'équation II.4 nous amène à introduire un schéma de notre modèle plus compact que celui de la figure II.1. La figure II.3 illustre celui-ci. On y représente les R entrées comme un rectangle noir (le nombre d'entrées est indiqué sous le rectangle). De ce rectangle sort le vecteur \mathbf{p} dont la dimension matricielle est $R \times 1$. Ce vecteur est multiplié par une matrice \mathbf{W} qui contient les poids (synaptiques) des neurones. Dans le cas d'un neurone simple, cette matrice possède la dimension $1 \times R$. Le résultat de la multiplication correspond au niveau d'activation qui est ensuite comparé au seuil b (un scalaire) par soustraction. Finalement, la sortie du neurone est calculée par la fonction d'activation f . La sortie d'un neurone est toujours un scalaire.

II.1.2. Fonctions de transfert

Jusqu'à présent, nous n'avons pas spécifié la nature de la fonction d'activation de notre modèle. Il se trouve que plusieurs possibilités existent. Différentes fonctions de transfert pouvant

être utilisées comme fonction d'activation du neurone sont énumérées au tableau II.1. Les trois les plus utilisées sont les fonctions «seuil» (en anglais «hard limit»), «linéaire» et «sigmoïde».

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne alors la valeur 0 (on peut interpréter ce 0 comme signifiant faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Utilisée dans le contexte d'un neurone, cette fonction est illustrée à la figure II.4a. On remarque alors que le biais b dans l'expression de $a = \text{hardlim}(\mathbf{w}^T \mathbf{p} - b)$ (équation II.4) détermine l'emplacement du seuil sur l'axe $\mathbf{w}^T \mathbf{p}$, où la fonction passe de 0 à 1. Nous verrons plus loin que cette fonction permet de prendre des décisions binaires.

Le tableau II-1 résume les de transfert couramment utilisée.

Nom de la fonction	Relation d'entrée/sortie	Icône	Nom Matlab
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlim
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$		hardlims
linéaire	$a = n$		purelin
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$		satlin
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$		satlins
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$		poslin
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$		logsig
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
compétitive	$a = 1$ si n maximum $a = 0$ autrement		compet

TAB. II.1 – Fonctions de transfert $a = f(n)$

La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

$$a = n. \quad (\text{II-5})$$

Appliquée dans le contexte d'un neurone, cette fonction est illustrée à la figure II.4 b. Dans ce cas, la sortie du neurone correspond à son niveau d'activation dont le passage à zéro se produit lorsque $\mathbf{w}^T \mathbf{p} = b$.

La fonction de transfert sigmoïde est quant à elle illustrée à la figure II.4 c. Son équation est donnée par :

$$a = \frac{1}{1+\exp^{-n}} \quad (\text{II.6})$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de b , respectivement. La fonction seuil est non-linéaire car il y a une discontinuité lorsque

$w^T p = b$. De son côté, la fonction linéaire est tout à fait linéaire. Elle ne comporte aucun changement de pente. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement, que la fonction «tangente hyperbolique» est une version symétrique de la sigmoïde.

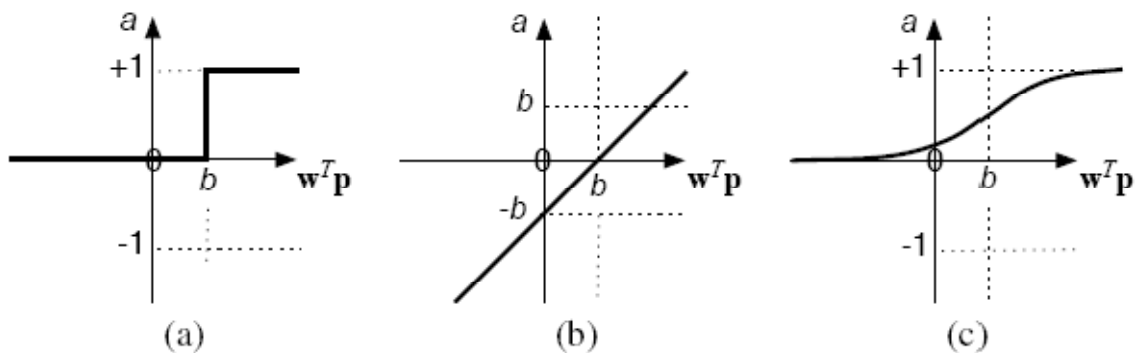


Fig. II.4 – Fonction de transfert : (a) du neurone «seuil» ; (b) du neurone «linéaire», et (c) du neurone «sigmoïde»

II.1.3. Architecture de réseau

Un réseau de neurones est un maillage de plusieurs neurones, généralement organisé en couches. Pour construire une couche de S neurones, il s'agit simplement de les assembler comme à la figure II.5. Les S neurones d'une même couche sont tous branchés aux R entrées. On dit alors que la couche est totalement connectée.

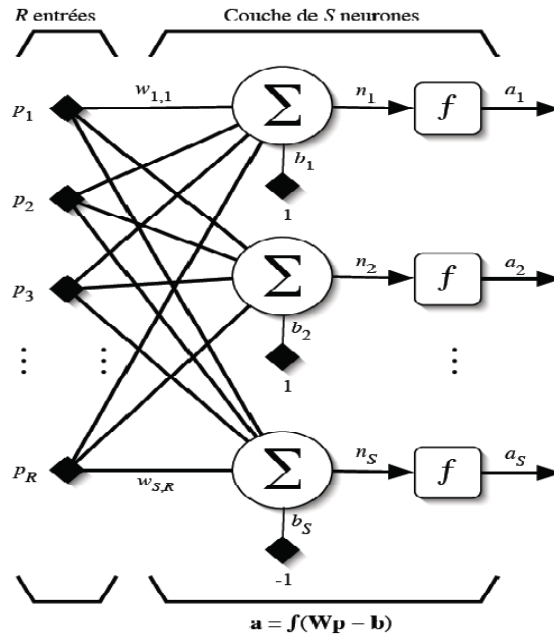


Fig. II.5 – Couche de S neurones

Un poids w_{ij} est associé à chacune des connexions. Nous noterons toujours le premier indice par i et le deuxième par j (jamais l'inverse). Le premier indice (rangée) désigne toujours le numéro de neurone sur la couche, alors que le deuxième indice (colonne) spécifie le numéro de l'entrée. Ainsi, w_{ij} désigne le poids de la connexion qui relie le neurone i à son entrée j . L'ensemble des poids d'une couche forme donc *une matrice \mathbf{W} de dimension $S \times R$* :

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad (\text{II.7})$$

Notez bien que $S \neq R$, dans le cas général (les nombres de neurones et d'entrées sont indépendants). Si l'on considère que les S neurones forment un vecteur de neurones, alors on peut créer les vecteurs $\mathbf{b} = [b_1 b_2 \dots b_S]^T$, $\mathbf{n} = [n_1 n_2 \dots n_S]^T$ et $\mathbf{a} = [a_1 a_2 \dots a_S]^T$. Ceci nous amène à la représentation graphique simplifiée, illustrée à la figure II.6. On y retrouve, comme à la figure II.3, les mêmes vecteurs et matrice. La seule différence se situe au niveau de la taille, ou plus précisément du nombre de rangées (S), de \mathbf{b} , \mathbf{n} , \mathbf{a} et \mathbf{W} .

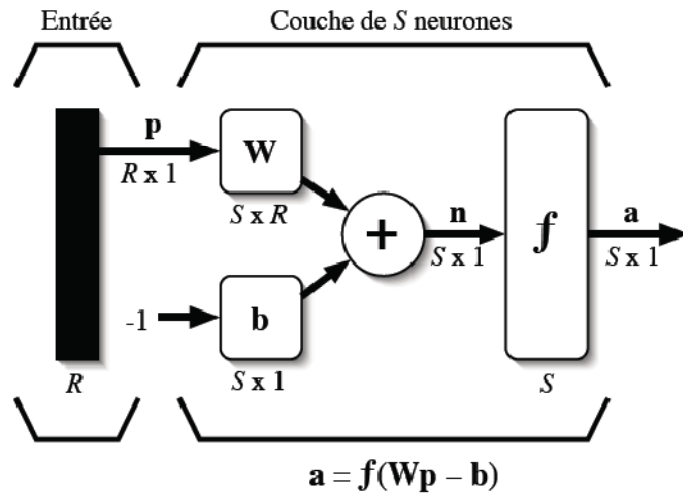


Fig. II.6 – Représentation matricielle d’une couche de S neurones

Finalement, pour construire un réseau, il ne suffit plus que de combiner des couches comme à la figure II.7. Cet exemple comporte R entrées et trois couches de neurones comptant respectivement S1, S2 et S3 neurones. Dans le cas général, de nouveau, S¹, S², S³. Chaque couche possède sa propre matrice de poids W^k, où k désigne l’indice de couche. Dans le contexte des vecteurs et des matrices relatives à une couche, nous emploierons toujours un exposant pour désigner cet indice. Ainsi, les vecteurs b^k, n^k et a^k sont aussi associés à la couche k.

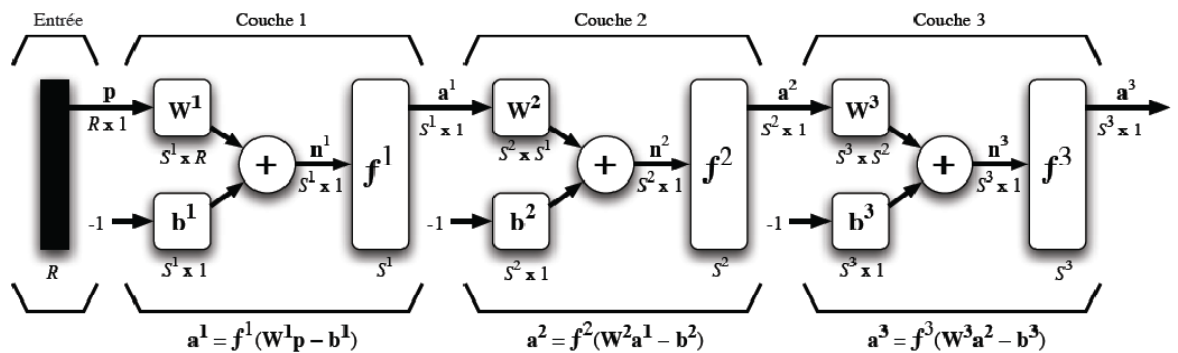


Fig. II.7 – Représentation matricielle d’un réseau de trois couches

Il importe de remarquer dans cet exemple que les couches qui suivent la première ont comme entrée la sortie de la couche précédente. Ainsi, on peut enfileur autant de couche que l’on veut, du moins en théorie. Nous pouvons aussi fixer un nombre quelconque de neurones sur chaque couche. En pratique, nous verrons plus tard qu’il n’est cependant pas souhaitable

d'utiliser trop de neurones. Finalement, notez aussi que l'on peut changer de fonction de transfert d'une couche à l'autre. Ainsi, toujours dans le cas général $f^1 \neq f^2 \neq f^3$.

La dernière couche est nommée «couche de sortie». Les couches qui précèdent la couche de sortie sont nommées «couches cachées». Nous verrons un peu plus tard pourquoi. Le réseau de la figure II.7 possède donc deux couches cachées et une couche de sortie.

Les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche. En utilisant deux couches (une couche cachée et une couche de sortie), à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, on peut entraîner un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire (cela peut cependant requérir un grand nombre de neurones sur la couche cachée). Sauf dans de rares cas, les réseaux de neurones artificiels exploitent deux ou trois couches.

Entraîner un réseau de neurones signifie modifier la valeur de ses poids et de ses biais pour qu'il réalise la fonction entrée/sortie désirée. Nous étudierons en détails, dans des chapitres subséquents, différents algorithmes pour y parvenir dans différents contextes. Pour spécifier la structure du réseau, il faut aussi choisir le nombre de couches et le nombre de neurones sur chaque couche.

Tout d'abord, rappelons que le nombre d'entrées du réseau (R), de même que le nombre de neurones sur la couche de sortie est fixé par les spécifications du problème que l'on veut résoudre avec ce réseau. Par exemple, si la donnée du problème comporte quatre variables en entrée et qu'elle exige de produire trois variables en sortie, alors nous aurons simplement $R = 4$ et $S^M = 3$, où M correspond à l'indice de la couche de sortie (ainsi qu'au nombre de couches). Ensuite, la nature du problème peut aussi nous guider dans le choix des fonctions de transfert. Par exemple, si l'on désire produire des sorties binaires 0 ou 1, alors on choisira probablement une fonction seuil pour la couche de sortie. Il reste ensuite à choisir le nombre de couches cachées ainsi que le nombre de neurones sur ces couches, et leur fonction de transfert. Il faudra aussi fixer les différents paramètres de l'algorithme d'apprentissage. Mais nous y reviendrons en temps et lieu !

Finalement, la figure II.8 illustre le dernier élément de construction que nous emploierons pour bâtir des réseaux dit «récurrents».

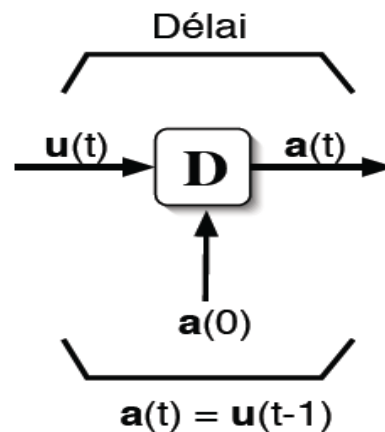


Fig. II.8 – élément de retard

Il s'agit d'un registre à décalage qui permet d'introduire un retard dans une donnée que l'on veut acheminer dans un réseau. La sortie retardée $a(t)$ prend la valeur de l'entrée u au temps $t - 1$. Cet élément de retard présuppose que l'on peut initialiser la sortie au temps $t = 0$ avec la valeur $a(0)$. Cette condition initiale est indiquée à la figure II.8 par une flèche qui entre par le bas de l'élément.

II.1.4. Le perceptron multicouches

Le perceptron multicouches (noté MLP pour Multi Layer Perceptron en anglais) est directement inspiré du raisonnement présenté au dessus. L'idée principale est de grouper des neurones dans une couche. On place ensuite bout à bout plusieurs couches et connecte complètement les neurones de deux couches adjacentes. Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche. Les neurones de la première couche sont reliés au monde extérieur et reçoivent tous le même vecteur d'entrée (c'est en fait l'entrée du réseau). Ils calculent alors leur sorties qui sont transmises aux neurones de la deuxième couche, etc. Les sorties des neurones de la dernière couche forment la sortie du réseau [13].

II.1.5. L'apprentissage (supervisé)

En ajustant les paramètres d'un MLP, on peut lui faire calculer toute sorte de fonction. Si on se donne une fonction vectorielle particulière, on peut tenter de faire apprendre cette fonction par

un MLP : c'est l'apprentissage. La méthode classique pour l'apprentissage supervisé consiste à se donner un ensemble d'exemples, c'est à dire un ensemble fini de couple de vecteurs (x_i, y_i) . Dans un tel couple, x_i désigne l'entrée du réseau et y_i la sortie désirée pour cette entrée. On écrit alors la fonction calculée par le réseau sous une forme paramétrique : $f(x, w)$ désigne la sortie du réseau quand on lui présente en entrée le vecteur x et qu'il utilise les poids synaptiques contenus dans le vecteur w . On se donne enfin une distance sur l'espace vectoriel de sortie, c'est à dire un moyen de mesurer l'erreur commise en un point par le réseau. Si cette distance est notée d , on cherche alors à trouver la valeur de w qui minimise la somme l'erreur totale commise par le réseau, c'est à dire la somme des distances entre les sorties obtenues et les sorties désirées, c'est à dire somme des $d(f(x_i, w), y_i)$. Cette erreur est une fonction de w et on peut donc utiliser les techniques classiques d'optimisation de fonction pour trouver son minimum [14].

II.1.6. La rétro-propagation

Les algorithmes d'optimisation de fonction efficaces utilisent en général la différentielle de la fonction considérée (c'est à dire son gradient car elle est à valeurs réelles). Quand les fonctions de transfert utilisées dans les neurones sont différentiables, et quand la fonction distance est aussi différentiable, l'erreur commise par un MLP est une fonction différentiable des coefficients synaptiques du réseau. L'algorithme de rétro-propagation permet justement de calculer le gradient de cette erreur de façon efficace : le nombre d'opérations (multiplications et additions) à faire est en effet proportionnel au nombre de connexions du réseau, comme dans le cas du calcul de la sortie de celui-ci. Cet algorithme rend ainsi possible l'apprentissage d'un MLP [13].

II.2.1. Erreurs dans la courbe de transfert de capteur.

Dans la production de capteur on désire produire de grandes séries de capteur à la fois, et qui répondent tous selon la même fonction transfert. Pour vérifier ceci les capteurs ont pu être mesurés individuellement. Quand on mesure le transfert d'un capteur nous pouvons être confrontés avec un ou plusieurs types d'erreurs suivantes [15]:

Offset: quand un signal physique d'entrée zéro (ou minimum) est appliqué, et le signal de sortie mesuré n'est pas zéro (ou minimum de gamme de sortie) il donne la valeur d'offset.

Gain, gamme, ou erreur plein échelle (full-scale error) : Le signal physique d'entrée maximum n'est pas exprimé par un signal électrique de sortie maximum.

Non linéarité : La variation de la sortie du capteur ne change pas linéairement avec la variation du signal physique d'entrée.

Sensibilité multiple : Le capteur est non seulement sensible au signal d'entrée mais également à d'autres paramètres.

Hystérésis : La réponse du capteur peut avoir deux valeurs différentes pour une même grandeur physique suivant un cycle (croissant ou décroissant). La différence entre ces deux valeurs donne la valeur de l'hystérésis.

Dérive de l'offset : La courbe de transfert du capteur change lentement dans le temps.

Les erreurs sont expliquées graphiquement dans la figure .II.9. Généralement, certaines de ces erreurs sont plus grandes que l'exactitude désirée. Pour corriger ces erreurs, tous les capteurs devraient être mesurés et calibrés individuellement pour garantir l'exactitude.

On peut observer les erreurs de Gain d'offset et également de linéarité et de sensibilité multiple pratiquement pour tous les types de capteurs.

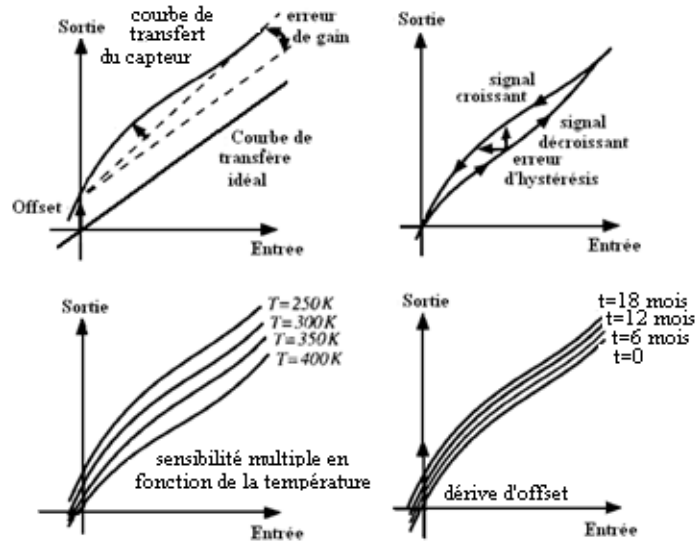


Fig. II.9 – Erreurs des courbes de transfert des capteurs

Les erreurs d'hystérésis et de la dérive de l'offset doivent être examinées pour quelques types de capteurs, mais ils peuvent être négligés pour d'autres capteurs. Bien qu'ils puissent être compliqués, les erreurs d'hystérésis pourraient être calibrées. Cependant, d'autres solutions de capteurs spécifiques devraient être considérées car elles sont être moins compliquées ou fournissent de meilleurs résultats. Les erreurs de la dérive de l'offset sont difficiles à compenser car ils exigeraient la prévision des futures erreurs « extrapolation », et la mesure de temps de réponse. Parfois il est possible de réduire la dérive de l'offset du capteur au moyen de calibrage-individuel, à l'aide d'un capteur additionnel.

II.2.2. Fonction de transfert de capteurs

Si nous voulons examiner des capteurs avec une sensibilité à une autre grandeur physique d'entrée, nous devons considérer la courbe de transfert du capteur en tant que deux fonctions. Nous pouvons exprimer la sortie électrique du capteur e_{out} comme fonction S de signal d'entrée de deux signaux physiques d'entrées φ_{in} et ψ_{in} .

$$e_{out} = S(\varphi_{in}, \psi_{in}) \tag{II-8}$$

Encore, nous pouvons transformer le signal de sortie électrique à un signal de sortie normalisée y , et les variables physiques d'entrée pour normaliser les signaux d'entrée x et z . Nous

pouvons alors examiner la fonction de transfert bidimensionnelle normalisée $y=f(x,y)$. Dans le cas idéal, la sortie normalisée y devrait dépendre linéairement de la variable d'entrée x avec l'unité de gain. Mais indépendant de l'autre entrée z (gain à zéro).

Ainsi la fonction normalisée idéale est donnée par le $y=g(x, y)=x$. les deux fonctions bidimensionnelles de transfert peuvent être présentées graphiquement comme une surface tridimensionnelles d'une image « 3D ». Les exemples sont montrés dans la figure II.10 dans ce cas, le but du calibrage est de traiter la fonction de transfert à capteur $f(X, Z)$ de telle manière que la fonction qui résulte $h(X, Z)$, laquelle ressemble à une fonction linéaire $g(X, Z)$ ceci exigera le calibrage des mesures $f(X_n, Z_m)$ pour différentes conditions d'entrée définis par $X_n[n = 1 \text{ à } N]$ et $Z_m[m = 1 \text{ à } M]$.

Le nombre de mesures de ceci s'appelle la calibration $N \times M$, nous pouvons définir l'erreur des deux dimensions de la surface par :

$$\varepsilon(X, Z) = h(X, Z) - g(X, Z). \quad (\text{II.9})$$

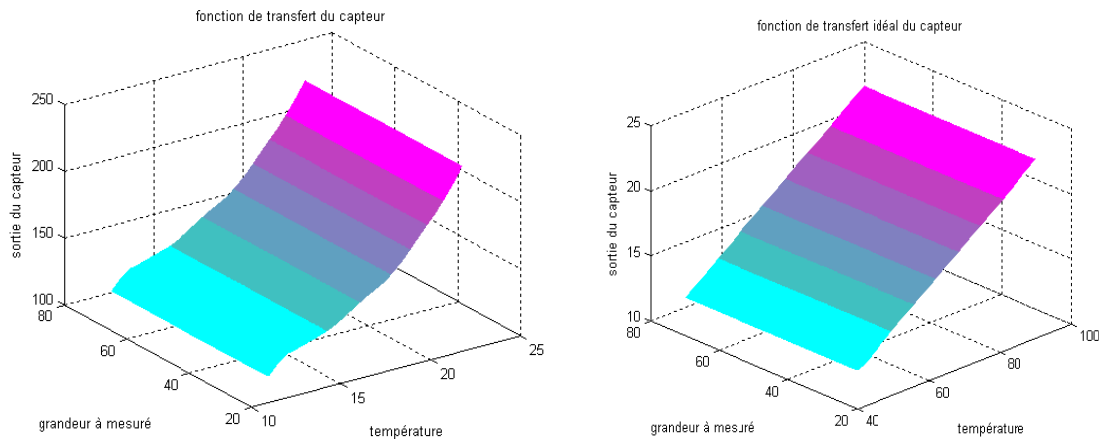


Fig.II.10 –Linéarisation idéale de la fonction de transfère

II.2.3. Linéarisation

La plupart des capteurs montrent une caractéristique de transfert non-linéaire, qui peut être représentée par un modèle physique. Le calibrage exigé peut être simplifié en appliquant d'abord une linéarisation systématique, les variations aléatoires restantes de la non-linéarité peuvent être alors calibrées en utilisant une méthode générale de linéarisation. De telles méthodes sont appliquées directement pour des capteurs à fonction de transfert plus ou moins linéaire, montrant seulement les erreurs de linéarité qui changent aléatoirement d'un dispositif à un autre. Ce qui suit explique quelques méthodes de linéarisation. Toutes ces méthodes sont basées sur l'utilisation de calibrage des mesures du signal de sortie du capteur. En plus pour linéariser le signal de sortie, le calibrage des mesures est également employé pour corriger l'erreur d'offset et de gain.

II.2.4. Linéarisation de caractéristique de capteur

Certains capteurs montrent une réponse non linéaire, d'une forme de non linéarité connue. Comme une erreur systématique devrait de préférence être corrigée par une linéarisation systématique, plutôt qu'au moyen d'interpolation des mesures multiples de calibrage. Un capteur avec une courbe de transfert logarithmique pourrait être écrit sous la forme de l'équation (II-10) avec a, b, c, et d sont des constantes dépendantes du capteur.

$$e_s = f(\varphi) = a + b \cdot \ln(c + d\varphi) \quad (\text{II-10})$$

Une courbe de transfert logarithmique peut facilement être linéarisée en utilisant un circuit électronique avec une fonction de transfert exponentielle « figure II.11 ». Nous pourrions employer la relation exponentielle entre la tension V_{be} et le courant de collecteur d'un transistor bipolaire. Le signal de sortie de capteur e_s est la tension V_{be} , et le courant de collecteur I_C est converti à une tension de sortie V_{out} par un convertisseur pour le transfert global de capteur et du circuit, nous réalisons.

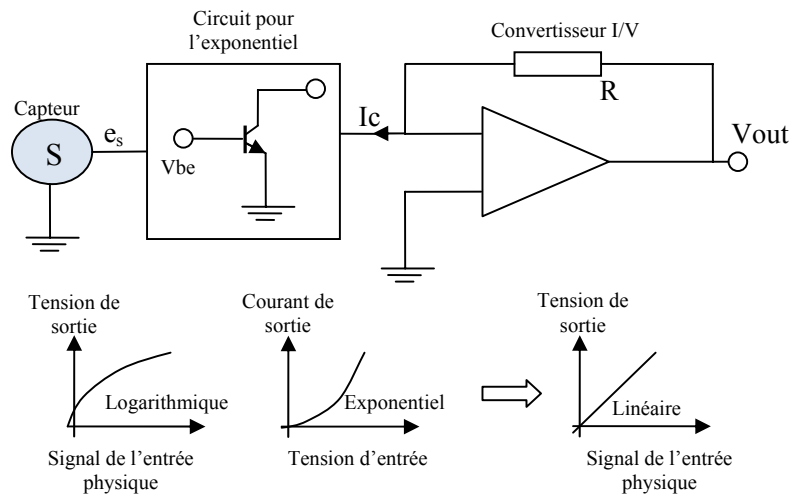


Fig.II.11 – Linéarisation de caractéristique de capteur

$$\begin{cases} V_{out} = h(\varphi) = R I_C = R e^{K_T V_{be}} \\ \quad = R e^{K_T \{a + b \ln(c + d \varphi)\}} \\ \quad = R e^{a K_T} e^{b K_T (c + d \varphi)} \end{cases} \quad (II-11)$$

Ce qui peut être écrit sous une forme simplifiée comme :

$$h(x) = A + B.x \quad (II-12)$$

II.2.5. Linéarisation basée sur l'ajustement de courbe « Curve-fitting »

Il est possible de contrecarrer le transfert non-linéaire du capteur avec un transfert inversement non linéaire quand l'expression de la courbe de transfert du capteur est connue. Il est également possible quand la fonction de transfert du capteur n'est pas connue avec précision mais est seulement caractérisée par quelques mesures de calibrage. Les techniques d'ajustement de courbe « Curve-fitting » de précision permettent de calculer une fonction mathématique qui intersecté la fonction réelle de capteur dans les points de mesure [16].

Ceci est habituellement fait en définissant une somme de poit « weighted » d'expressions basées sur la sortie de capteur $y = f(x)$. Les facteurs de poids sont calculés sur la base du $f(X_n)$ de la fonction de transfert du capteur. On peut alors essayer de calculer la fonction inverse exacte de la fonction trouvée d'ajustement de courbe. Une meilleure approche est de composer une fonction $H [f(x)]$ d'ajustement de courbe pour le transfert inverse du capteur. L'ajustement de courbe se rapporte alors à assortir la fonction de transfert résultante $h(x) = [f(x)]$ de la fonction de transfert linéaire désirée $g(x)$. La fonction de linéarisation peut être exprimée sous la forme suivante :

$$h(x) = \sum_{n=0}^N C_n \cdot E_n \{f(x)\} \quad (\text{II-13})$$

L'expression dépend de la technique spécifique d'ajustement de courbe utilisée, et pourrait être, par exemple, une fonction sinusoïdale, exponentielle ou polynôme de l'ordre croissant (n). Le facteur de poids C_n , également désigné sous le nom des coefficients de calibrage, sont calculés, en utilisant les mesures de calibrage $f(X_n)$ et les valeurs correspondantes pour le transfert désiré $g(X_n)$.

Les calculs exigés peuvent se devisés en deux groupes. D'abord, le calcul des valeurs correctes pour les coefficients de calibrage pendant ou après la phase de calibrage. Ceci peut être fait extérieurement du capteur (sur un ordinateur). Le calcul secondaire du signal de sortie linéarisé basé sur la sortie et les coefficients de calibrage. Cette correction doit être intégrée avec le capteur intelligent, dans le matériel analogue ou numérique, ou dans le logiciel dans un microcontrôleur connectant le capteur. La mémoire doit être présente pour stocker les coefficients de calibrage.

Conclusion

La majorité des capteurs montrent des différentes erreurs de mesure, la connaissance de ces erreurs nous permettent de les corriger, les courbes de transfert non linéaires peuvent être ajustées pour les connaître puis les corriger. Cette correction peut être effectuée à partir d'un modèle de réseau de neurones, on peut construire des perceptrons multicouches. Ces outils permettent de calculer des fonctions vectorielles, adaptables à un ensemble d'exemples par le biais d'algorithmes d'optimisation utilisant la technique de la rétro-propagation. De cette façon, on peut espérer faire apprendre une fonction complexe à un MLP, contenant assez peu de neurones. On obtiendra ainsi une modélisation analytique compacte d'une fonction obtenue expérimentalement.

Chapitre III : Modélisation du
capteur d'humidité par les

ANN

Introduction

L'objectif de ce troisième chapitre est de créer un modèle à base des réseaux de neurone « ANN-model » pour le substituer à la réponse capacitive du modèle utilisé (modèle CHS « Chapitre I ») et de la modélisation statique des réseaux de neurones « Chapitre II » réalisés à l'aide du logiciel **Matlab**. L'implantation du ANN-model sur le simulateur **SPICE** ainsi que les résultats de simulation obtenus sont également reportés.

La construction d'un modèle à base de ANN pour le CHS, est de reproduire fidèlement son comportement lors de la phase de conception du capteur intelligent (chapitre IV). Autrement dit, ce modèle servira à la simulation comportementale du capteur intelligent, vu que le modèle analytique du CHS est limité à des conditions moins dynamiques.

III.1. Modélisation du CHS

Pour créer le modèle ANN du CHS au moyen des réseaux de neurones on choisit une base de données caractérisée, dans notre cas, par l'humidité appliquée au CHS, la température de l'environnement où le CHS est placé et la réponse du CHS. Dans une deuxième étape on sépare la base d'apprentissage et celle de test, puis on fait l'entraînement d'un réseau de neurones sur la base d'apprentissage avec l'algorithme de rétropropagation et finalement on mesure la performance du modèle obtenu avec la base de test.

III.1.1. Choix de la base de données

Le capteur que nous avons étudié couvre une gamme d'humidité de 45% à 95% et un intervalle de température entre 25 à 75°C. La base de données doit couvrir cet ensemble de valeurs [8]. En se basant sur les résultats expérimentaux du chapitre I, et en choisissant un pas d'humidité 2,5% et trois températures 25, 50 et 75°C, notre base de données possèdera donc 21 X 3 éléments et 3 vecteurs de la forme $X(T, H, C)$, et l'apprentissage du réseau de neurones est fait avec une base de données de 189 éléments figure III.1.

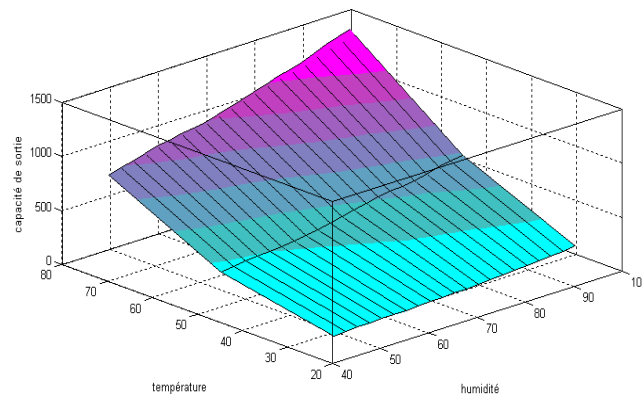


Fig.III.1- Représentation en 3 dimensions de la base de données choisie

III.1.2. Création d'une base de validation

Comme son nom l'indique cette base de données est utilisée pour tester « valider » les résultats obtenus après l'apprentissage. On doit noter ici qu'il n'y a pas de règles précises concernant cette séparation, néanmoins, d'une manière générale la base de test représente entre 10% et 25% de la base de données, suivant le problème étudié. Les deux bases de données ainsi obtenues par cette séparation doivent impérativement couvrir l'espace de fonctionnement. Dans le cas pratique, si la température de travail du CHS, varie entre 25°C et 75°C, alors les deux bases englobent des différentes valeurs distribuées au long de cet intervalle, le même principe est appliqué à l'intervalle de variation d'humidité (45% à 95%). Dans notre cas la base d'apprentissage est composée de 189 éléments, quand à la base de test elle est composée de 21 éléments. Il est important de ne pas utiliser aucun élément de la base de test pendant toute la durée de l'apprentissage. Cette base est réservée uniquement à la mesure finale de la performance. Autrement dit, elle sert à vérifier si le réseau de neurones a une bonne performance sur les exemples qu'il n'a pas appris « base de test ». Avec les réseaux de neurones, il existe toujours le risque de sur-apprentissage, c'est-à-dire, quand le réseau a pris trop de paramètres pour représenter une fonction qui n'est pas très complexe. La base de validation permet de mettre en évidence le problème s'il se présente. Le sur apprentissage se traduit par une augmentation de l'erreur sur la base de validation [17].

III.1.3.L'apprentissage du réseau de neurone

Comme nous avons vu précédemment, l'apprentissage nécessite une base de données et une base de tests, et le nombre de couches et des neurones utilisés dans chaque couche.

Le bon choix, de type des fonctions d'activations et du nombre d'itérations nécessaires pour avoir le seuil d'estimation voulue « S » qui représente la valeur minimale de l'erreur quadratique moyenne « EQM » qu'on désire obtenir, est aussi nécessaire pour faire notre apprentissage.

La figure III.2 représente l'organigramme qui interprète notre programme structuré en Matlab.

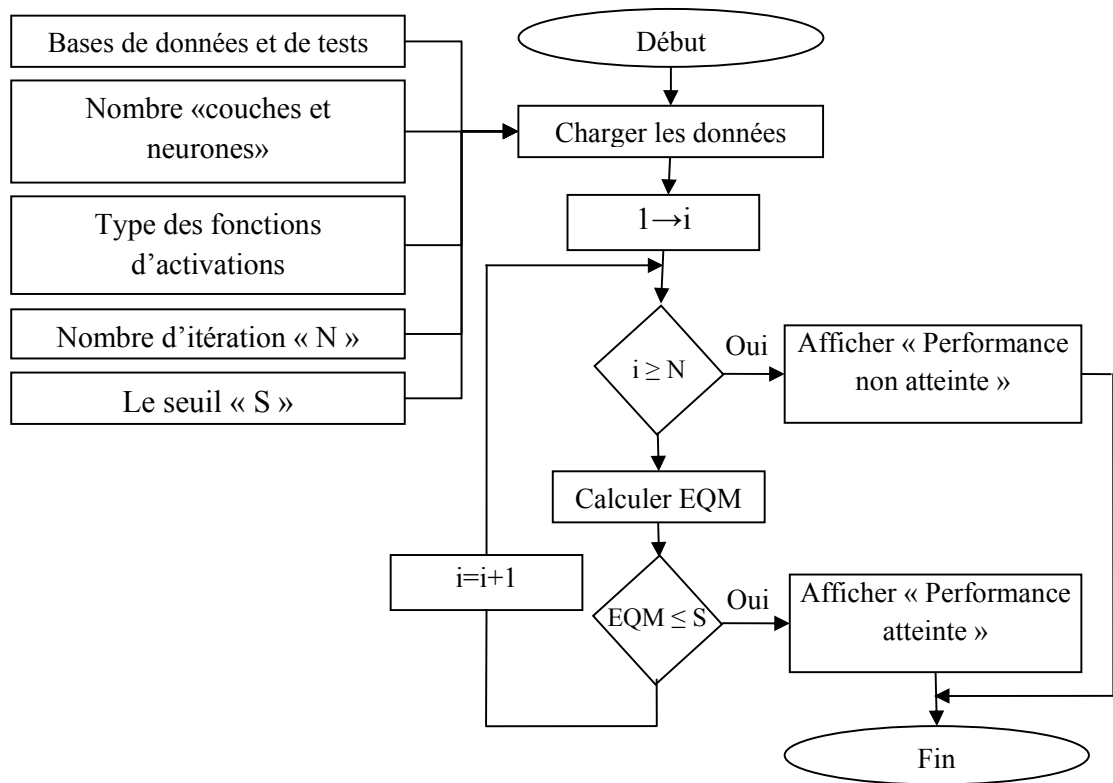


Fig III.2-Organigramme pour le programme d'apprentissage

Dans notre cas, puisque le nombre des neurones dans la couche de sortie est déterminé par le nombre de sorties du système à modéliser, le CHS possède une seule sortie C « capacité » donc un neurone pour la couche de sortie.

Pour que le modèle ANN exprime fidèlement la variation de la réponse du CHS, il s'agit de trouver le nombre optimal des couches cachées et le nombre de neurones par couche. Nous avons envisagé deux couches, 4 neurones pour la première couche et 5 neurones pour la deuxième couche cachée « figure III.3 ».

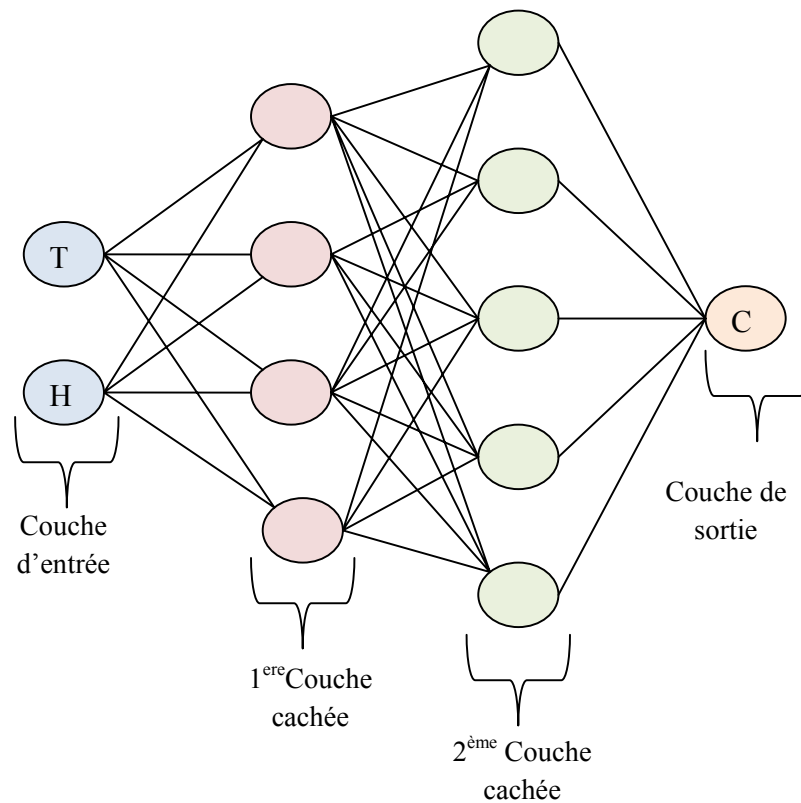


Figure III.3 - Schémas des couches de neurones de l'architecture optimisée

Cette architecture a été choisie par un test de réseau avec un nombre de neurones entre 1 et 6 neurones pour la première couche et entre 0 et 6 neurones pour la deuxième, puis nous avons observé la variation de l'erreur d'apprentissage et de test. Le tableau III.1 représente les différents résultats obtenus, cependant l'erreur de test converge vers le minimum dans l'architecture numéros 27 ce qui justifie notre choix.

	Nombre de neurones 1 ^{er} couche	Nombre de neurones 2 ^{ème} couche	Le seuil « S » Pour EQM=10 ⁻⁸
1	1	0	1886.5125
2	1	1	15.30288
3	1	2	2.6060
4	1	3	1.80466
5	1	4	11.2534
6	1	5	18.9731
7	1	6	11.8570
8	2	0	17.7176
9	2	1	20.40993
10	2	2	11.5386
11	2	3	8.1313
12	2	4	6.5250
13	2	5	8.6254
14	2	6	8.0825
15	3	0	32.18986
16	3	1	14.9064
17	3	2	11.8550
18	3	3	9.2842
19	3	4	4.9679
20	3	5	1.6880
21	3	6	0.52292
22	4	0	18.49098
23	4	1	10.2610
24	4	2	1.74168
25	4	3	0.61161
26	4	4	0.93212
27	4	5	0.01015
28	4	6	0.06659
29	5	0	18.67096
30	5	1	1.09208
31	5	2	0.35022
32	5	3	0.24496
33	5	4	0.04261
34	5	5	0.04120
35	5	6	0.09523
36	6	0	10.2365
37	6	1	5.6978
38	6	2	2.3654
39	6	3	0.07012
40	6	4	0.2356
41	6	5	0.0660
42	6	6	1.6020

Tab III.1 - Variation de l'erreur du test en fonction des différentes architectures

De même pour les choix des types des fonctions d'activations, nombre maximum d'itération et l'EQM. Le test et l'observation, de la variation de l'erreur, pour des différentes valeurs est la solution pour avoir les meilleurs résultats. Afin d'évaluer l'influence du seuil S sur le nombre d'itérations nécessaires pour obtenir $EQM < S$. Nous avons choisi plusieurs valeurs pour S , les figures III.4 et III.5 et III.6 représentent l'évolution du EQM en fonction de nombre d'itérations pour des différentes valeurs du seuil.

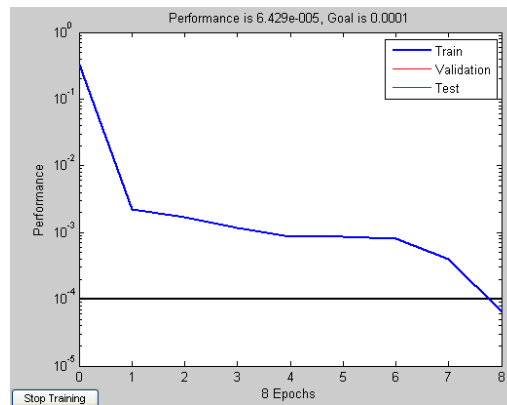


Figure III.4 - EQM en fonction du nombre d'itérations pour $S=10^{-4}$

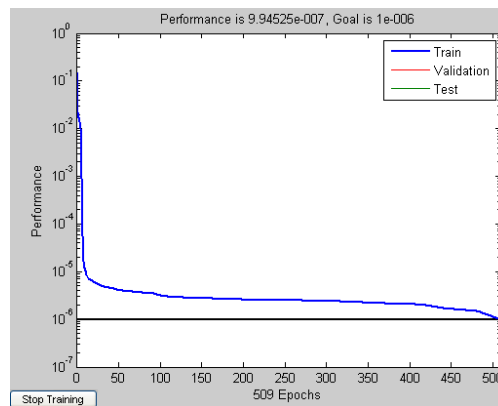


Figure III.4 - EQM en fonction du nombre d'itérations pour $S=10^{-6}$

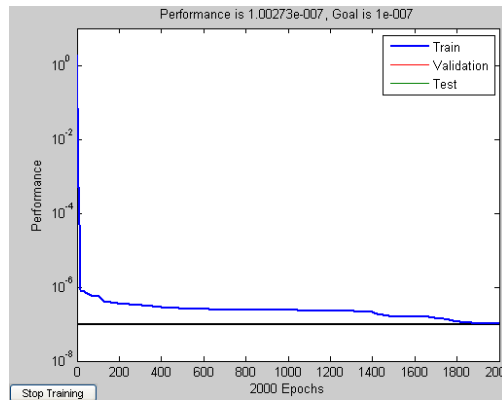


Figure III.5 - EQM en fonction du nombre d'itérations pour $S=10^{-7}$

Le nombre d'itérations montré par les figures précédentes indique que plus le seuil « S » est faible, plus ce nombre d'itérations est important et vis-vers-ça.

III.1.4. Mesure de la performance du modèle obtenu

La comparaison entre la base de données initial et celle obtenue après l'apprentissage, utilisant la base de test, indique que notre modèle exprime fidèlement la variation de la réponse du CHS. Les figures III.6, III.7 et III.8 présentent la performance du modèle obtenu pour les trois températures 25, 50 et 75°C.

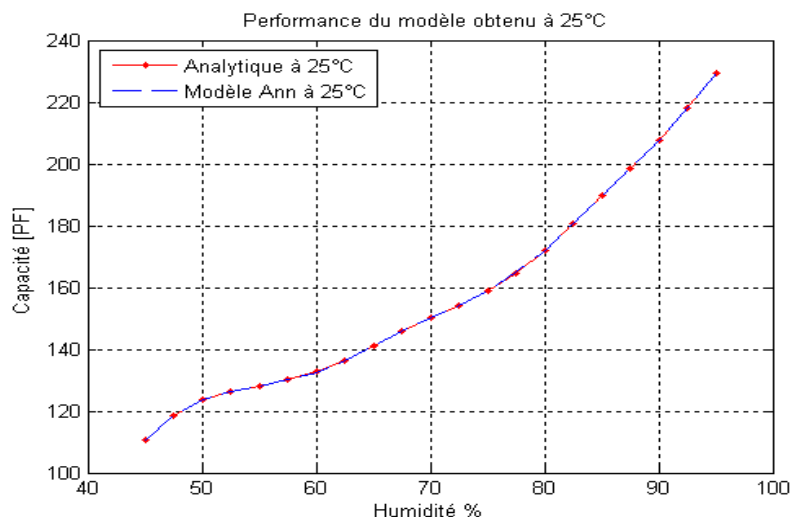


Figure III.6 – la performance du modèle obtenu à 25°C

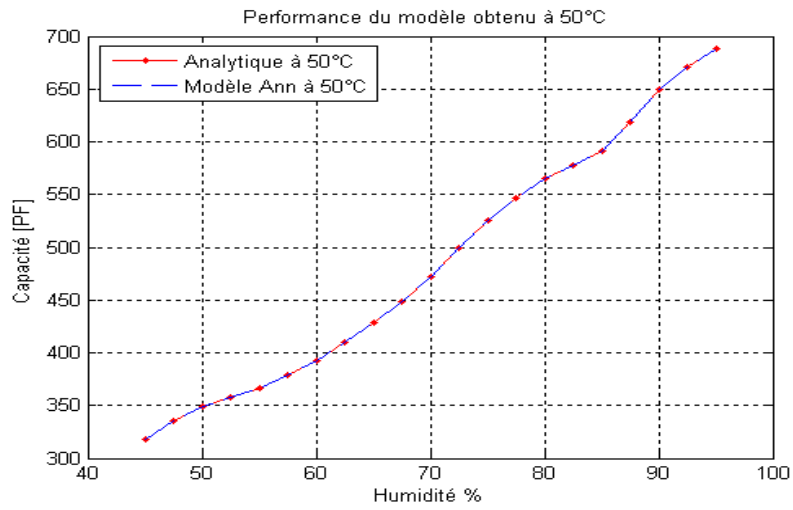


Figure III.7 – la performance du modèle obtenu à 50°C

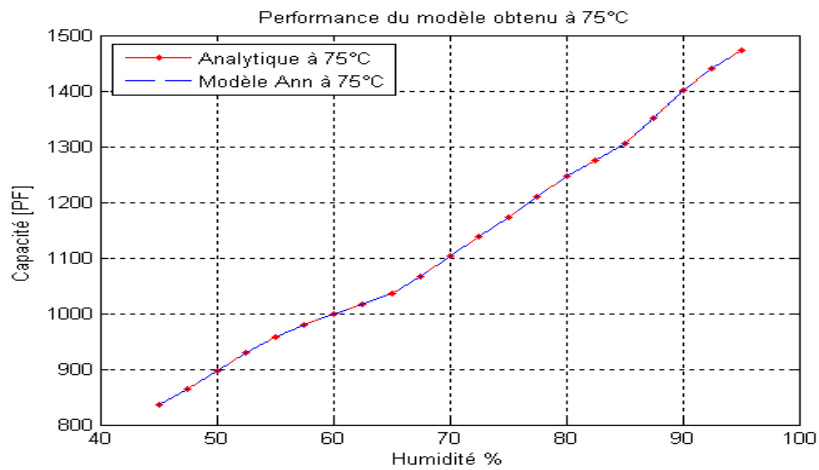


Figure III.8 – la performance du modèle obtenu à 75°C

Le tableau III.2 résumé tous les paramètres utilisés pour l'optimisation du modèle ANN du capteur CHS.

Paramètre	Valeur optimisée	
Architecture	Feed-forward MLP (perceptron multi-couches)	
Couche cachée	2	
Règle d'apprentissage	Rétropropagation des erreurs (Back propagation)	
Nombre de Neurones	Couche d'entrée	2
	1ère couche	4
	2ème couche	5
	Couche de sortie	1
La fonction de transfert	1ère couche	Sigmoid
	2ème couche	Sigmoid
	Couche de sortie	Linéaire
Définition des entrées	H (%)	
	Max	95%
	Min	45%
Définition des sorties	C_H (nF)	
	Max	1500
	Min	110
EQM de test	10^{-8}	
EQM d'apprentissage	$9.899 \cdot 10^{-8}$	
Base de données	Base d'apprentissage	189
	Base de validation	21

Tableau III.2 : Paramètres optimisés du réseau de neurones

III.2. Modèle ANN du CHS sur PSPICE

L'utilisation des boîtes ABM « Analog Behavioral Modelling » de la bibliothèque de PSPICE, nous permet d'implanter le modèle ANN sur ce simulateur. Le modèle ANN du CHS possède 10 neurones 4 pour la première couche cachée, 5 pour la deuxième et une seule pour la couche de sortie, chaque neurone du réseau est remplacé par une boîte ABM sur PSPICE et une autre ABM pour la mise en échelle de la capacité de sortie. La figure III.8 représente la structure du réseau des ABM de notre modèle.

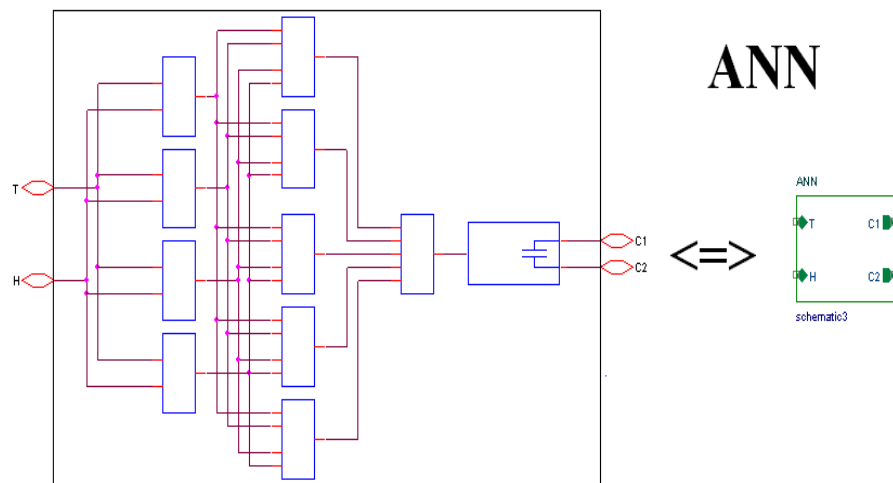


Figure III.9 – Implantation du modèle ANN sur PSPICE

Chaque ABM équivalente à un neurone est caractérisé par l'équation de ce dernier, l'équation de l'ABM 1 par exemple :

$$1/(1+\exp(-(B11+W11*V(T)+W12*V(H))))$$

La forme d'exponentiel dans l'équation est due au choix de la fonction de transfert sigmoïde dans la première couche cachée, B11 est le premier biais pour la première couche cachée, dans la matrice du biais « Bni », W111 et W112 sont respectivement le premier et le deuxième poids pour la première couche cachée, dans la matrice des poids « Wnij ».

- ❖ Bni : la matrice du biais « seuil W0 »
 - n : numéro de la couche
 - i : numéro du neurone
- ❖ Wnji : la matrice des poids
 - n : numéro de la couche
 - j : numéro du neurone
 - i : numéro du neurone de la couche précédente

La netlist du composant CHS est :

*----- Modèle CPH -----

*----- la couche d'entrée

.EXTERNAL INPUT H

.EXTERNAL INPUT T

*----- la sortie C ' pin C1, pin C2 '

.EXTERNAL OUTPUT C1

.EXTERNAL OUTPUT C2

*----- la première couche cachée

E_CHS_ABM1 N24600 0 VALUE {1/(1+exp(-(B11+W111*V(T)+W112*V(H))))}
 E_CHS_ABM2 N24530 0 VALUE {1/(1+exp(-(B12+W121*V(T)+W122*V(H))))}
 E_CHS_ABM3 N24542 0 VALUE {1/(1+exp(-(B13+W131*V(T)+W132*V(H))))}
 E_CHS_ABM4 N24556 0 VALUE {1/(1+exp(-(B14+W141*V(T)+W142*V(H))))}

*----- la deuxième couche cachée

E_CHS_ABM5 N41663 0 VALUE {1/(1+exp(-
 (B21+W211*V(N41207)+W212*V(N41511)+ W213*V(N41208)+W214*V(N41512)))) }
 E_CHS_ABM6 N41696 0 VALUE {1/(1+exp(-
 (B2+W221*V(N41207)+W222*V(N41511)+W223*V(N41208)+W224*V(N41512)))) }
 E_CHS_ABM7 N41761 0 VALUE {1/(1+exp(-
 (B23+W231*V(N41207)+W232*V(N41511)+W233*V(N41208)+W234*V(N41512)))) }
 E_CHS_ABM8 N41973 0 VALUE {1/(1+exp(-
 (B24+W241*V(N41207)+W242*V(N41511)+W243*V(N41208)+W244*V(N41512)))) }
 E_CHS_ABM9 N41725 0 VALUE {1/(1+exp(-
 (B25+W251*V(N41207)+W252*V(N41511)+W253*V(N41208)+W254*V(N41512)))) }

*----- la couche de sortie

E_CHS_ABM10 N394804 0 VALUE
 {B31+W311*V(N41663)+W312*V(N41696)+W313*V(N41761)+W314*V(N41973)+W315*V(
 N41725)}

* ----- la mise en échelle (C en nF)

C_ABM11 1 2 VALUE { (V(N394804) * 1E-9) }

*----- END

III.3. Confirmation du comportement capacitif du modèle CHS

Le but de cette simulation est de prouver le comportement capacitif de la sortie du composant obtenu auparavant, par un simple circuit RC de charge et décharge de la capacité, dans notre cas le CHS joue le rôle de cette capacité, qui est en série avec une résistance et un générateur de signal carré.

Les figures III.10 et III.11 représentent respectivement le circuit de simulation « Time domain » et le signal obtenu à la sortie de CHS.

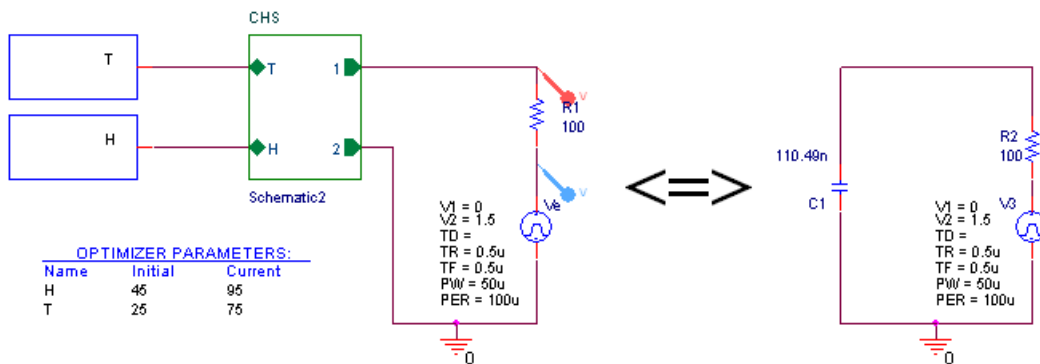


Figure III.10 – Circuit RC obtenu par le capteur CHS

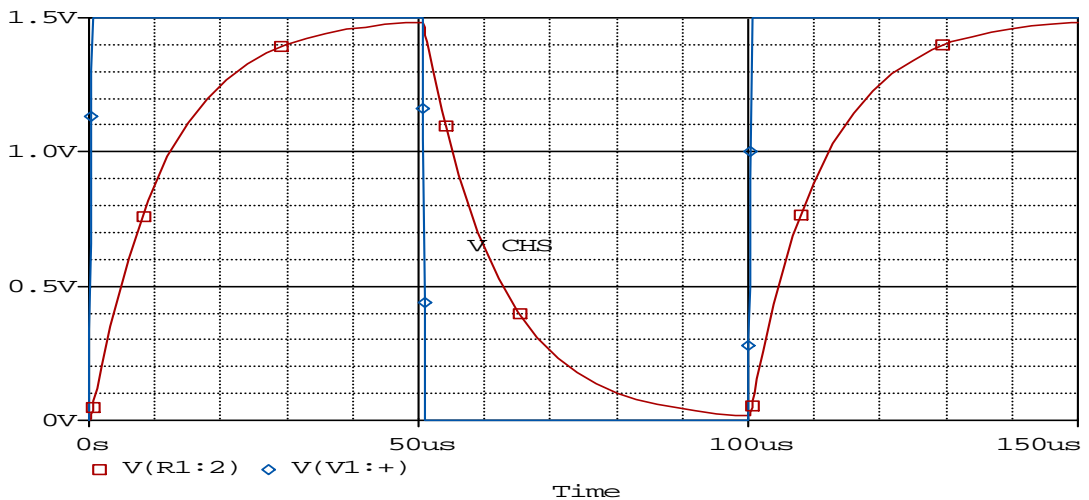


Figure III.11 – Signaux d'entrée et de sortie du circuit RC

III.4. Convertisseur Capacité / Tension (CVC)

Le circuit précédemment simulé montre la nature capacitive de notre capteur CHS, cependant, la variation de cette capacité en fonction de l'humidité, est une variation d'une grandeur passive qui nécessite une conversion à une grandeur active « tension ou courant » pour l'exploiter. Le circuit présenté par la figure III.12 est un convertisseur capacité tension CVC « Capacity to Voltage Converter » à base des capacités commutées SCI « Switched Capacitor Interface », on fait appel à une interface, cette interface délivre une tension proportionnelle à la capacité du CHS.

Lorsque $V_{com}=0$ la capacité $C(H)$ se charge avec la tension V_{ref} au moment où la capacité C_s se décharge à la masse. Pour $V_{com}=1$ la charge $Q=V_{ref}.C_h$ stockée dans la capacité $C(CHS)$ est transférée à la capacité C_s ($V_0.C_s=V_{ref}.C_h$) qui produit une tension de sortie $V_0 = K.C_h$ avec $K=V_{ref} / C_s$. Finalement la tension V_0 est amplifiée avec un gain en tension 'G' ce qui donne $V_{S1} = G . V_0$.

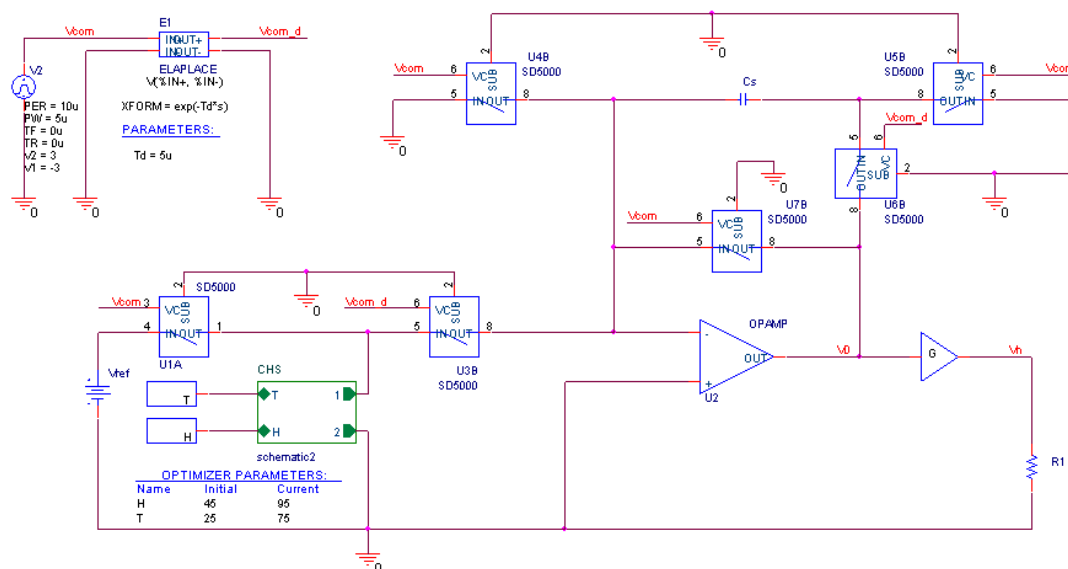


Figure III.12 - Schéma électrique du CVC

Les trois courbes obtenues par une simulation paramétrique, pour trois températures 25 50 et 75°C sur une gamme d'humidité entre 45 et 95% sont illustrées par la figure III.13.

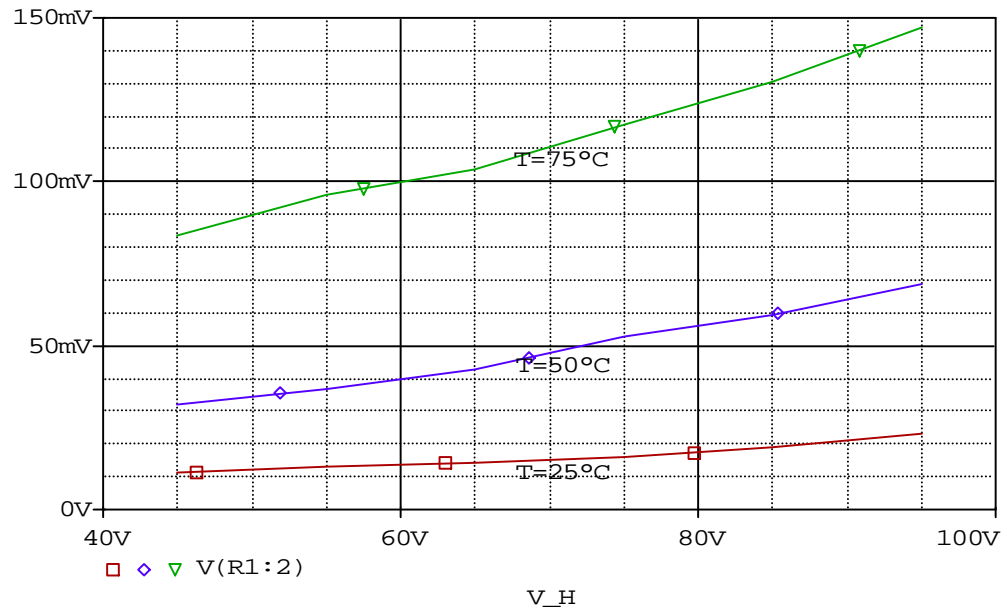


Figure III.13 – Résultats de la simulation paramétrique pour les températures 25 50 75°C

Cette dernière simulation démontre la reproduction fidèle du comportement du CHS, la linéarisation de la réponse du CHS et l'élimination de l'effet de la température sera notre objectif dans le chapitre suivant.

Conclusion

La reproduction d'un modèle du capteur CHS, qui délivre la même sortie de CHS, avec ses avantages et ses inconvénients à été effectuée dans ce chapitre. La création du modèle par les réseaux de neurones et Matlab, ensuite l'implantation de ce modèle sur l'environnement de simulation PSPICE et les différentes simulations effectuées, ont été présentés dans cette partie.

Chapitre IV : Modèle
intelligent (INV-ANN)

Introduction

Il est possible de contrecarrer la non linéarité de capteur avec un transfert inversement non linéaire quand l'expression de la courbe de transfert du capteur est connue. Les capteurs intelligents intègrent des circuits de conditionnement, dont le rôle est la correction des réponses. L'utilisation des réseaux de neurones comme des composants de mesure, permettant de corriger la réponse des capteurs, a connu un essor considérable au cours de ces dernières années. En effet, les ANNs présentent l'avantage d'une grande adaptation aux différents problèmes causés par la non idéalité des capteurs [18] [19].

Ce chapitre est consacré à, premièrement concevoir un composant de mesure « INV-ANN » permettant la correction de la réponse du capteur d'humidité capacitif « CHS ». Ensuite à l'élaboration du schéma complet du capteur intelligent puis à la présentation de quelques simulations pour montrer la validité du modèle.

IV.1. Modélisation du INV-ANN

Pour la création du modèle inverse INV-ANN, dit aussi modèle intelligent, on commence par la linéarisation de la réponse du modèle CHS obtenue au en utilisant les réseaux de neurones. On choisi une base de données caractérisée, cette fois par la tension relevée à la sortie du convertisseur capacité tension CVC, la température du milieu de détection et la réponse linéarisée du INV-ANN. Dans une deuxième étape on sépare la base d'apprentissage et celle de test, puis on fait l'entraînement d'un réseau de neurones sur la base d'apprentissage avec l'algorithme de rétropropagation et finalement on mesure la performance du modèle obtenu avec la base de test.

IV.1.1. Linéarisation de la sortie du modèle CHS

Pour éliminer la non linéarité de la réponse sur notre modèle, il est nécessaire d'effectuer une linéarisation. Choisisant à titre d'exemple la courbe à 25°C. La figure IV.1 illustre l'entrée du INV-ANN (V_{S1} à 25°C, 50°C et 75°C) et la sortie V_S linéarisée.

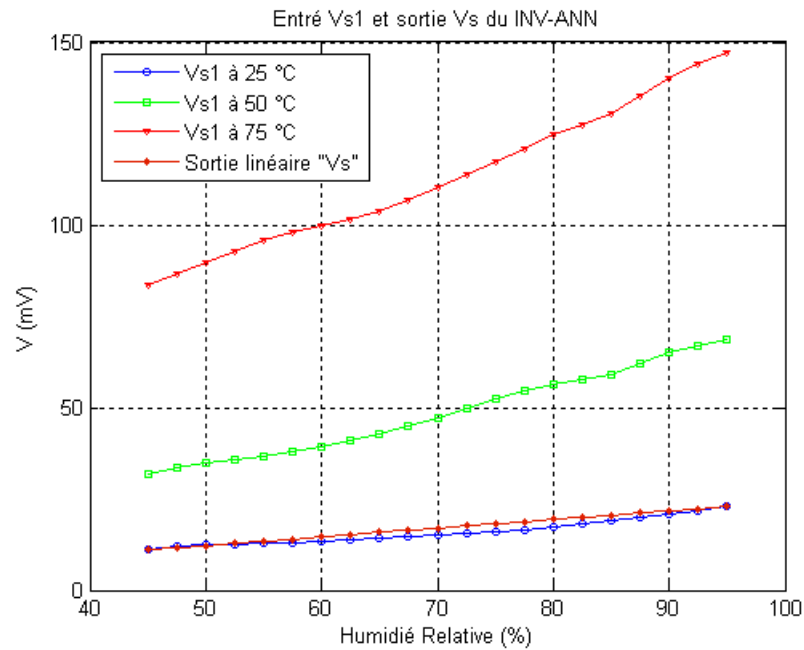


Fig.IV.1-Linéarisation de la réponse à 25°C

IV.1.2.Création des bases de données et de tests

La base de données de INV-ANN possède 63 vecteurs de la forme X (T, H, C, Vs1, Vs). avec T est la température, H est l’humidité, C est la capacité de capteur CHS, Vs1 est la tension a la sortie de CVC et Vs est la tension après la linéarisation. La figure IV.2 représente les éléments T, Vs1 et Vs de la base de données du vecteur X après la linéarisation.

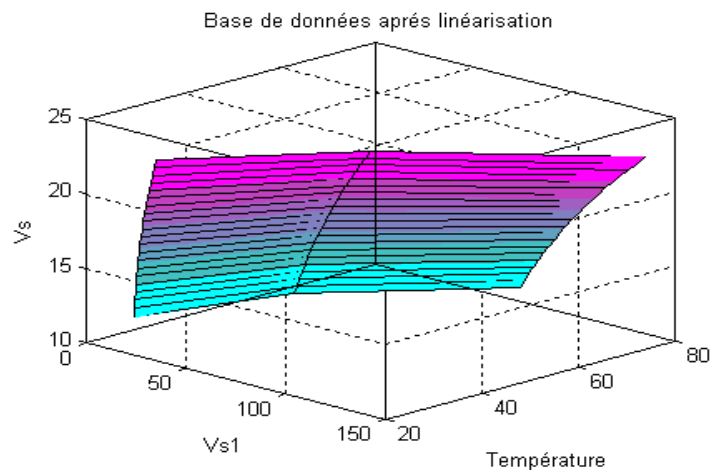


Fig.IV.2-Représentation en 3 dimensions des éléments essentiels de la base de données après linéarisation

Après l'obtention de la base de données, il est nécessaire de réserver un ensemble des éléments pour la mesure finale de la performance ; c'est la base de validation.

IV.1.3.L'apprentissage du réseau de neurone

Après la création de la base de données et celle de validation, qui sont nécessaires pour l'apprentissage, on garde la même structure neuronal ; le nombre des couches cachées « 2 », le nombre des neurones utilisés dans chaque couche cachée (« 4 » pour la première et « 5 » pour la deuxième), et on garde aussi le même choix de type des fonctions d'activations. Pour que le modèle INV-ANN exprime précisément la linéarisation de la réponse du CHS. Cette structure est illustrée par la figure IV.3.

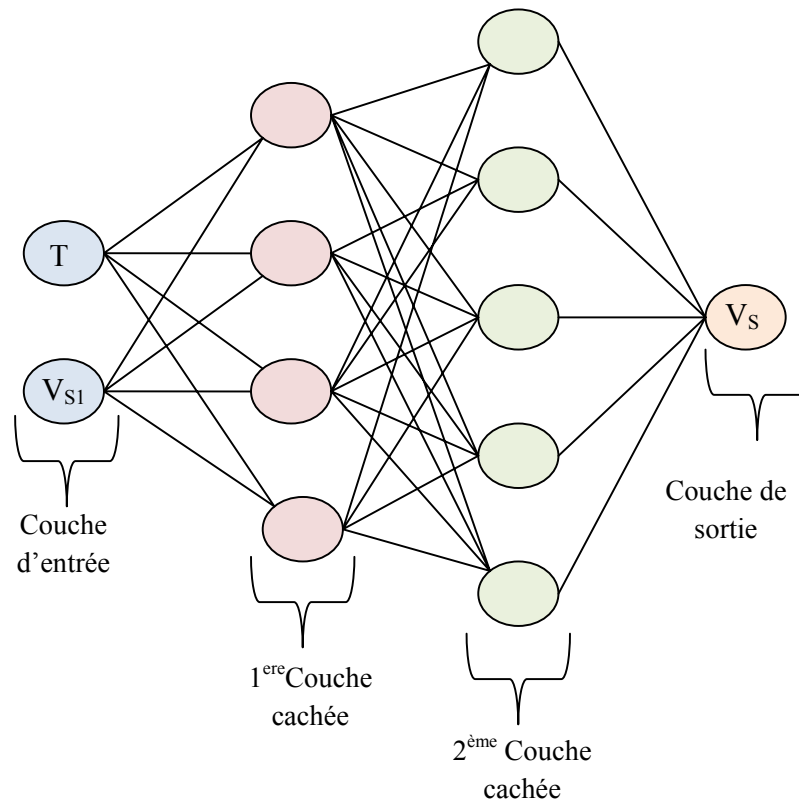


Fig.IV.3- Représentation de la structure neuronale de INV-ANN

Nous avons, maintenant, tous les éléments essentiels pour commencer l'apprentissage du modèle inverse INV-ANN respectant les mêmes procédures adoptés à l'ANN pour le

développement du modèle INV-ANN. La figure IV.4 illustre l'évolution de l'erreur globale sur la base d'apprentissage en fonction du nombre d'itérations.

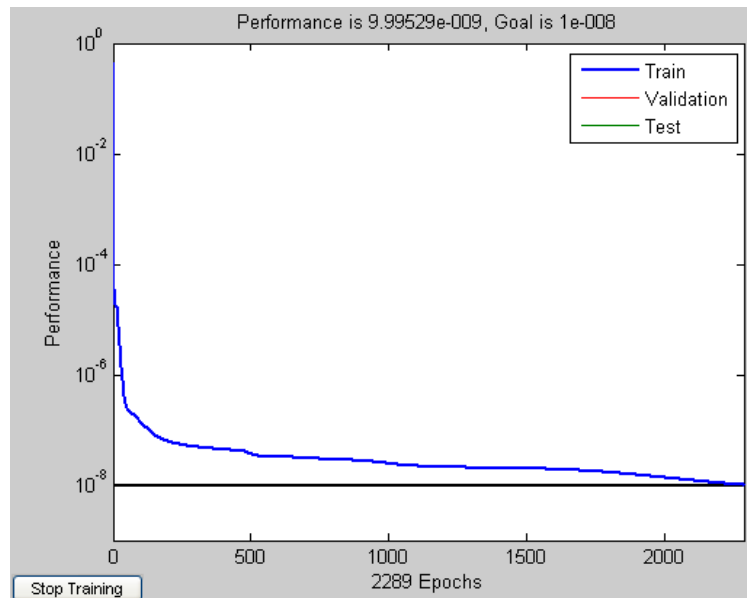


Fig.IV.4-Evolution de l'erreur EQM en fonction du nombre d'itérations pour $S=10^{-8}$

IV.1.4.Mesure de la performance du modèle obtenu

Un programme Matlab nous permet de faire une comparaison entre la base de données linéarisée et celle obtenue après l'apprentissage. La figure IV.5 présente le signal d'entrée de INV-ANN à 25°C avec les deux courbes de linéarisation obtenues après l'apprentissage et celle idéale pour montrer la performance du modèle obtenu.

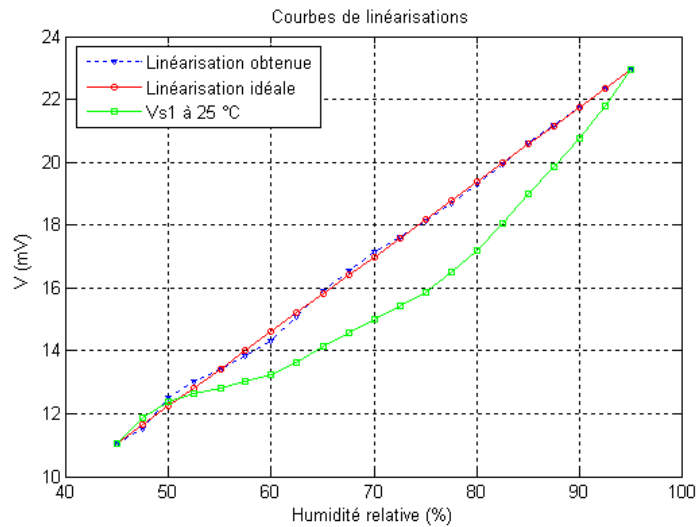


Fig.IV.5-performance du modèle après l'apprentissage

Le tableau IV.1 résume les différents paramètres optimisés.

Paramètre	Valeur optimisée		
Architecture	perceptron multi-couches (Feed-forward MLP)		
Couche cachée	2		
Règle d'apprentissage	Rétropropagation des erreurs (Back propagation)		
Nombre de Neurones	Couche d'entrée	2	
	1ère couche	4	
	2ème couche	5	
	Couche de sortie	1	
La fonction de transfert	1ère couche	Sigmoid	
	2ème couche	Sigmoid	
	Couche de sortie	Linéaire	
Définition des entrées		V_{S1} (mV)	T (°C)
	Max	150	75°C
	Min	11	25°C
Définition des sorties		V_S (mV)	
	Max	150	
	Min	11	
EQM de test	10^{-8}		
EQM d'apprentissage	$9.9777 \cdot 10^{-8}$		
Base de données	Base d'apprentissage	189	
	Base de validation	21	

Tableau IV.1 : Paramètres optimisés du réseau de neurones

Par analogie, avec la modélisation du CHS « chapitre III » on procède à la modélisation du modèle inverse « capteur intelligent ». A la fin de la phase d'apprentissage et d'optimisation on obtient une architecture semblable à celle du modèle ANN. L'algorithme d'apprentissage et l'optimisation ont été étudiés au chapitre III.

Nous avons utilisé un programme Matlab pour l'apprentissage qui nous donne les matrices des paramètres du réseau de neurones « bias et poids ».

IV.2. Modèle INV-ANN du CHS sur PSPICE

Les matrices des bias et des poids nous permet d'implanter le modèle INV-ANN sur le simulateur PSPICE, utilisant les boîtes ABM, le modèle INV-ANN du CHS possède 10 neurones, chaque neurone du réseau est remplacé par une boîte ABM sur PSPICE. La figure IV.4 présente la structure du réseau des ABM du modèle inverse.

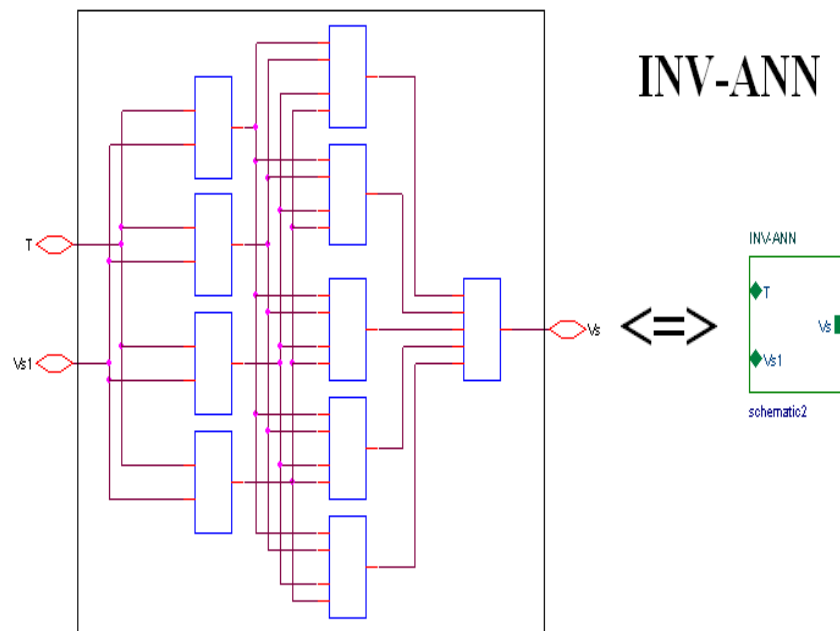


Fig.IV.4-Présentation de la structure des ABM de INV-ANN

IV.3. Modélisation du capteur intelligent

Ce modèle inverse présenté n'est rien qu'un correcteur de non linéarité provoqué par l'effet de la température. Cependant, ce modèle de correction « INV-ANN » couplé avec le modèle de CHS « ANN » et le convertisseur capacité/tension « CVC », nous donne le capteur intelligent désiré. Un schéma synoptique du capteur intelligent est montré par la figure IV.5.

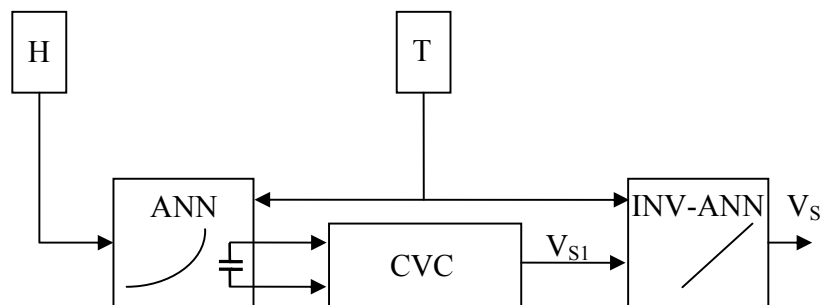


Fig.IV.4-Schéma synoptique du capteur intelligent

IV.4. Test du capteur intelligent

Le circuit pratique de test est représenté par la figure IV.5, afin d'évaluer les performances de notre capteur intelligent. Nous avons envisagé les différents cas, qui traitent les problèmes de la non linéarité et la dépendance à la température. Nous validons le modèle du capteur intelligent par un test pour des valeurs arbitraires attribués à l'humidité et à la température.

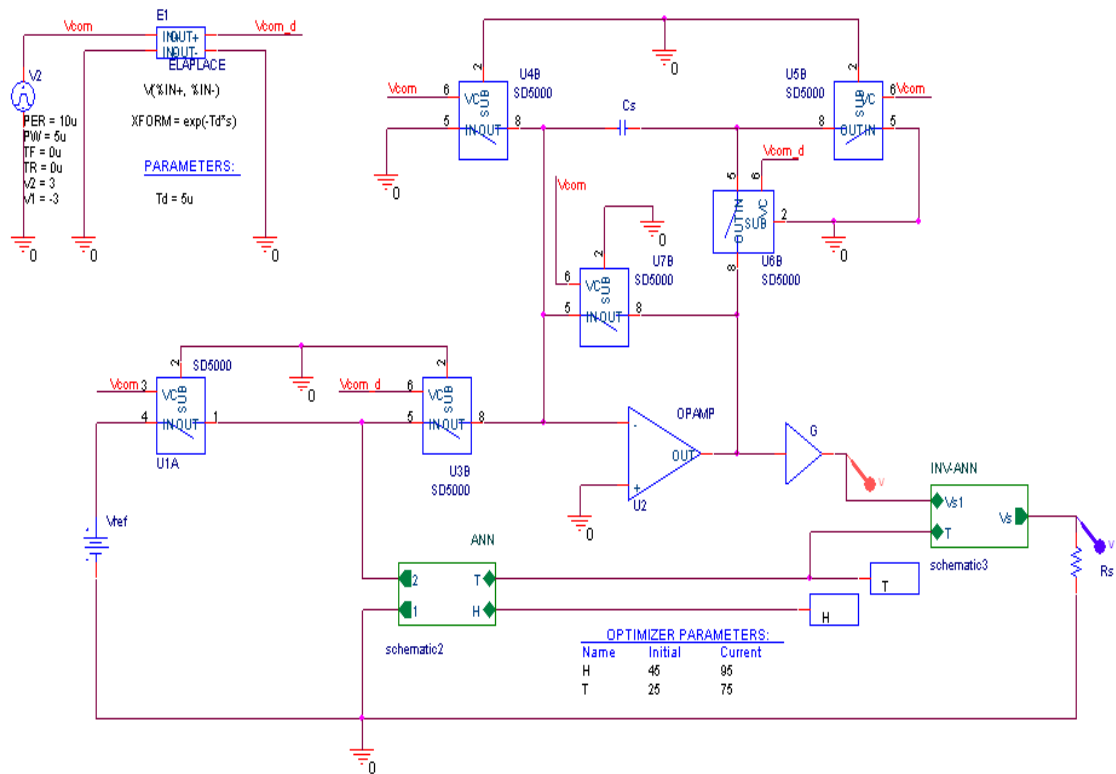


Fig.IV.5-Schéma pratique du capteur intelligent

IV.5. Test pour un cycle d'humidité

Le premier test est effectué pour montrer l'efficacité de la linéarisation appliquée au capteur CHS par le correcteur INV-ANN, pour cela la figure IV.6 illustre la réponse du capteur intelligent, pour une variation entre 45% et 95% à température fixe 25°C.

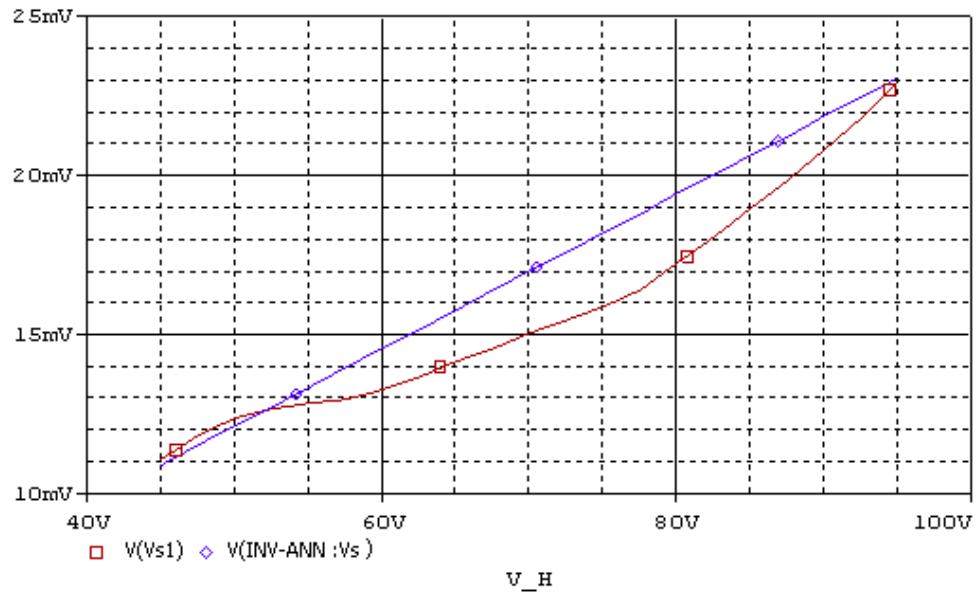


Fig IV.6 – Réponse du capteur intelligent à T=25°C pour un cycle d’humidité

La figure précédente montre une correction totale pour la température 25°C, la même réponse est aussi délivrée pour une simulation paramétrique pour les trois températures 25, 50 et 75°C. La figure IV.7 présente l’entrée V_{S1} et la sortie V_S du modèle inverse INV-ANN.

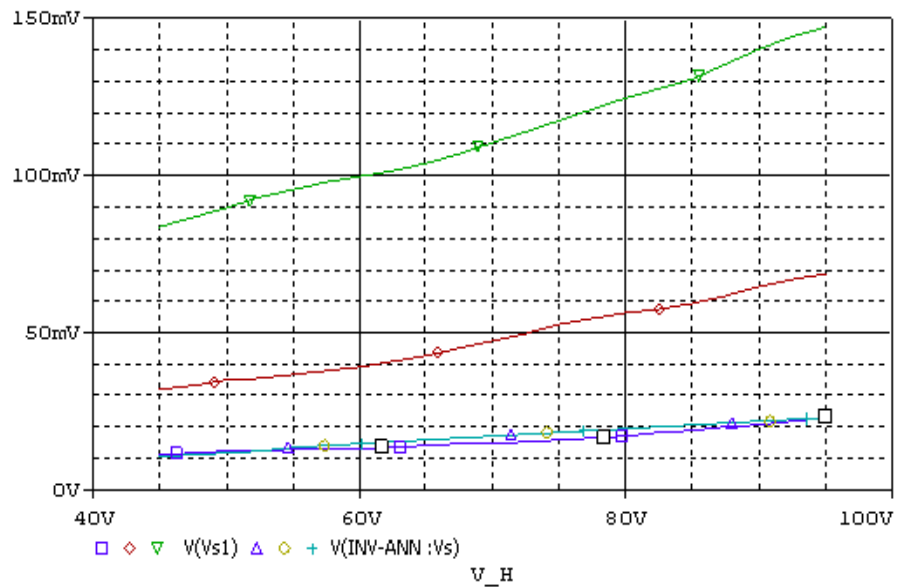


Fig IV.7 – Réponse du capteur intelligent pour une simulation paramétrique à T=25, 50 et 75°C pour un cycle d’humidité

Cette dernière simulation montre une élimination totale des courbes dues à la température, et leur remplacement par un seul signal de sortie V_S de INV-ANN.

IV.6. Test pour un cycle de température

Fixant maintenant l'humidité et on fait varier la température de 25°C à 75°C, si la réponse est stable, donc on peut dire que la variation de la température n'a aucun effet sur notre capteur. La figure IV.7 montre une simulation paramétrique pour deux valeurs d'humidité 45% et 60%.

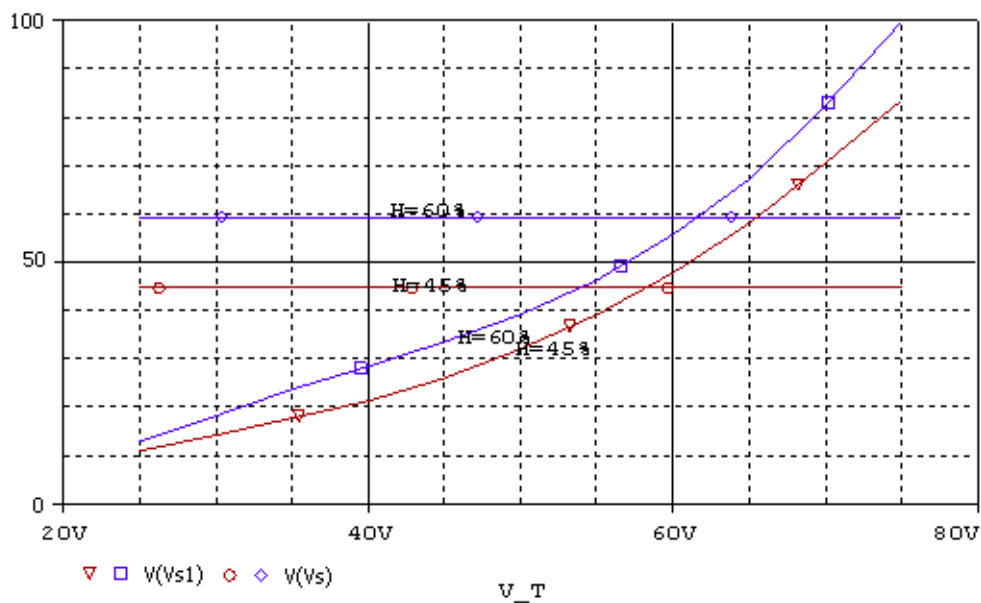


Fig IV.7 – Réponses de ANN et INV-ANN pour une simulation paramétrique pour les valeurs d'humidité 45% et 60%

La sortie est stable ce qui implique que la réponse du capteur est complètement indépendante de la température.

Conclusion

Les résultats de simulation obtenus, montrent que le modèle INV-ANN développé a permis de corriger la sortie du capteur CHS, tout en éliminant la non linéarité et la dépendance de la température. L'association de cet élément de correction, modélisé à base des ANNs a apporté des améliorations considérables sur le fonctionnement du CHS.

Conclusion générale

Conclusion générale

Les capteurs intelligents constituent un domaine de recherche de pointe de par leur fonction de monitoring et de contrôle de paramètres physico-chimiques et biologiques, dans tous les domaines de l'activité humaine. Dans le cadre d'applications de détection, d'analyse ou de surveillance. Leurs applications dans des domaines tels que la sécurité (systèmes d'identification et d'authentification), la gestion des biens, le bâtiment, le transport, la fabrication, la détection environnementale, les hydrocarbures, le textile, la maintenance prédictive industrielle, le monitoring médical, ouvrent de nouvelles opportunités de développement économique.

Le principe du capteur intelligent étudié repose sur un dispositif capacitif MEMS dont le diélectrique est constitué par l'air dont on veut connaître l'humidité. Une variation de cette capacité indique une variation d'humidité. Malheureusement, toute variation de la grandeur d'influence qui est la température induit une erreur de la réponse en plus de réponse non linéaire. Pour arriver à compenser toutes ses dérives, nous avons fait appel, dans ce mémoire, aux réseaux de neurones d'abord pour reproduire fidèlement la nature de la réponse du modèle du capteur d'humidité appelé modèle direct, que nous avons inséré dans la bibliothèque de PSPICE. Puis, en second lieu nous avons conçu un modèle inverse appelé aussi capteur intelligent qui permet d'éliminer toutes les imperfections que nous avons relevées. Comme perspective de ce travail, on peut imaginer l'intégration matérielle de ce modèle sur des FPGA ou un ASIC qui sera pourquoi pas commercialisé.

Bibliographie

Bibliographie

- [1] **Gert J.W. Visscher**
Measurement, Instrumentation, and Sensors Handbook
Chapitre 7 "Humidity and Moisture Measurement" CRC Press LLC(2000).
- [2] **N. M. White and J. D. Turner**
Thick film sensors: past, present and future
Measur. Sci. Technol, 8:pp.1-20 (1997).
- [3] **W. Qu. and J. U. Meyer**
A novel thick film ceramic humidity sensitive characteristics of α -Fe₂O₃ ceramic sensor
J. Am. Ceram. Soc., 75:pp.546-551 (1992).
- [4] **C. Laville and C. Pellet**
Comparison of three humidity sensors for a pulmonary function diagnosis microsystem
IEEE Sensors J., n°2, 2002, pp.96-101.
- [5] **M. Dokmeci and K. Najafi, 2001**
A high-sensitivity polyimide capacitive relative humidity sensor for monitoring anodically bonded hermetic micropackages.
Microelectromechanical Systems (JMEMS), 10:197-204.
- [6] **C. D. Ahrens**
Meteorology today
West Publishing Company, St. Paul, MN (1985).
- [7] **J. Crank,**
The mathematics of diffusion, 2nd ed, Clarendon Press, Oxford, Chapter 4:44-69 (1975).
- [8] **Chia-Yen Lee and Gwo-Bin Lee**
MEMS-based Humidity Sensors with Integrated Temperature Sensors for Signal Drift Compensation 0-7803-8313 -5/03/2003 IEEE
- [9] **Aboubacar CHAEHOI**
Conception et modélisation de MEMS monolithiques CMOS en technologie FSBM
UNIVERSITE MONTPELLIER II « 2005 »
- [10] **Hafiane LAMINE**
Conception d'un capteur de pression intelligent
UNIVERSITE DE BATNA « 2005 »
- [11] **Claude TOUZET**
LES RESEAUX DE NEURONES ARTIFICIELS INTRODUCTION AU CONNEXIONNISME
Juillet 1992
- [12] **Marc Parizeau**
RESEAUX DE NEURONES
UNIVERSITE DE LAVAL Automne 2004
- [13] **Fabien Moutarde**
Introduction aux réseaux de neurones
Ecole des Mines de Paris, Avril 2007
- [14] **Rémi COULOM**
Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur
INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE « 2002 »
- [15] **Gert van der Horn, Johan H. Huijsing**
Inegrated Smart Sensors: design and calibration

KLUWER ACADIMIC PUBLISHER

1998

[16] **Won Young Yang, Wenwu Cao, and Tae-Sang Chung and John Morris**

APPLIED NUMERICAL METHODS USING MATLAB

John Wiley & Sons, Inc2005

[17] **Paul BOURRET, James REGGIA, Manuel SAMUELIDES**

Réseaux neuronaux une approche connexionniste de l'intelligence artificielle,

Ouvrage de l'édition TEKNEA, 1991

[18] **Kodjo Agbossou, T.James Agbebi, Demagna Koffi et Mohammed Elhiri**

Conception d'un capteur intelligent pour la détection des vapeurs de styrène dans l'industrie

J. Phys.III France 4 2031-2045 « 1994 »

[19] **ARPAIA, P. – DAPONTE, P. – GRIMALDI, and D. – MICHAELI, L.**

ANN-Based Error Reduction for Experimentally Modelled Sensors,

IEEE Trans. on Instrumentation and Measurement, vol. 51, no. 1, pp. 23-30, (2002).

Résumé

Ce travail vise à concevoir un capteur d'humidité intelligent, dont l'objectif est d'éliminer, la non linéarité et la sensibilité multiple, de la sortie du capteur utilisé. Ce capteur d'humidité est de type MEMS capacitif. A l'aide des réseaux de neurone et l'environnement Matlab on est passé par la phase d'apprentissage pour concevoir un modèle ANN. Un modèle de ce composant a été inséré à la bibliothèque du simulateur PSPICE, dont la sortie reproduit fidèlement le comportement du capteur d'humidité MEMS utilisée. Du fait que notre capteur est de type capacitif, le modèle obtenu sur PSPICE traduit la variation de l'humidité par une variation d'une capacité. Cette dernière est une grandeur passive qui nécessite une conversion en une grandeur active. A cet égard nous avons réalisé un circuit de conversion capacité/tension. Une linéarisation, par un programme Matlab, est appliquée à la réponse de l'ANN dont le but est de créer une base de donnée d'un élément de correction permettant de corriger sa réponse non linéaire. L'apprentissage pour cette nouvelle base de donnée nous donne le modèle inverse INV-ANN. Les trois blocs, modèle ANN, modèle inverse INV-ANN et le convertisseur capacité tension, constituent le capteur intelligent.

Mots clés :

Capteur d'humidité, Modèle direct, Modèle inverse, Réseaux de neurones, MLP, MEMS.

Abstract

This work aims to achieve a design of a smart humidity sensor, the goal of this study is to eliminate the non-linearity and the cross sensitivity of the output sensor used. The humidity sensor is a MEMS Capacitive kind. Using neuronal networks and Matlab environment we have done the training to design an analytical model ANN and create a model for this component in the PSPICE simulator library, where the output of this model is identical to the output of the MEMS humidity sensor used. Because our sensor is a capacitive type, the obtained model on PSPICE reflects the humidity variation by a capacity variation, which is a passive magnitude; it requires a conversion to an active magnitude, why we realize a conversion capacity/voltage circuit. A linearization, by a Matlab program, is applied to ANN response whose goal is to create a database for an element of correction, can correct its nonlinear response. The training for the new database provides us with the inverse model INV-ANN. The three blocks; ANN model, model reverse INV-ANN and the capacity/ voltage converter, represent the smart sensor.

Keys words:

Humidity Sensor, Direct Model, Inverse Model, Neuronal Network, MLP, MEMS.