



وزارة التعليم العالي و البحث العلمي

Ministère l'Enseignement Supérieur et de la Recherche Scientifique

جامعة عبد الحميد ابن باديس

Université Abdelhamid Ibn Badis de Mostaganem

كلية العلوم و التكنولوجيا

Faculté des Sciences et de la Technologie

N° d'ordre :/GM/2020



MEMOIRE DE FIN D'ETUDUE

MASTER EN GENIE ELECTRIQUE

Filière : Génie électrotechnique

Spécialité : électrotechnique industrielle

Par

Belkiseria Oussama

Modélisation neuronale d'un système électrique LIT

Soutenue le : .../06/2021

Devant le Jury :

Président	REZINI SALIHA	MCB -Mostaganem
Encadreur	YAGOUBI BENABDELLAH	PROF -Mostaganem
Examineur	DAOUD MOHAMED	MCA -Mostaganem

Anne universitaire 2020 - 2021

Dédicaces

Tout d'abord, nous remercions Allah de nous avoir donné le courage de terminer ce modeste travail. Nous remettons ce mémoire en cadeau à nos chères familles pour nous avoir poussé et motivé dans nos études, ce projet de fin d'études représente le fruit de leurs soutient et encouragements, qu'ils nous ont prodigués tout au long de nos études, qu'ils soient remerciés par cette modeste dédicace Enfin à nos chers amis pour nous avoir soutenus

REMERCIEMENTS

Nous aimerions profiter de cette occasion pour remercier, M.Yagoubi Benabdellah, pour son inspiration et sa direction continue tout au long de notre travail. Ses conseils nous ont aidés tout au long de la recherche et de l'écriture de ce mémoire.. Nous adressons nos remerciements aux enseignants de jury D'avoir accepté d'examiner notre modeste travail. Nous voudrions également exprimer notre gratitude à nos Enseignants du département de Génie électrique qui par leurs expériences, savoir et leurs enthousiasmes nous ont beaucoup aidé à nous développer autant que personnes et à connaître à vrais notre chemin professionnel. Et remerciements particuliers à Mme Wahiba Chali

Nous adressons enfin nos vifs remerciements à toutes personnes qui ont contribué à la réussite de notre travail.

Résumé

La modélisation neuronale est l'un des outils les plus importants dans pratiquement tous les domaines scientifiques et, en particulier, en intelligence artificielle. Les chercheurs dans le domaine de la modélisation, utilisent souvent les réseaux de neurones pour représenter les modèles réels. Dans ce travail, nous avons abordé la modélisation d'un filtre passe bande RIF par un réseau de neurones représenté par une seule couche interne de 15 neurones et une seule sortie avec un neurone. Nous avons choisi d'utiliser la fonction tangente hyperbolique comme fonction d'activation à cause d'une part de ces valeurs continues entre -1 et 1, et d'autre part de sa dérivabilité qui permet d'appliquer l'algorithme de la retro-propagation en se basant sur le gradient. Après avoir testé notre modèle neuronale en commençant par un neurone et une fonction d'activation linéaire puis ajouté au fur et à mesure des neurones ainsi que changer la fonction d'activation linéaire par une hyperbolique, on a abouti à concevoir le modèle neuronale désiré.

ملخص

تعد النمذجة العصبية واحدة من أهم الأدوات في جميع مجالات العلوم تقريبًا ، وعلى وجه الخصوص ، في الذكاء الاصطناعي. غالبًا ما يستخدم الباحثون في مجال النمذجة الشبكات العصبية لتمثيل نماذج حقيقية. في هذا العمل ، اقتربنا من نمذجة مرشح النطاق الترددي **FIR** بواسطة شبكة عصبية ممثلة بطبقة داخلية واحدة من **15** خلية عصبية ومخرج واحد مع خلية عصبية. لقد اخترنا استخدام دالة الظل الزائدية كوظيفة تنشيط لأنه من ناحية هذه القيم المستمرة بين **-1** و **1** ، ومن ناحية أخرى لاشتقاقها مما يجعل من الممكن تطبيق خوارزمية الرجعية - التكاثر على أساس التدرج. بعد اختبار نموذجنا العصبي بالبدء بخلايا عصبية ووظيفة تنشيط خطية ، ثم إضافة الخلايا العصبية أثناء تقدمنا وكذلك تغيير وظيفة التنشيط الخطي إلى وظيفة قطعية ، تمكنا من تصميم النموذج العصبي المطلوب.

Abstract :

Neural modeling is one of the most important tools in virtually every field of science and, in particular, in artificial intelligence. Researchers in the field of modeling often use neural networks to represent real models. In this work, we approached the modeling of an FIR bandpass filter by a neural network represented by a single internal layer of 15 neurons and a single output with a neuron. We have chosen to use the hyperbolic tangent function as an activation function because on the one hand these continuous values between -1 and 1, and on the other hand its derivability which allows to apply the back-propagation algorithm based on the gradient. After testing our neural model by starting with a neuron and a linear activation function, then adding neurons as we went along and replacing the linear activation function with a hyperbolic function, we were able to design the neural model wish.

Mots cle:

Intelligence artificielle- les réseaux de neurones- filtre passe bande

LISTE DES TABLEAUX

Chapitre 1

Table 1.2: déplacement et moment généralisé 15

Chapitre 2

Table 2.1: tableau de strejc 18

LISTE DES FIGURES

Chapitre 1

Figure 1.1:neuron biologique.....	5
Figure1.2: apprentissage supervise	6
Figure1.3:principe de l'apprentissage non supervise	7
Figure1. 4:perceptron multicouche	7
Figure 1.5: information fournies par une représentation bond graph.....	9
Figure1.6:transfer de puissance de A vers B.....	9
Figure 1.7: circuit électriques en série.....	10
Figure 1.8: modélisation en bond graph.....	11
Figure 1.9:circuit RL en parallèle.....	12

Chapitre 2

Figure 2.1 : réponse indicielle du procède.....	19
Figure 2.2:reponse indicielle du procède	21

Chapitre 3

Figure 3.1:resau de perceptron multi couche	25
Figure 3.2 : Représentation temporelle et fréquentielle du signal d'entrée en haut, et celles du signal désiré en bas.	28
Figure 3.4 : filtre passe bande de fréquence centrale 0,2Hz et de largeur de bande B=0,1Hz	29
Figure 3.5 : filtrage par un passe bande	29
Figure 3.6 : architecture formule de neurone	30
Figure 3.7 : Exemple de perceptron multicouche	31
Figure 3.8 : fonction tanh(x)	31
Figure 3.9 : filtrage neuronale pour un nombre de neurones 5.....	34
Figure 3.10 : A gauche : Représentation fréquentielle et temporelle du signal de sortie neuronale avec 10 neurones, et à droite : celles du signal désiré.	35
Figure 3.11 : A gauche : Représentation fréquentielle et temporelle du signal de sortie neuronale avec 15 neurones dans la couche interne, et à droite : celles du signal désiré.....	36

SOMMAIR

Chapitre 1

1.1	Définition du système :	3
1.2	Principe et Définition d'un réseau de neurones (RN) :	3
1.2.1	Historique :	3
1.2.2	Le neurone biologique et le neurone artificiel :	4
1.3	Classification selon l'apprentissage	5
1.3.1	Apprentissage supervisé :	6
1.3.2	Apprentissage non supervisé :	6
1.3.3	Perceptron multicouches (MLP) :	7
1.4	Principes de modélisation en bond graph :	8
1.4.1	Concepts et définitions : port, lien, bond graph :	8
1.4.2	Le formalisme Bond Graph :	9
1.5	La construction de bond graph :	10
1.5.1	Procédure pour la modélisation des systèmes électriques :	13
1.6	La causalité :	13
1.6.1	Règles de causalité :	14
1.6.2	Rôle de la causalité :	14
1.7	Les variables mises en jeu :	14
1.7.1	Éléments BG :	15
1.8	Les jonctions :	16
1.9	Conclusion :	16

Chapitre 2

2	Introduction :	17
2.1	Méthodes d'identification du modèle paramétrique : [15, 16, 17,18]	17
2.1.1	Méthode de (Strejc) : [15]	18
2.1.2	Méthode de Broida : [16]	20

Chapitre 3

3	Introduction :	21
3.1	L'approche neurale — réseaux de neurones :	22
3.2	Choix de l'architecture du réseau :	23
	Structure monocouche :	23
	Structure multi couche :	24

3.3	Filtres LIT numériques fondamentaux.....	25
3.3.1	Résultats et discussion	27
3.3.2	Application du passe bande.....	28
3.4	Modèle neuronale.....	30
3.4.1	Les caractéristiques neuronales :	30
3.4.2	Représentation d'un neurone formel :.....	30
3.5	Choix de l'algorithme d'apprentissage :	32
3.5.1	Algorithme LMS (back propagation) :.....	33
3.5.2	Application du réseau de neurones.....	33

Introduction générale

Les procédés énergétiques possèdent un comportement fortement non linéaire dû principalement à l'interaction mutuelle de plusieurs phénomènes de natures diverses et associant des composants technologiques qui mettent en œuvre des énergies de natures différentes (mécanique, chimique, thermodynamique,). Le comportement dynamique de ce type de système est décrit généralement par des équations différentielles non linéaires. La démarche classique basée sur les équations du premier principe (consistant à étudier les échanges en stationnaire ou en transitoire) devient compliquée en raison du caractère multi énergie et non stationnaire de ce type de procédé. L'outil bond graph (graphe de liaison ou graphe à liens) à vocation pluridisciplinaire permet justement par sa nature graphique, à l'aide d'un langage unifié, d'afficher explicitement la nature des échanges de puissance dans le système, tels que les phénomènes de stockage, de transformation et de dissipation d'énergie et de mettre en évidence la nature physique et la localisation des variables d'état. Par ailleurs, le modèle bond graph est évolutif, ce qui permet aisément d'affiner le modèle (en fonction des hypothèses de modélisation retenues) par simple ajout de nouveaux éléments (perte thermique, effet d'inertie, etc.) sans avoir à reprendre la démarche depuis le début. La méthode est adaptée à l'obtention de modèles intégrés, c'est pourquoi l'intérêt de la modélisation par bond graph des phénomènes rencontrés en génie énergétique et en génie des procédés est évident. L'outil bond graph a été défini initialement en 1959 au MIT (Boston, USA) par Paynter [2], puis introduit en Europe uniquement vers les années 1970. Cette démarche de modélisation est très pédagogique pour la compréhension et l'analyse de la dynamique des systèmes. La méthodologie a été développée à partir de 1966 principalement par Karnopp, Rosenberg et Thoma [3]. La modélisation par bond graphs des systèmes à structure rigide (mécanique, électrique) a connu d'énormes développements ; par contre, la modélisation des systèmes thermiques, thermodynamiques ou tout simplement en génie énergétique et des procédés reste encore un domaine ouvert en raison de la complexité de ces phénomènes. Ce sont pourtant ces types de processus, présents dans les industries à risque qui nécessitent pour leur connaissance et leur contrôle des modèles de plus en plus précis et exploitables. Ce langage répond particulièrement bien à ces besoins par les caractéristiques suivantes :

- une approche énergétique qui permet une décomposition du système étudié en sous-systèmes qui échangent de la puissance et qui structurent la procédure de modélisation,
- une terminologie unifiée pour tous les domaines physiques, fondée sur la notion d'analogie entre phénomènes,
- une représentation graphique pour visualiser les transferts de puissance, mais aussi de causalité,
- une souplesse inhérente qui permet de faire évoluer le modèle en ajoutant des phénomènes négligés

- une écriture systématique des équations mathématiques issues du modèle bond graph sous forme d'équations différentielles,
- un support pour une analyse structurelle des propriétés du modèle en vue de la conception de systèmes de surveillance ou de commande [4].

L'évolution technologique durant les dernières années a permis aux scientifiques d'élaborer et de perfectionner des méthodes pour différents domaines. L'évolution des ordinateurs en particulier et la capacité d'intégration de composants formidable atteintes à nos jours ont permis une grande vitesse de calcul et une grande capacité mémoire. Parmi ces méthodes, il existe une méthode qui est utilisée dans plusieurs domaines de recherches et de différentes manières, ainsi elle peut être utilisée d'une manière complètement soft en utilisant uniquement l'ordinateur ou d'une manière hard en utilisant les circuits intégrés. Cette méthode est celle des réseaux de neurones artificiels (RNA). Les réseaux de neurones artificiels sont des outils puissant capables d'être utilisés dans près que tous les domaines technologiques, et on peut citer : le traitement du signal, vision, parole, prévision, modélisation, aide à la décision, robotique, évaluation des écosystèmes, identification des bactéries, commande des processus, modélisation des systèmes physiques reconnaissance des formes, mesure, instrumentation,[1].

Dans le chapitre suivant, nous allons donc pouvoir passer en revue des méthodes d'apprentissage statistique appliquées à la classification et la régression. Après une brève introduction, où nous allons présenter la notion de modèle biologique et neurone formel, nous décrivons les réseaux de neurones artificiels, modélisation générale, l'architecture des réseaux, les types d'apprentissage des réseaux de neurones. Nous rappelons les réseaux de neurones les plus utilisés (Perceptron Multi Couches (PMC)), ainsi que l'algorithme de retro-propagation du gradient et formalisation d'apprentissage. Enfin nous décrivons les différentes applications de RNAs.

CHAPITRE 01 : modélisation d'un système électrique

1.1 Définition du système :

Sous l'hypothèse que le système est unidirectionnel, la première étape consiste à différencier les causes (entrées) des effets (sorties) et à distinguer les grandeurs pertinentes des perturbations. A la figure 1.1 nous adoptons la représentation d'un procédé et des informations dont nous disposons pour sa description sous la forme d'un système S avec des entrées-sorties

1.2 Principe et Définition d'un réseau de neurones (RN) :

Est un système d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sortie, ces (RN) sont une métaphore des structures cérébrales et de traitement parallèle et distribué d'information et comportent plusieurs élément de traitement appelé neurone.

Chaque neurone fonctionne indépendamment des autres de telle sorte que l'ensemble est un système parallèle fortement interconnecté.

L'information détenue par le réseau de neurone est distribuée à travers l'ensemble des constituants et non localisée dans une partie de mémoire sous la forme d'un symbole.

Le réseau de neurone ne se programme jamais pour réaliser une ou telle tâche, il est entraîné sur des données acquises, grâce à mécanisme d'apprentissage qui agit sur les constituants du réseau afin de réaliser au mieux la tâche souhaitée.

On peut comparer la correspondance entre les propriétés respectives de neurones biologiques et neurones artificiels

1.2.1 Historique :

L'origine de l'inspiration des réseaux de neurones artificiels remonte à 1890 où W. James[5], célèbre psychologue américain, introduit le concept de mémoire associative. Il propose ce qui deviendra une loi de fonctionnement pour l'apprentissage des réseaux de neurones, connue plus tard sous le nom de loi de Hebb. Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W.

McCulloch et W. Pitts du neurone formel [6]. Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes. C'est ensuite que D. Hebb, physiologiste américain, présente en 1949 les propriétés des neurones par le conditionnement chez l'animal. Ainsi, un conditionnement de type pavlovien tel que, nourrir tous les jours à la même heure un chien, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose, explique en partie ce type de résultats expérimentaux. Les premiers succès de cette discipline remontent à 1957, lorsque F. Rosenblatt développe le modèle du Perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance des formes. Notons qu'à cette époque les moyens à sa disposition étaient limités et c'était une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes. C'est alors qu'en 1960, l'automaticien Widrow développe le modèle Adeline (Adaptative Linéairement). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétro-propagation de gradient très utilisé aujourd'hui 1969 : M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques avec les Perceptrons Multi Couches. M. Minsky et S. Papert publient ensuite en 1969 un ouvrage qui met en évidence les limitations théoriques du Perceptron. Ces limitations concernent l'impossibilité de traiter des problèmes non linéaires en utilisant ce modèle. Quelques années d'ombre se sont ensuite succédé de 1967 à 1982.

Le renouveau de cette discipline reprend en 1982 grâce à J. J. Hopfield, un physicien reconnu

. Il présente une théorie du fonctionnement et des possibilités des réseaux de neurones.

1.2.2 Le neurone biologique et le neurone artificiel :

Le premier neurone réalisé remonte à 1943 quand MM. Mac Culloch et Pitts, ont fait une tentative de modélisation mathématique du cerveau humain, ils ont réalisé un neurone avec une sortie binaire. Ils ont démontré théoriquement qu'un réseau de ces neurones formels peut reproduire des fonctions logiques et mathématiques. [7] Un neurone est une cellule du système nerveux, il est composé de :

- Les dendrites : ce sont des réseaux réceptifs arborescents de fibres nerveuses qui conduisent les signaux électriques dans la cellule.
- Le corps cellulaire : il somme efficacement les signaux reçus et en fixe le seuil.

- L'axone : c'est une simple fibre longue qui conduit le signal de la cellule aux autres neurones. Le point de contact entre un axone d'une cellule et une dendrite d'une autre cellule est appelé synapse. C'est l'arrangement des neurones et des forces des synapses individuelles, déterminées par un processus chimique complexe qui établit la fonction effectuée par le réseau de neurones.

La Figure 1.1 illustre un schéma d'un neurone biologique :

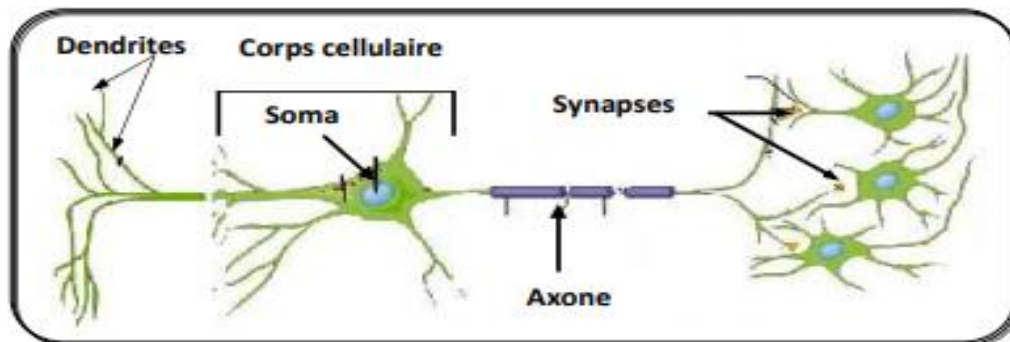


Figure 1.1:neurone biologique

1.3 Classification selon l'apprentissage

Apprentissage dans les réseaux de neurones :

L'apprentissage est une phase de développement d'un réseau de neurone, elle consiste à estimer les paramètres des neurones du réseau, à fin que celui-ci remplisse au mieux sa tâche qui lui est affectée. Pour cela il nécessite :

- Un ensemble d'exemples d'apprentissages :

En effet les réseaux de neurones sont des fonctions paramétrées, utilisées pour réaliser des modèles statistiques à partir d'exemples (dans le cas de la classification) ou de mesures (dans le cas de la modélisation) ; leurs paramètres sont calculés à partir de ces exemples ou couples {entrée, sortie}.

- La définition d'une fonction de coût qui mesure l'écart entre les sorties du réseau de neurones et les sorties désirées.

- Un algorithme de minimisation de la fonction de coût par rapport aux paramètres.

1.3.1 Apprentissage supervisé :

Dans ce cas, un superviseur (ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée), que le réseau de neurones doit associer à un vecteur d'entrée. L'apprentissage consiste dans ce cas à ajuster les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

Apprentissage semi-supervisé ou apprentissage par renforcement :

Ce mode d'apprentissage,

– Suppose qu'un comportement de référence n'est pas possible, mais en revanche, il est possible d'obtenir des indications qualitatives (vrai, faux, ...) sur les performances du réseau.

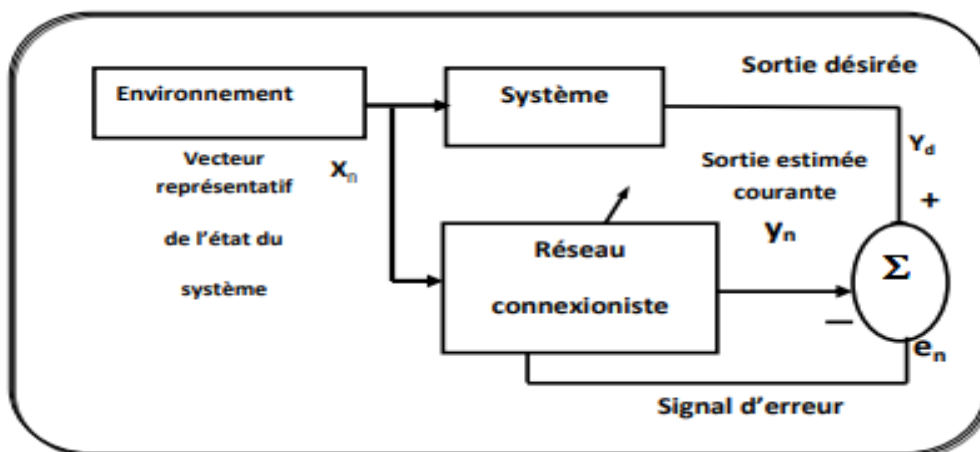


Figure1.2: apprentissage supervise

1.3.2 Apprentissage non supervisé :

L'apprentissage supervisé s'effectue sous le contrôle d'un expert, alors que l'apprentissage non supervisé est autodidacte. Les paramètres internes du réseau ne sont modifiés qu'avec les seuls stimuli, aucune réponse désirée n'est prise en considération. La Figure 1.8 montre que la sortie du réseau n'est pas utilisée par la procédure d'apprentissage

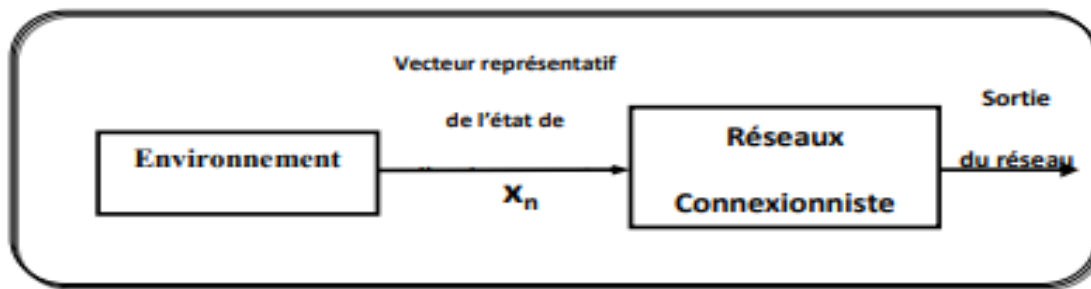


Figure1.3:principe de l'apprentissage non supervisé

Il faut souligner que l'efficacité de l'apprentissage augmente avec le nombre d'exemples d'apprentissage. Ceci est un facteur très important qu'il fallait noter : quelque soit l'algorithme choisi, la qualité de l'apprentissage des réseaux de neurones est d'autant meilleur que l'on dispose d'un ensemble d'apprentissage riche en exemples. On sait par ailleurs que la capacité de généralisation des réseaux de neurone nécessite également des exemples nombreux, qui de plus, doivent être bien distribués dans le domaine de validité souhaitée par le modèle.

1.3.3 Perceptron multicouches (MLP) :

Le perceptron multicouche MLP (Figure 1.4) est un réseau neuronal de type Feed-forward, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté les couches d'entrée et de sortie) et il n'y a ni connexions entre les neurones d'une même couche ni entre deux couches non successives. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions sigmoïdes.

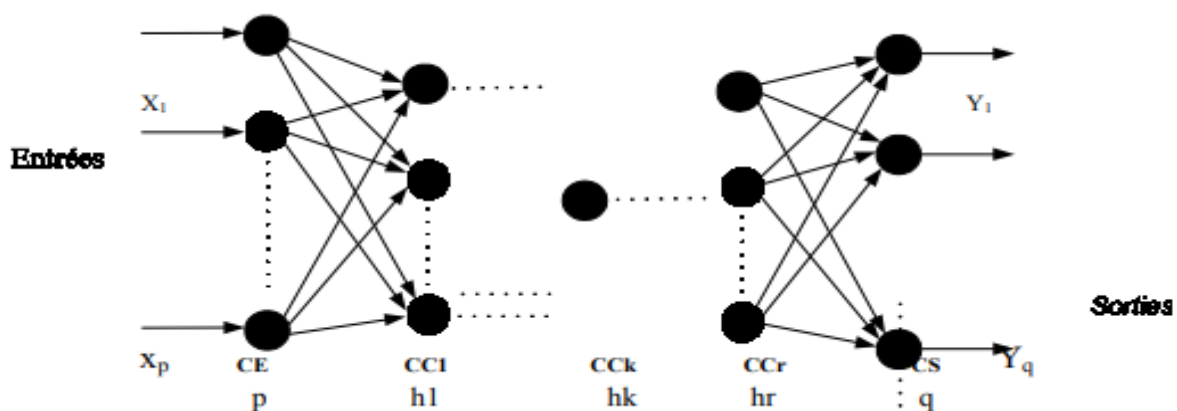


Figure1. 4:perceptron multicouche

CE : couche d'entrée.

CC : couche cachée.

CS : couche de sortie.

p : nombre de neurones d'entrée.

Q : nombre de neurones de sortie.

r : nombre de couches cachées.

h_k : nombre de neurones dans la couche cachée numéro k .

1.4 Principes de modélisation en bond graph :

Dans ce chapitre, nous présentons succinctement le formalisme Bond Graph (BG) comme un autre exemple de modélisation. La mise en perspective entre méthode BG et démarche de conception système en Génie Electrique opérera, tant au plan méthodologique (méthode de conception par la synthèse et l'analyse), que vis a vis des champs applicatifs (éléments de production et stockage électrochimique, systèmes de transport, énergies nouvelles,...) actuellement investis

1.4.1 Concepts et définitions : port, lien, bond graph :

Le bond graph est un outil de modélisation des systèmes dynamiques, il se base sur le transfert de puissance entre les différentes parties du système considéré, il décrit sa décomposition physique en sous-systèmes et en composants

L'ensemble de ces multiports connectés entre eux par des liens représente le bond graph ou « graphe de liaison » ou « graphe à liens ». Nous retiendrons le nom de bond graph pour des questions de commodité. D'une manière plus globale, le bond graph peut être défini comme un graphe linéaire où les nœuds sont les éléments multiports et les arcs les liens. Toutefois le bond graph est beaucoup plus riche que le graphe. En bond graph les liens (appelés liens de puissance) représentent la puissance ou le flux d'énergie échangé entre les deux multiports connectés. Cette puissance s'exprime comme le produit de deux variables complémentaires. Indépendamment du domaine physique considéré (thermique, électrique, chimique, ...), ces variables sont appelées « variables généralisées » d'effort et de flux notées respectivement par e et f . Ce lien (bond) porte les deux variables de puissance et est désigné par le symbole d'une demi-flèche comme indiqué sur la figure 1.5 Par convention, le flux est toujours représenté du côté de la demi-flèche. La puissance instantanée en fonction du temps t rentrant ou sortant du port est le produit effort flux (f)

$$\mathbf{P} = \mathbf{e} \cdot \mathbf{f}$$

La direction de la puissance positive est représentée par le sens de la demi-flèche. Une représentation bond graph permet de mettre en évidence quatre informations :

la liaison physique entre les deux systèmes par le lien bond graph, la causalité par le trait causal (développé plus loin), le type d'énergie utilisé par la nature des variables de puissance et enfin la direction de la puissance transmise par la direction de la demi flèche

Une représentation bond graph permet de mettre en évidence quatre informations [8]

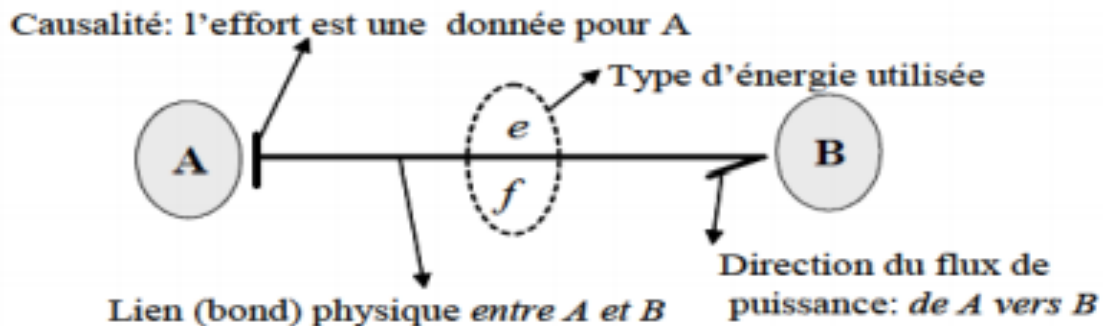


Figure 1.5: information fournies par une représentation bond graph

1.4.2 Le formalisme Bond Graph :

Le formalisme bond graph [9][10][11][12] a été introduit par H. Paynter en 1961 et formalisé par Karnopp et Rosenberg en 1975. Cette méthodologie est entrée en Europe à la fin de l'années 70 par les Pays-Bas (Université de Twente) et la France (société Alstom). L'outil bond graph est maintenant utilisé régulièrement dans quelques entreprises, en particulier dans l'industrie automobile (PSA, Renault, Ford, Toyota, General Motors,). Cette méthode illustre les transferts énergétiques dans le système en utilisant des liens de puissance (bond, en anglais). Un lien de puissance est symbolisé par une demi-flèche, dont l'orientation indique le sens de transfert de puissance. Ainsi la Figure 1. Représente le transfert de puissance du sous-système A vers le sous-système B

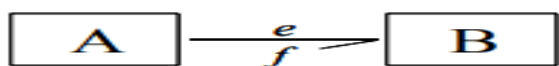


Figure 1.6: transfert de puissance de A vers B

L'outil bond graph comporte 4 étapes de modélisation [13] :

- **Technologique** : par le bond graph amonts.
- **Physique** : représentation des phénomènes physiques par les éléments qui lui convient (stockage, dissipation, transformation).
- **Mathématique** : déduction du modèle mathématique après l'affectation des causalités.
- **Algorithmique** : simulation et validation du modèle trouvé.

1.5 La construction de bond graph :

Exemple : Cas d'une jonction 0

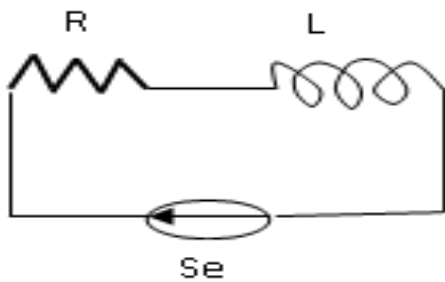


Figure 1.7: circuit électriques en série

Une jonction 0 à chaque nœud de potentiel et une jonction 1 entre deux jonctions 0. Ensuite on lie les jonctions ainsi que les éléments correspondants à chaque jonction 1.

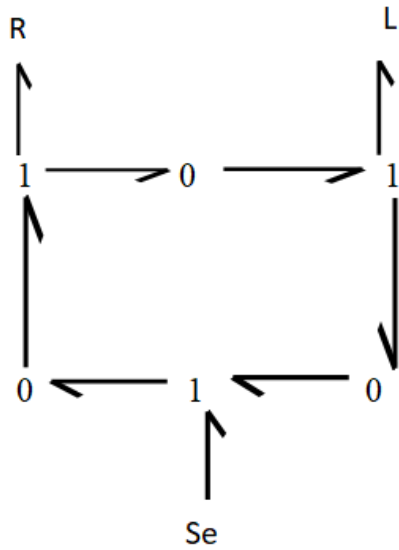
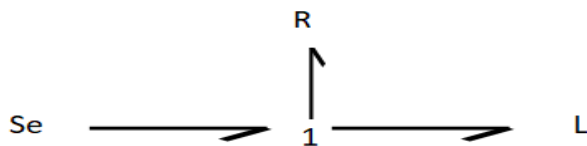


Figure 1.8: modélisation en bond graph

On choisit une jonction 0 comme référence et on la supprime ainsi que tous les liens qui lui sont attachés.

Simplifier :



Le résultat est donc une jonction 1 qui signifie que le même courant (flux) passe dans les éléments et par conséquent ces éléments sont en série.

$$f_S = f_R = f_L$$

Et

$$P_S - P_R - P_L = 0 \Rightarrow e_S - e_R - e_L = 0$$

Cas d'une jonction 0 :

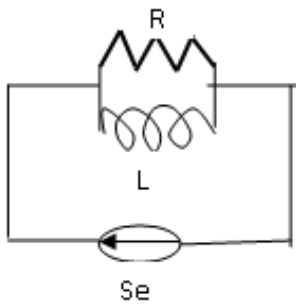
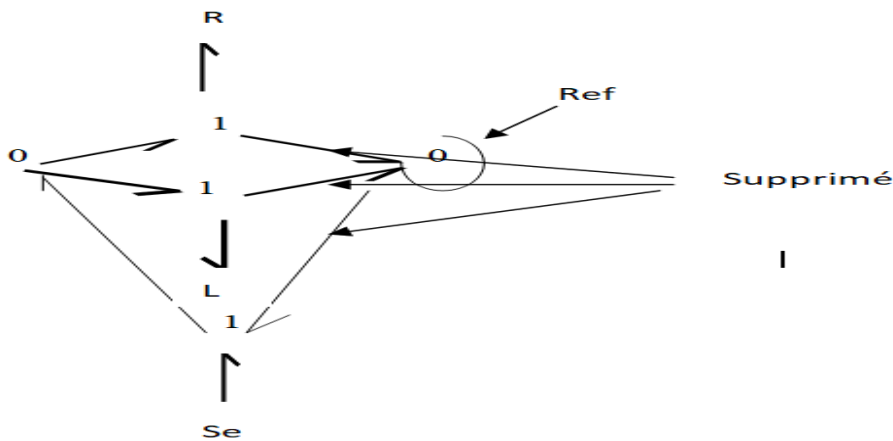


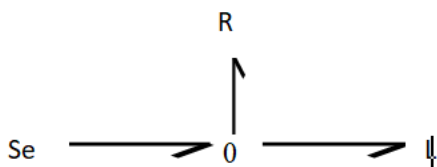
Figure 1.9:circuit RL en parallèle

Choisit une jonction 0 à chaque nœud de potentiel et une jonction 1 entre deux jonctions 0. Ensuite on lie les jonctions ainsi que les éléments correspondants à chaque jonction 1.

On choisit une jonction 0 comme référence et on la supprime ainsi que tous les liens qui lui sont attachés



Simplifier :



Le résultat est donc une jonction 0 qui signifie que la même tension (effort) est appliquée à chaque élément et par conséquent ces éléments sont en parallèles.

$$V_S = V_R = V_L$$

et

$$P_S - P_R - P_L = 0 \Rightarrow i_S - i_R - i_L = 0$$

1.5.1 Procédure pour la modélisation des systèmes électriques :

Pour la construction d'un BG on peut suivre une procédure propre au domaine concerné. Procédure pour les systèmes électriques :

1. Fixer un sens de circulation pour le courant. Il sera pris comme sens de transfert de la puissance.
2. Rechercher tous les nœuds de potentiels. Placer une jonction 0 par nœud.
3. Placer une jonction 1 entre deux jonctions 0 et y attacher les éléments soumis à la différence de potentiels correspondants.
4. Relier les jonctions par des liens, en respectant le sens de transfert de la puissance.
5. Choisir un nœud de référence (ou plusieurs suivant les cas) et supprimer la (ou les) jonction(s) 0 qui y est (sont) associée(s), ainsi que tous les liens qui sont attachés. Simplifier si possible :



Pour montrer chaque étape de la procédure pour les systèmes électriques on présente l'exemple basique d'un circuit

1.6 La causalité :

Nous abordons ici une des notions majeures des Bond Graph et, de façon plus générale, des systèmes énergétiques. En effet, ce paragraphe montre en résumé comment le respect d'une causalité physique (intégrale) garantit les lois fondamentales d'association d'éléments échangeant de la puissance. Nous verrons également comment l'étude des « chemins » suivis par la causalité donne de précieux enseignements en termes d'analyse modale, structurelle, et mène à la mise en forme mathématique des modèles (fonction de transfert, équation d'état,). Le travail d'analyse systémique (simplification de modèle, analyse de stabilité,), objet de ce mémoire, est en majeure partie fondé sur ce concept. Pour décrire les relations de "cause à effet" et écrire les lois associées aux éléments sous forme causale mathématiquement solvable, il faut d'abord déterminer les variables qui sont imposées aux éléments (les données) et celles qu'imposent ces éléments (les conséquences). Dans un bond graph, cette propriété est représentée par un trait orthogonal au lien, appelé trait causal.

- Causalité obligatoire pour les sources.
- Causalité obligatoire pour les éléments R avec des lois non inversibles.

- Causalité intégrale préférentielle pour les I et C, éléments de stockage d'énergie.

Le trait causal est placé près (respectivement loin) de l'élément ou jonction pour lequel l'effort (respectivement le flux) est une donnée

Le sens de la causalité n'a rien à voir avec celui du transfert de la puissance. Pour bien faire comprendre cette notion, l'exemple de la batterie d'accumulateur est tout à fait parlant pour l'électricien. En effet, que celui se charge (puissance reçue) ou de décharge (puissance fournie), un accumulateur idéal impose systématiquement sa tension (effort) au système environnant.

Pour bien distinguer l'assignation causale et l'égalité, nous utilisons des notations différentes

1.6.1 Règles de causalité :

-Un trait causal près de la jonction 0

- Causalité arbitraire pour les éléments R linéaires (dissipation d'énergie)

- Causalité intégrale préférentielle pour les I (inertie) et C (stockage d'énergie).

- Causalité obligatoire pour les sources.

Seul trait causal loin de la jonction 1

-Un seul

- Causalité obligatoire pour les éléments R avec des lois non inversibles.

1.6.2 Rôle de la causalité :

La causalité du bond graph permet d'établir la structure d'un modèle et d'en déduire les équations qui le régissent.

1.7 Les variables mises en jeu :

Les variables d'Energie (variables généralisées) :

Les variables généralisées sont introduites pour décrire l'état du système et sont définies par :

Le déplacement généralisé

L'impulsion généralisée

Variables d'puissance (Variables Conjugées) :

La puissance échangée P s'exprime comme le produit de deux variables complémentaires. Indépendamment du domaine considéré, ces variables sont appelées Les "variables généralisées" de tension et de flux, respectivement. Ces contrôles variables.

De puissance: $P=e.f$

Par convention le flux est toujours représenté du côté de la demi-flèche [14]

Le moment p (l'intégrale de l'effort par rapport au temps) et le déplacement q (l'intégrale du flux par rapport au temps).

Nous utilisons les éléments pour représenter des phénomènes qui lient les variables généralisées. Nous pouvons les séparer en trois catégories

1.7.1 Éléments BG :

On peut aussi généraliser la notion mécanique d'inertie (stockage d'énergie pour l'effort), (stockage d'énergie pour le déplacement généralisé) et résistance (dissipation d'énergie). Le tableau ci-dessous résume les différentes étapes de généralisation des éléments au cas électrique.

Table1. 1:déplacement et moment généralise

	Mécanique	Electrique	Notation unifiée (déplacement généralisé)
Déplacement x	$x = \int v.dt$ $v : vitesse \Leftrightarrow \mathbf{f} : flux$	$q = \int i.dt \quad (charge\ électrique)$ $i : courant \Leftrightarrow \mathbf{f} : flux$	$q = \int \mathbf{f}.dt$
Moment généralisé (quantité de mouvement)	$p = \int Fdt$ $F : force \Leftrightarrow \mathbf{e} : effort$	$p = \int Vdt$ $V : tension \Leftrightarrow \mathbf{e} : effort$	Notation unifiée (moment généralisé) (quantité de mouvement p)

1.8 Les jonctions :

Les Eléments de jonction notés $0,1$, servent à coupler un des éléments R,C , et les source, et composent la structure de jonction correspondant à l'architecture du modèle étudié. Ils sont conservatifs de puissance.

a. **Jonction0**: associe les éléments soumis à un même effort (**Fig.1.6**). Les relations qui la caractérisent sont :

Égalité des efforts pour tous les liens ayant une extrémité sur la jonction,
Somme algébrique des puissances = 0.

b. **Jonction1**: elle associe les éléments soumis à un même flux. Les relations, qui la caractérisent sont:

- Égalité des flux pour tous les liens ayant une extrémité sur la jonction,
- Somme algébrique des puissances = 0.

1.9 Conclusion :

On conclue ce chapitre par la remarque vue sur la richesse de l'outil bond graph en tant que support pour l'étude des systèmes complexes, de la modélisation à la simulation, de l'analyse à la synthèse des lois de commande. On a présenté dans ce chapitre l'outil bond graph et ses caractéristiques en mono-énergie (simple domaine de représentation). Toutefois, il est capable de représenter des phénomènes physiques où différents domaines physiques interagissent, il est appelé bond graph multi-énergie (le lien comporte plusieurs variables de puissance). Cependant, toutes les propriétés décrites précédemment (affectation de causalité, mise en équation, analyse, ... etc.) restent valables. Pour le bond graph multi-énergie il faut encore signaler l'existence du pseudo bond graph pour la modélisation des domaines physiques où le produit effort-flux n'a pas de sens physique comme dans l'hydraulique, la thermodynamique, et la chimie. Plus de détails sur ces concepts sera le sujet. Le chapitre suivant présente un état de l'art sur la surveillance des processus.

CHAPITRE2 : Identification neuronale d'un système électrique

2 Introduction :

L'identification de système est une technique de l'automatique consistant à proposer une structure entre sa commande et sa réponse afin de déterminer à partir des mesures des entrées/sorties, les valeurs des paramètres du modèle. Ainsi trouvé, le modèle doit être capable de reproduire aussi fidèlement que possible le système physique, chimique, biologique, économique L'informatique peut aider à exécuter la tâche d'identification par la génération de signaux test, l'acquisition des données et mesures, leur traitement en temps réel et enfin la validation du modèle. Un système linéaire peut être déterminé par plusieurs méthodes d'identification. Les premiers travaux ont été réalisés dans les années 1970 pour qu'elles puissent être appliquées à des procédés industriels. La procédure d'identification s'effectue essentiellement en deux étapes : la première consiste à fixer la forme des équations, c'est l'étape qualitative la seconde a pour but de trouver les valeurs numériques des coefficients qui interviennent dans ces équations c'est l'étape quantitative ou estimation des paramètres. Ces valeurs numériques sont déterminées pour que le comportement du modèle soit le plus proche de celui du système, cette approximation se mesure à l'aide d'un critère à choisir. Du point de vue historique, l'origine de la théorie sur l'estimation est créditée à Gauss, qui, en 1795 inventa la méthode des moindres carrés. Les premières études utilisant l'erreur quadratique dans le processus aléatoires sont dues à Kolmogorov, Krein et Wiener vers la fin des années 1963.[15] Nous étudierons dans le chapitre suivant les méthodes générales de l'identification du modèle non paramétrique et paramétrique des systèmes non bruités.

2.1 Méthodes d'identification du modèle paramétrique : [15, 16, 17,18]

Dans cette partie, on va étudier les méthodes utilisant le modèle paramétrique. Ce dernier consiste à caractériser un système par l'ensemble des paramètres qu'il convient de déterminer, ce qui revient à trouver le vecteur paramètres correspondant. Ces méthodes sont souvent utilisées dans l'industrie pour identifier les systèmes dynamiques. Il existe deux types :

Méthodes paramétriques graphiques (déterministes).

Méthodes paramétriques statistiques.

Méthodes paramétriques graphiques : [15 ,16]

Nous présentons successivement quelques méthodes fondamentales susceptibles d'obtenir un modèle paramétrique à partir du modèle non paramétrique

2.1.1 Méthode de (Strejc) : [15]

La méthode graphique de Strejc consiste à chercher un modèle mathématique d'ordre supérieur ou égale à $p=2$ avec excitation retardée qui représente le mieux le système réel continu (stable ; non résonant). La fonction de transfert d'un tel système dans l'espace de Laplace est donnée par

$$H(s) = g \cdot \frac{e^{-st_r}}{(1 + \tau \cdot s)^p} \quad 1$$

Les paramètres à identifier sont :

-g : le gain statique.

- r : le retard.

t: la constante de temps.

p : l'ordre

Table 2: tableau de Streijc

$P = c_1$	$\frac{t_1}{t_2} = c_2$	$\frac{t_1}{\tau} = c_3$	$\frac{t_2}{\tau} = c_4$
1	0	0	1
2	0.104	0.282	2.718
3	0.218	0.805	3.695

4	0.319	1.425	4.465
5	0.410	2.100	5.119
6	0.493	2.811	5.699
7	0.570	3.549	6.226
8	0.642	4.307	6.711
9	0.709	5.081	7.164
10	0.773	5.869	7.590

Pour identifier les paramètres du modèle à partir de la réponse indicielle du système, on trace la tangente au point d'inflexion Q. Cette dernière permet de déterminer T1 et T2

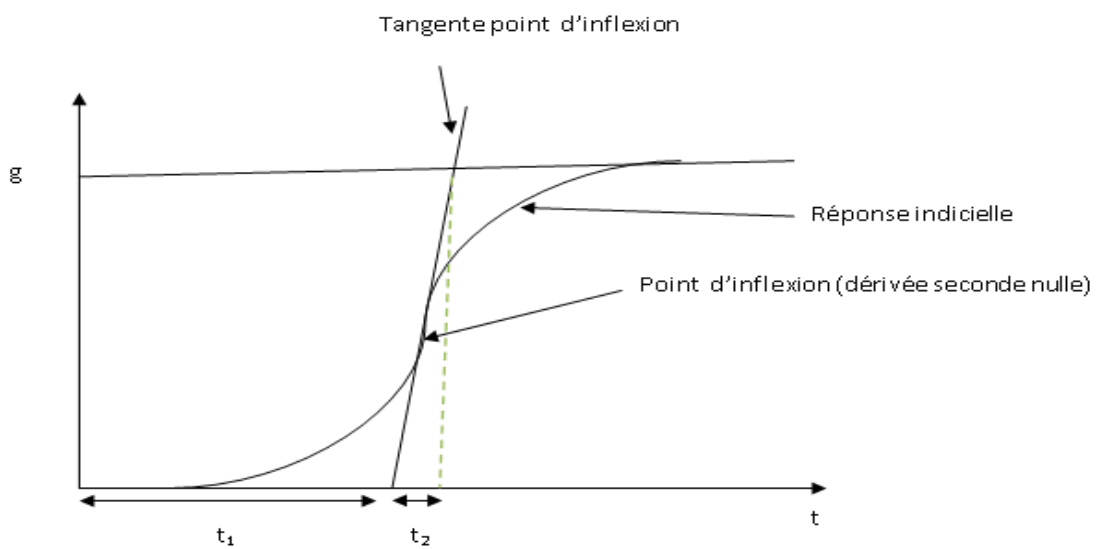


Figure 10 : réponse indicielle du procédé

Figure 2.1: réponse indicielle du procédé

Rapport $\frac{t_1}{t_2}$ de la colonne (c_2) (immédiatement inférieur à celui calculé)

-Le gain g est déterminé à t infini

Les autres coefficients sont déterminés à partir du graphe et du tableau (1) ci-dessous

-On trace la droite tangente au point d'inflexion pour déterminer t_1 et t_2 .

- L'ordre p correspond à celui du

-La constante du temps est $\tau = \frac{t_2}{c_4}$

-Le retard est $t_r = t_1(\text{mesuré}) - t_1(\text{tableau} = \tau * c_3)$

2.1.2 Méthode de Broida : [16]

Cette méthode consiste à chercher un modèle mathématique linéaire (eq. différentielle) du 1^{er} ordre avec excitation retardée qui représente le mieux le système réel linéaire (eq. différentielle) du 1^{er} ordre étudié. Ce modèle mathématique peut être décrit par

Et fonction de transfert

$$y(t) + \tau \frac{dy(t)}{dt} = g \cdot x(t - t_r) \quad 2$$

et de fonction de transfert

$$H(s) = g \cdot \frac{e^{-st_r}}{1 + \tau \cdot s} \quad 3$$

ns $\tan t$ (gain statique) et t_r retard (temps mort d'identification)

Détermination du modèle mathématique

Déterminer le modèle mathématique linéaire revient donc à déterminer ses coefficients constants (τ constante de temps, g (gain statique) et t_r retard) de l'équation correspondants à la meilleure approximation du système réel linéaire du 1^{er} ordre.

Pour cela on excite le système réel par un signal échelon défini par

On obtient la solution suivante pour la sortie (réponse indicielle) du système

$$y(t) = g \cdot \left(1 - \exp(-(t - t_r) / \tau) \right) \cdot e(t - t_r) \quad 4$$

Etude graphique de la réponse indicielle

A partir de ce graphe on peut déterminer les coefficients du modèle d'ordre 1, soient ;

A partir de ce graphe on peut déterminer les coefficients du modèle d'ordre 1, soient ;

$\tau = \text{constante de temps}$, $g = (\text{gain statique})$ et t_r

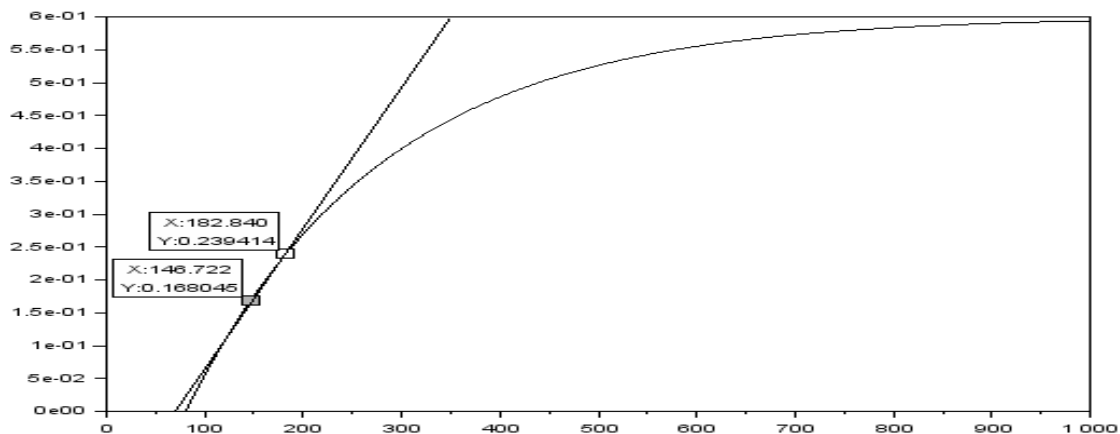


Figure 11:reponse indicielle du procédé

Chapitre 03 : Simulation et discussion

3 Introduction :

Le traitement d'un signal dynamique suscite beaucoup d'intérêt chez les chercheurs œuvrant dans le champ des réseaux de neurones. Il existe une littérature abondante sur ce sujet, notamment sur la reconnaissance du langage naturel et des séquences pour le contrôle d'un processus. Cependant, l'architecture utilisée pour traiter ce genre de problème est constituée principalement d'une structure récurrente. Car, cette forme permet de caractériser l'état d'une cellule de la couche de sortie au temps t en se basant sur l'information qui a précédé aux temps $(t-1)$ $(t-2)$, etc.. Cette caractérisation de l'état de la cellule en question permet éventuellement de déterminer son état futur. Parmi les algorithmes qui se sont les plus illustrés, on trouve les architectures de Jordan[20], de Paul Werbos [21] et de RyatoKamimura [22, 23]. Ce type de réseau de neurones est souvent associé à un traitement dynamique, traitement dont les paramètres varient dans le temps. Chacune des structures susmentionnées est unique dans son genre et permet d'effectuer

une tâche spécifique. Elles se caractérisent entre autres par des points forts et des points faibles. Parmi les avantages apparents des structures récurrentes, on constate que :

- la descente du gradient dans l'apprentissage est rapide, décroissante et stable;
- la convergence est quasi-absolue; — le temps de convergence est court;
- les états actuels des cellules de sorties sont mieux définis (connaissance des états précédents); — le gradient est guidé dans la recherche du minimum.

Parmi les inconvénients, on remarque:

- le besoin d'une grande capacité de mémoire pour stocker l'information: matrices des poids pour les états précédents;
- la faiblesse dans la généralisation ;
- la faiblesse à composer avec des données bruitées [24]

Pour ce qui est de l'autre type d'architecture à propagation-avant, elle est connue pour son traitement statique. C'est un traitement qui s'apparente au précédent mais qui n'utilise pas de rétroaction (feed-back). La majorité des applications trouvées dans la littérature sur les réseaux de neurones utilise ce type de réseau: l'architecture à propagation-avant. Ce modèle est caractérisé par une faiblesse dans le traitement dynamique (signal dont les paramètres varient dans le temps) de l'information étant donné sa nature statique, mais ceci ne fait pas l'unanimité auprès des scientifiques concernés. Une pléthore de travaux ont été effectués sur ce type de réseaux, particulièrement depuis l'apparition de l'algorithme de la rétro propagation [25]. La principale force de ce type de réseau réside dans sa capacité de discrimination et de classification

3.1 L'approche neurale — réseaux de neurones :

L'approche des réseaux de neurones est une technologie de traitement de l'information. On peut la voir comme une approche d'un niveau supérieur, utilisant des langages procéduraux comme outils de programmation (Basic, Fortran, C, etc..) dans le but de

Concevoir l'architecture du réseau, d'une part, et d'assumer son fonctionnement (apprentissage et reconnaissance) d'autre part. L'approche des réseaux de neurones n'utilise

pas un langage particulier comme c'est le cas pour la technologie des systèmes experts (lisp, Prolog, etc.. — programmation objet —); la simulation des réseaux de neurones s'exécute sur les mêmes systèmes d'ordinateurs (hardware). Cependant chacune de ces technologies celle des réseaux de neurones et des systèmes experts, possède une approche spécifique du traitement de l'information, ce qui en fait un meilleur outil, adapté à la résolution de problèmes particuliers. Le principe de fonctionnement des réseaux de neurones multicouches consiste à établir des relations intrinsèques entre les données fournies à l'entrée et à la sortie du réseau. En d'autres termes, le réseau doit apprendre la fonction F qui existe entre l'entrée et la sortie de celui-ci. Cette relation se trouve incarnée dans les données présentées lors de l'apprentissage.

Les applications des réseaux de neurones s'étendent à plusieurs champs, dont la médecine, la géologie, la robotique, l'inspection structurale, le contrôle de processus, l'inspection dans l'industrie manufacturière, l'analyse et le traitement de l'image, les télécommunications, le filtrage du signal (parole ou image) , etc..

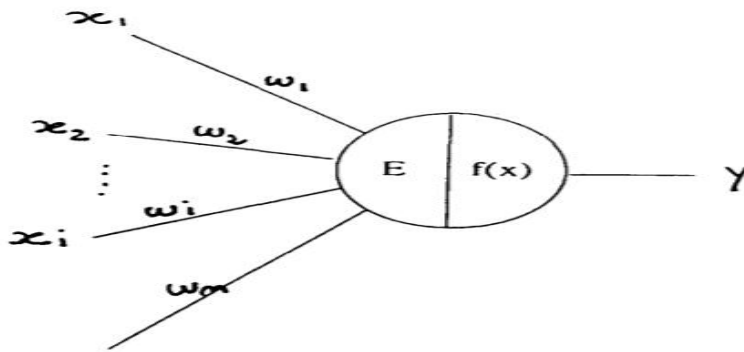
L'unité de base du traitement de l'information dans les réseaux de neurones est le neurone ou "Processing Élément" (unité de traitement). Les recherches sur le cerveau ont identifié au-delà de 300 types différents de neurones. L'unité de traitement citée précédemment en fait partie. Chaque variété de neurone se trouvant dans le réseau opère à sa manière et caractérise l'implantation d'une fonction spécifique.

3.2 Choix de l'architecture du réseau :

Structure monocouche :

C'est une architecture ayant un ensemble de n entrées (vecteurs d'entrée de dimension n), une cellule de discrimination (s), et un élément de sortie (y). Pour rendre ce réseau fonctionnel et dynamique, il faut un algorithme qui modifie les valeurs des connexions synaptiques (poids). Cet algorithme agit au niveau des connexions en changeant la valeur du lien entre deux ou plusieurs neurones. Ce type de réseau effectue un filtrage linéaire (suppression du seuil: $y=s$); il s'apparente à celui des filtres adaptatifs classiques. Les travaux de Bernard Widrow en sont un bon exemple. Lorsqu'on veut établir une classification avec ce type de réseau, il présente une limitation sur le type de données à traiter. La principale

contrainte réside dans son incapacité à effectuer une classification telle que les éléments soient non-linéairement séparables (ex: réaliser l'opération du "ou exclusif")



Structure multi couche :

Nous avons précédemment étudié les perceptrons (réseaux monocouche) et nous avons vu que les neurones de sortie étaient chacun connectés aux mêmes informations.

Nous les avons perçus comme une couche (alignés verticalement). Ainsi, une couche est constituée de neurones étant connectés aux mêmes informations mais n'étant pas connectés entre eux. Il s'agit maintenant de généraliser le perceptron. On peut ainsi disposer les neurones en plusieurs couches. Ainsi les informations en entrée sont connectés à tous les neurones de la première couche, tous les neurones de la première couche sont connectés à tous les neurones de la seconde couche, et ainsi de suite jusqu'à la dernière couche, appelée couche de sortie. Toutes les couches excepté la couche de sortie sont considérées comme "couches cachées". Toutefois, n'ayez crainte : il a été prouvé que dans la plupart des cas, un réseau à 2 couches (informations -> une couche -> couche de sortie) où chaque neurone de la couche cachée a comme fonction d'activation la fonction sigmoïde et chaque neurone de la couche de sortie a comme fonction d'activation une fonction linéaire permet d'approximer une fonction continue. Il s'agit du Théorème de Cybenko. Pour des fonctions discontinues, nous n'avons aucune garantie.

Du fait du résultat précédent, nous allons restreindre notre étude aux réseaux neuronaux à une seule couche cachée. La couche cachée permet plus d'interaction et instinctivement, on est conscient que notre réseau de neurone pourra apprendre des "fonctions" plus

complexes en rajoutant une couche. Le fonctionnement n'est pas pour autant complexe. On a toujours le choix de la fonction d'activation, et l'évaluation de la sortie d'un neurone se déroule de la même manière. La seule différence est que la sortie d'un neurone de la couche cachée sera l'une des informations d'entrées des neurones de la couche de sortie.

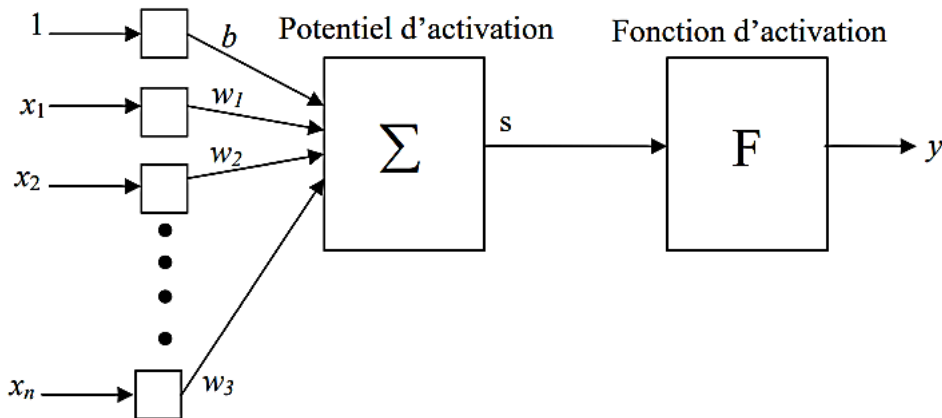


Figure 12:resau de perceptron multi couche

3.3 Filtres LIT numériques fondamentaux

Il existe quatre types de filtres fondamentaux selon la forme de leur réponses fréquentielles (RF) (ou réponse impulsionnelle RI).

a) Filtre passe bas idéal

Sa RF est définie par

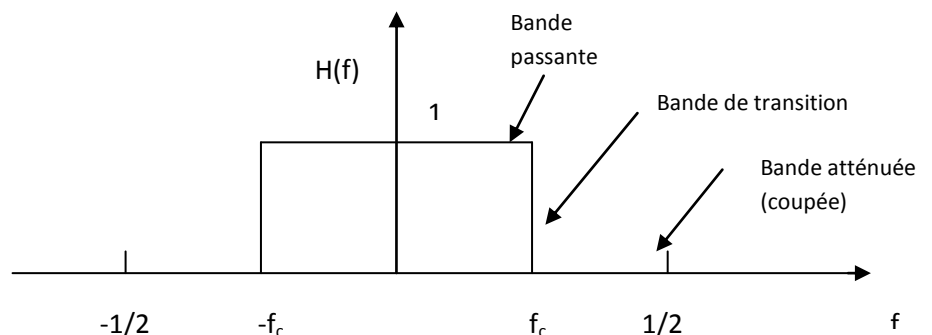
$$H(f) = \begin{cases} 1 & \text{si } |f| \leq f_c \\ 0 & \text{ailleurs} \end{cases} \quad 5$$

et sa RI est

$$h(n) = F^{-1}[H(f)] = \frac{\sin(2\pi f_c n)}{\pi n}$$

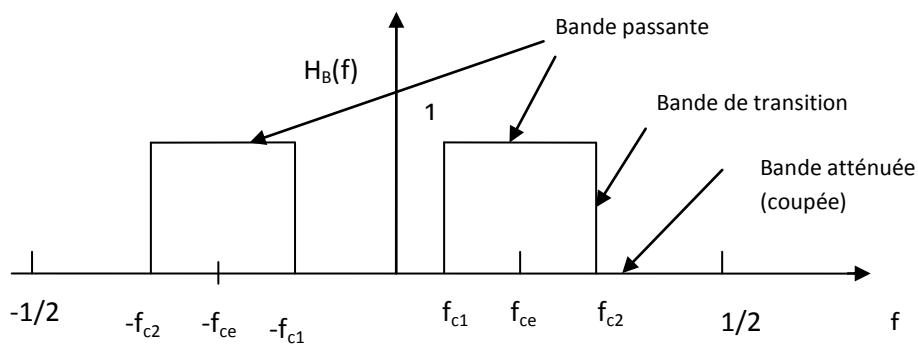
ou en posant $B = 2f_c$ largeur de bande du passe bas

$$h(n) = \frac{\sin(\pi B n)}{\pi n} \quad 6$$



b) Filtre passe bande

$$H_B(f) = \begin{cases} 1 & \text{si } f_{c1} \leq |f| \leq f_{c2} \\ 0 & \text{ailleurs} \end{cases} \quad 7$$



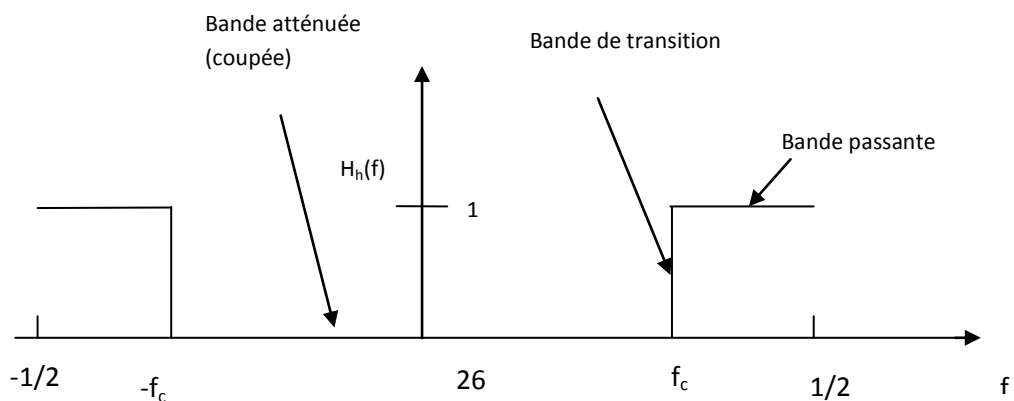
On peut aussi déterminer cette RF $H_B(f)$ ainsi que la RI $h_B(n)$ à partir de celles du filtre passe bas comme suit

$$H_B(f) = H(f - f_{ce}) + H(f + f_{ce}) \Rightarrow$$

$$h_B(n) = 2.h(n) \cdot \cos(2\pi f_{ce} n) = 2 \cdot \frac{\sin(\pi B n)}{\pi n} \cdot \cos(2\pi f_{ce} n) \quad 8$$

c) Filtre passe haut

$$H_h(f) = \begin{cases} 1 & \text{si } |f| \geq f_c \\ 0 & \text{ailleurs} \end{cases} \quad 9$$

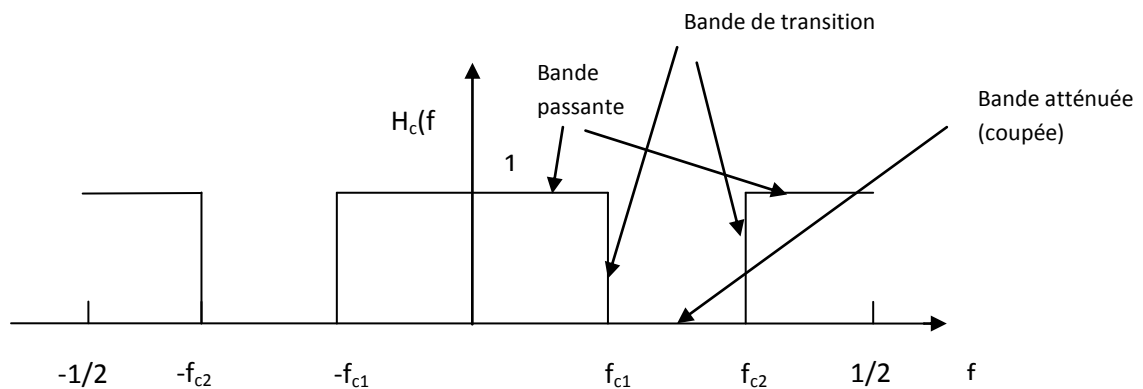


La RI d'un filtre passe haut peut être déduite de (8) en remplaçant simplement f_{ce} par

$$f_{ce} = \frac{f_c + 1/2}{2} = \frac{1-B}{2} \quad \text{où } B \text{ largeur de la bande du passe haut}$$

d) Filtre coupe bande

$$H_c(f) = \begin{cases} 0 & \text{si } f_{c1} \leq |f| \leq f_{c2} \\ 1 & \text{ailleurs} \end{cases}$$



La RI d'un filtre coupe bande peut être aussi obtenue en considérant un filtre passe bas en parallèle avec un passe haut.

$$h_{cB}(n) = h(n) + h_n(n)$$

Remarque :

On peut constater que quel que soit le filtre peut être obtenu à partir du filtre passe bas (ou passe bande) en modifiant la RI (6) ou (8) selon la nature du filtre désiré.

3.3.1 Résultats et discussion

Dans notre travail, nous avons représenté un filtre numérique passe bande RIF par un modèle neuronal. Pour cela nous avons appliqué un passe bande à un signal composé de trois signaux cosinus superposés x_1 , x_2 et x_3 de fréquences f_1 , f_2 et f_3 respectivement.

La figure suivante représente le signal composé des trois signaux en haut et celui désiré en bas. Les graphes à gauche représentent les signaux dans le domaine du temps et ceux à droite leurs versions dans le domaine fréquentiel.

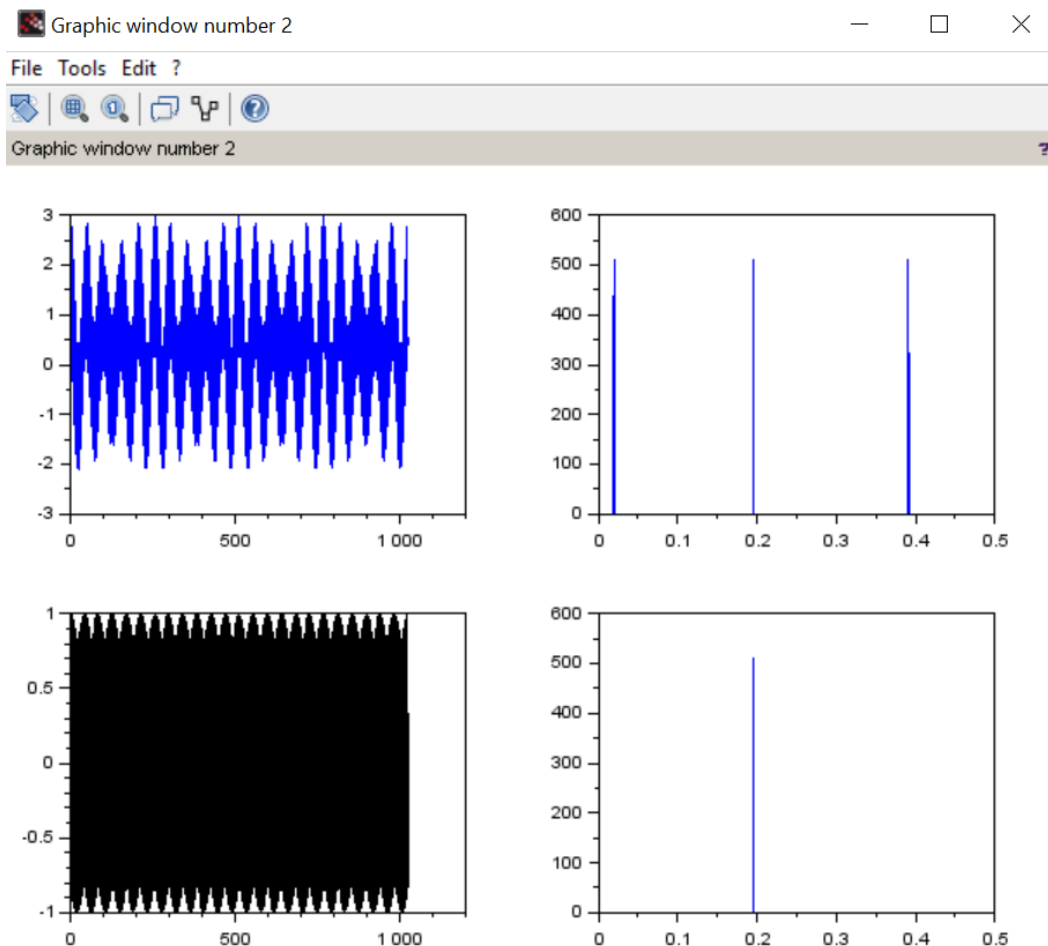


Figure 13 : Représentation temporelle et fréquentielle du signal d'entrée en haut, et celles du signal désiré en bas.

3.3.2 Application du passe bande

L'objectif de ce travail est de filtrer le signal x_2 par un passe bande et de comparer les résultats à ceux obtenus par le modèle neuronal. Le signal désiré est le signal x_2 de fréquence $f_1=0,2\text{Hz}$. Le filtre passe bande que nous avons conçu et qui permet de filtrer ce signal, est représenté sur la figure ci-dessous.

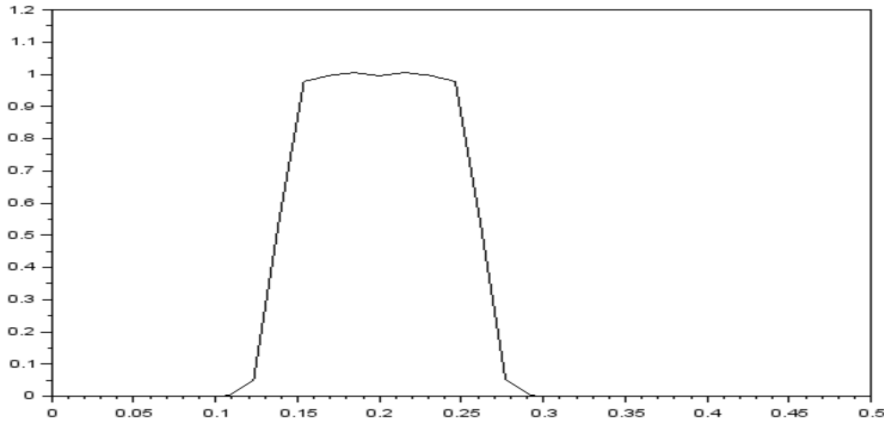


Figure 14 :filtre passe bande de fréquence centrale 0,2Hz et de largeur de bande B=0,1Hz

La figure suivante à gauche représente le signal filtré et celle à droite le signal désiré. En comparant le signal filtré à celui désiré, on peut constater que les TF sont très similaires à l'exception d'une distorsion en bas du pic de la TF du signal filtré. On peut dire, donc, que le filtre passe bande a pu filtrer le signal désiré avec une très bonne approximation

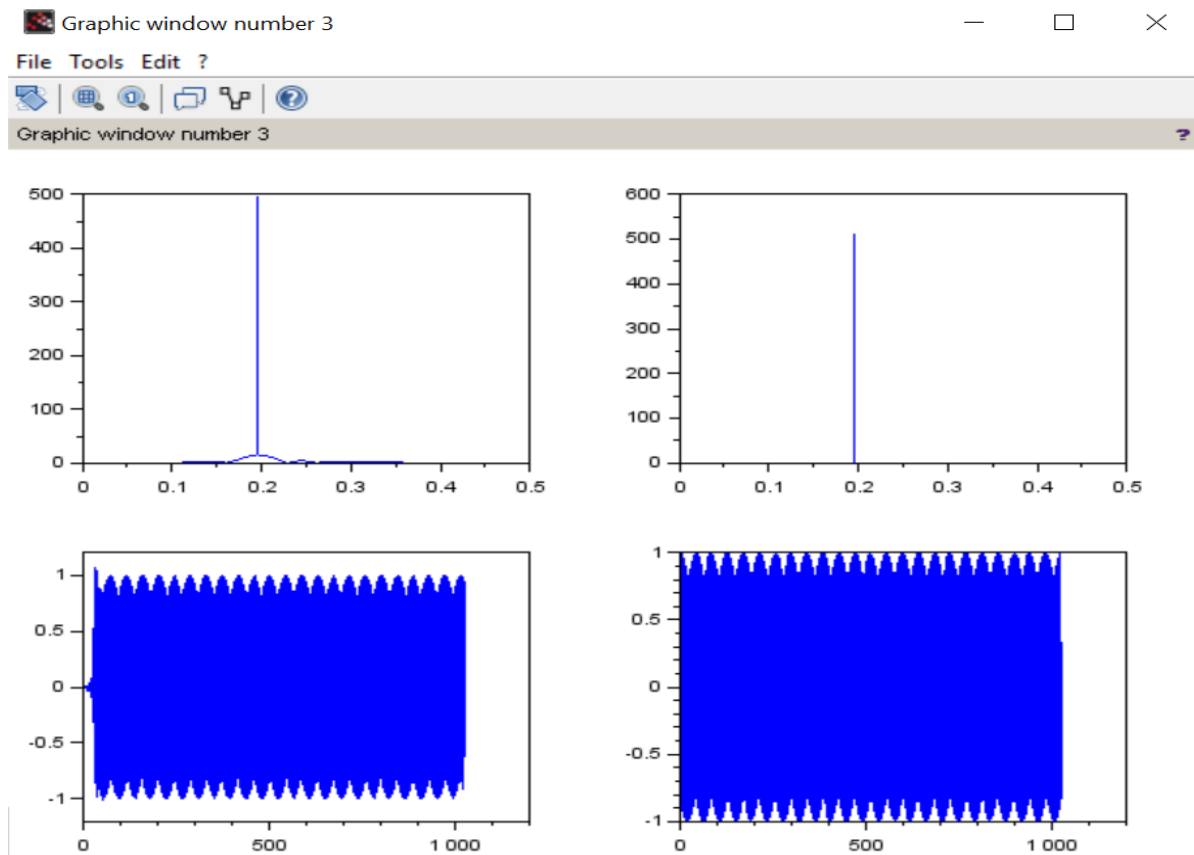


Figure 15 : filtrage par un passe bande

3.4 Modèle neuronale

3.4.1 Les caractéristiques neuronales :

Les caractéristiques du modèle neuronal utilisé dans notre travail sont :

- Nombre de couches
- Fonction d'activation
- Nombre de neurone dans chaque couche
- Paramètres du réseau du neurone (poids)

Exemple : Le perceptron simple

est dit simple parce qu'il ne dispose que de deux couches ; la couche en entrée et la **couche** en sortie. Le réseau est déclenché par la réception d'une information en entrée.

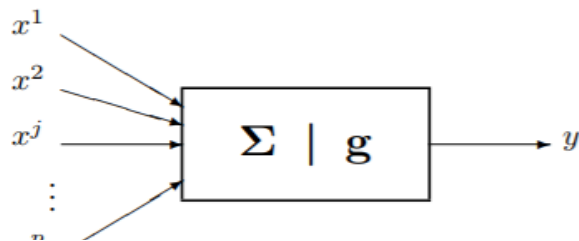


Figure 16 : architecture formelle de neurone

3.4.2 Représentation d'un neurone formel :

Le réseau de neurones que nous avons conçu pour représenter un passe bande a l'allure suivante avec la tangente hyperbolique comme fonction d'activation.

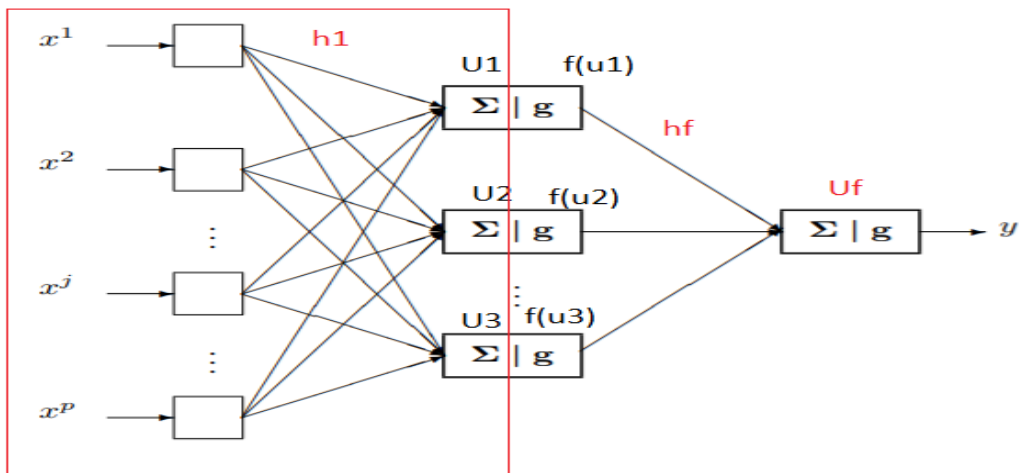


Figure 17 : Exemple de perceptron multicouche

La fonction d'activation correspondante à notre application est une tangente hyperbolique donnée par l'expression suivante:

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \text{ sh}$$

L'allure de son graphe est le suivant

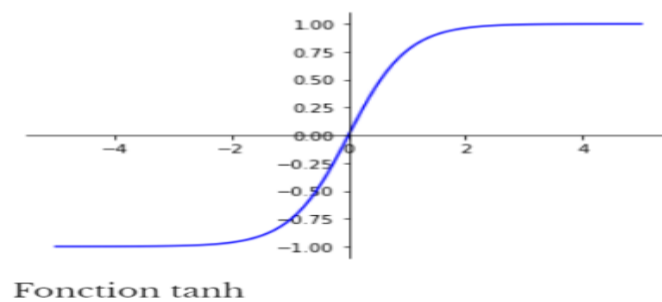


Figure 18 : fonction tanh(x)

La **fonction** hyperbolique ou **tanh** est souvent utilisée dans les réseaux de neurones comme **fonction d'activation**. Elle fournit une sortie entre -1 et +1. Il s'agit d'une extension de la sigmoïde logistique; la différence est que la sortie s'étend ici entre -1 et +1.

3.5 Choix de l'algorithme d'apprentissage :

Types d'apprentissage :

L'objectif de la phase d'apprentissage des RNA est de trouver, parmi toutes les fonctions paramétrées par les poids synaptiques, celle qui s'approche le plus possible de l'optimum. L'apprentissage consiste, donc à minimiser une fonction de coût à l'aide des algorithmes d'optimisation. L'un d'eux est décrit dans la section suivante. Il existe deux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé. Le superviseur, ou professeur, fournit au réseau des couples d'entrée-sortie. Il fait apprendre au réseau l'ensemble de ces couples, par une méthode d'apprentissage, comme la rétro-propagation du gradient de l'erreur, en comparant pour chacun d'entre eux la sortie effective du réseau et la sortie désirée. L'apprentissage est terminé lorsque tous les couples entrée-sortie sont reconnus par le réseau. Ce type d'apprentissage se retrouve, entre autres, dans le perceptron. Le deuxième type c'est l'apprentissage non supervisé. Cet apprentissage consiste à détecter automatiquement des régularités qui figurent dans les exemples présentés et à modifier les poids des connexions pour que les exemples ayant les mêmes caractéristiques de régularité provoquent la même sortie. Les réseaux auto-organiseurs de Kohonen (1981) sont les réseaux à apprentissage non supervisé les plus connus [26] . La plupart des algorithmes d'apprentissage des réseaux de neurones formels sont des algorithmes d'optimisation (méthode supervisée). La méthode d'optimisation la plus célèbre et la plus utilisée en analyse neuronale est la rétro-propagation du gradient de l'erreur.

3.5.1 Algorithme LMS (back propagation) :

Plusieurs variantes de cet algorithme ont été développées par des chercheurs travaillant dans des domaines différents. La première formulation de la version actuelle a été faite par Werbos en 1974. Il a été appliqué pour les réseaux multicouches par Rumlehart en 1986. L'algorithme de la rétro-propagation altère les coefficients synaptiques (W_i) du réseau dans le sens inverse du gradient du critère d'erreur, en utilisant seulement les données d'entrée/sortie

Rétro-propagation: Cette étape est similaire à la précédente. Cependant, les calculs s'effectuent dans le sens inverse (de la sortie vers l'entrée). A la sortie du réseau, on forme le critère de performance e en fonction de la sortie réelle de système et sa valeur désirée. Puis, on évalue le gradient de e par rapport aux différents poids en commençant par la couche de sortie et en remontant vers la couche d'entrée.

$$h(i) = h(i - 1) + U \cdot e \cdot \frac{\partial e}{\partial h}$$

U : le pas du gradient

e : erreur entre la sortie neuronale y_n et la sortie réelle y_r . $e = y_r - y_n$

3.5.2 Application du réseau de neurones

Le but de ce travail est de concevoir un réseau de neurones qui effectue la même tâche qu'un passe bande RIF. Pour cela, nous avons commencé par choisir un perceptron multicouche dont les caractéristiques sont les suivantes

- nombre d'entrées : $p = 123$
- une seule couche interne et un seul neurone dans la couche de sortie
- nombre de neurone : $N = 4$ dans la couche interne
- le pas de gradient : $\mu = 0.1$
- fonction d'activation : \tanh

Après apprentissage du ANN, nous avons obtenus les résultats suivants

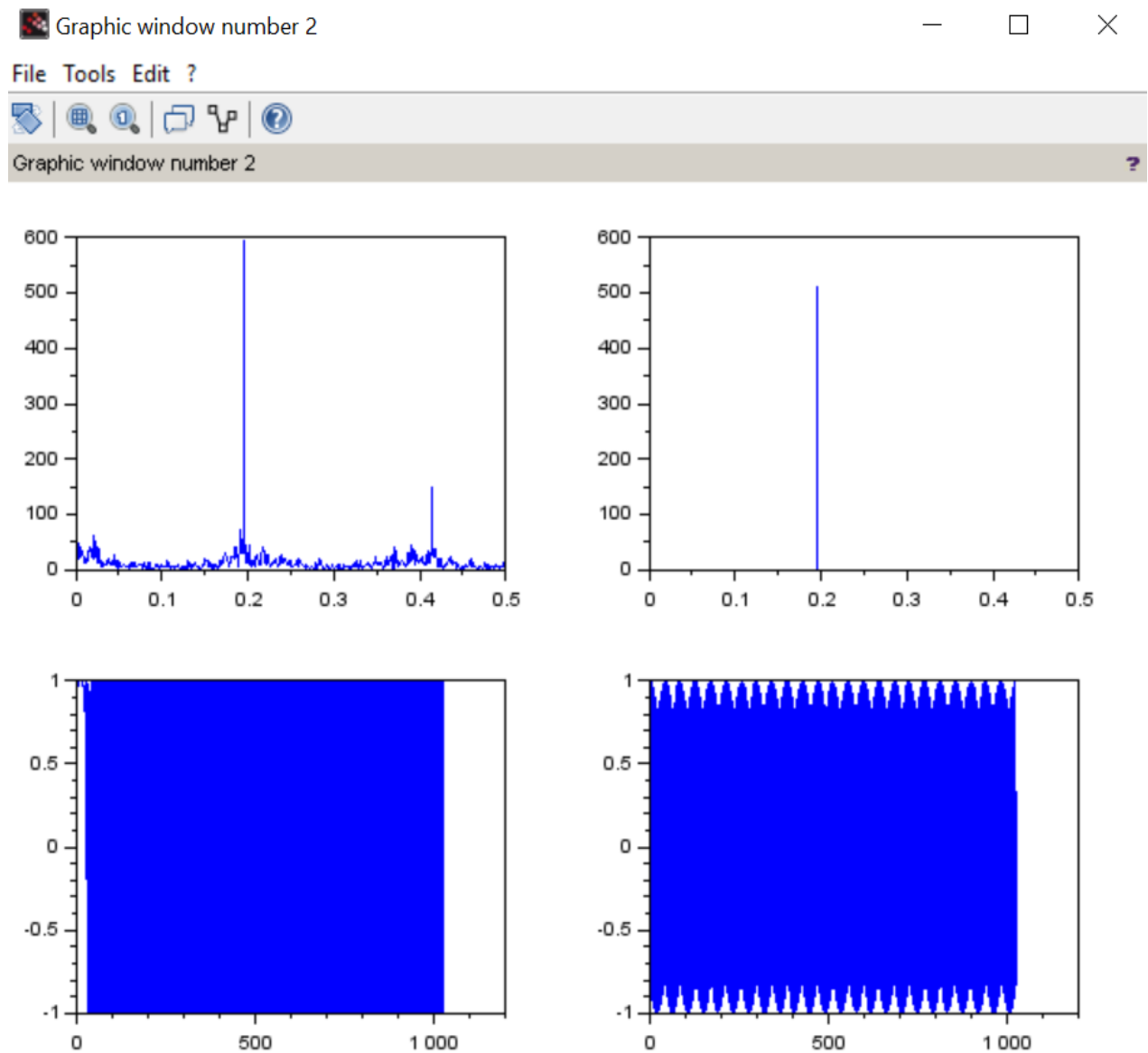


Figure 19 : filtrage neuronale pour un nombre de neurones 5

On constate que le ANN avec une couche interne de 4 neurones, n'arrive pas à éliminer complètement le pic correspondant au signal x_3 . Pour cela, nous avons augmenté le nombre de neurones dans la couche interne pour se rapprocher des résultats obtenus avec le filtre passe bande.

La figure suivante montre les résultats de l'application du ANN avec un nombre de neurones dans la couche interne égale à 10. On remarque que le pic de la TF du signal x_3 n'est pas encore complètement éliminé

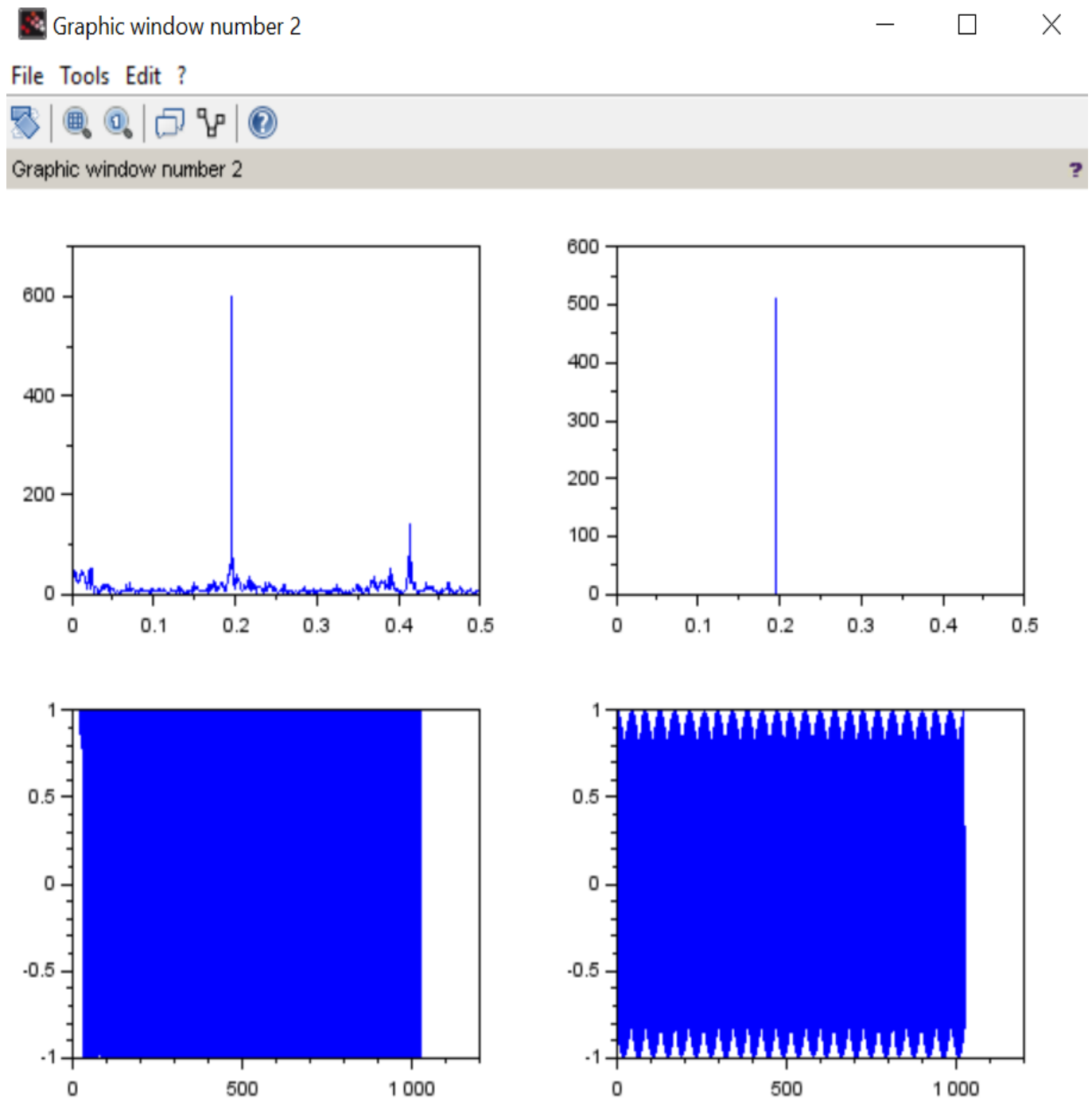


Figure 20 : A gauche : Représentation fréquentielle et temporelle du signal de sortie neuronale avec 10 neurones, et à droite : celles du signal désiré.

Nous avons, donc, essayé d'augmenter encore le nombre de neurones dans la couche interne jusqu'à 15 neurones. Les résultats sont illustrés dans la figure suivante dans laquelle, il est facile de constater qu'une grande partie du pic de x_3 a disparu. Cependant, en comparant le signal obtenu par le modèle neuronal proposé à celui obtenu par filtrage

passer bande, on peut dire que le filtre passe bande a abouti à des résultats légèrement meilleurs que ceux obtenus par le modèle neuronal proposé

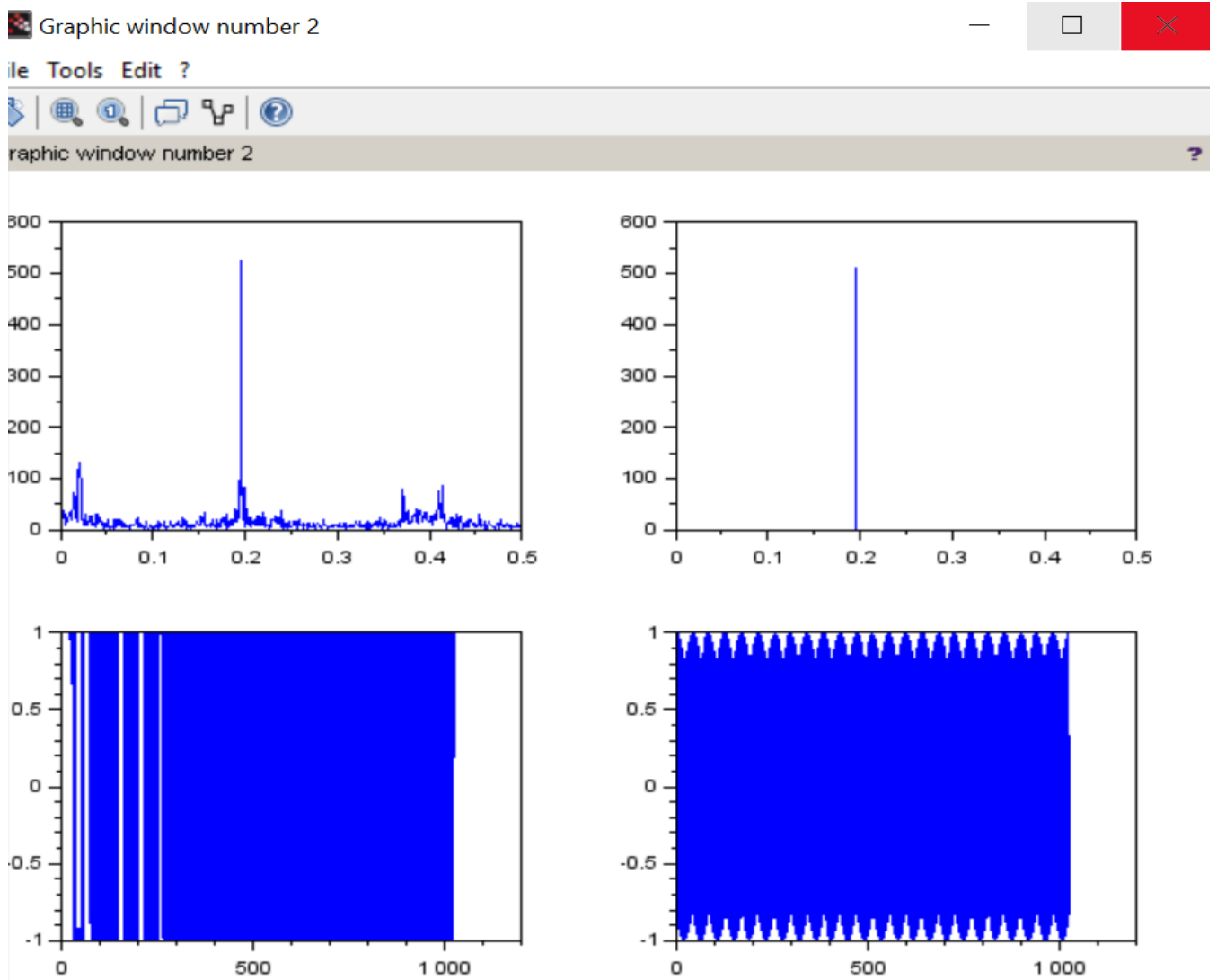


Figure 21 : A gauche : Représentation fréquentielle et temporelle du signal de sortie neuronale avec 15 neurones dans la couche interne, et à droite : celles du signal désiré.

Conclusion Générale

Dans ce travail nous avons conçu un réseau de neurones dont le but de modéliser un filtre RIF numérique passe bande. Nous avons choisi d'utiliser la fonction tangente hyperbolique comme fonction d'activation à cause d'une part de ces valeurs continues entre -1 et 1, et d'autre part de sa dérivabilité qui permet d'appliquer l'algorithme de la retro-propagation en se basant sur le gradient.

Après avoir testé notre modèle neuronale en commençant par un neurone et une fonction d'activation linéaire puis ajouté au fur et à mesure des neurones ainsi que changer la fonction d'activation linéaire par une hyperbolique, on a aboutit à concevoir le modèle neuronale désiré. Ce dernier est constitué d'une seule couche interne de onze neurones et une seule sortie avec un neurone. L'apprentissage du ANN a été fait afin que le modèle neuronale final parvienne à jouer le rôle d'un filtre passe bande RIF. Pour cela, nous avons effectué une étude comparative entre notre modèle neuronale obtenu et un filtre passe bande RIF que nous avons conçu. Les résultats obtenus sont encourageants, ce qui permet de proposer en perspective à concevoir des modèles neuronaux pour le filtrage des signaux, et ainsi remplacer les filtres en générales par des réseaux de neurones.

Bibliographie

- [1] :C.Touzet « les réseaux de neurone artificiels. Introduction ou connexionnisme »
- [2] :paynter.H.Analyses and design of engineering system. MIT presse (1961)
- [3] : KarnoppD.andR.RobsenbergSystemsdynamics.AunifiedApproch ,WileyIntersciences ;
New York , 1990.
- [4] :B.ouldBoumama ,A.K.Samantary,K.Medjaher ,,M.Starsowiecki et G.Dauphin-Tanguy
Et (2004).Model Builder u singfunctional and bond graph tools for FDI desing .control et
Engineering ,CEP , VOL.13/7pp . 875-891.
- [5] :] W. James, Psychology (Briefer Course), New york: Holt, Chapter XVI, "Association ,"pp.
253-279, 1890.
- [6] : W. McCulloch and W. Pitts, "A logicalcalculus of the ideas immanent in nervousactivity,"
Bulletin of MathematicalBiophysics 5: 115-133, 1943
- [7] : G.Dreyfus,J-M. Martinez, M.Samuelides, M.B. Cordon, F. Badran, S. theria, L. herault,
"Réseaux de neurons" , Eyrolesedit
- [8] : Busson F., « Les bond graphs multi-énergies pour la modélisation et la surveillance
en géniedesprocédés»,Thèsededoctorat,UniversitédessciencesettechnologiesdeLille,
France,2002.

[9] P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, "Modélisation et identification des processus (tome 2)," Méthodes et pratiques de l'ingénieur, éditions Technip, Paris, 1992.

[10] :] G. Dauphin-Tanguy, "Les bond graphs et leur application en mécatronique," Techniques de l'Ingénieur : Informatique Industrielle, Automatique, S 7 222, Paris, 1999.

[11] : G. Dauphin-Tanguy, Les Bond Graphs, édition Hermès, Paris, 2000.

[12] : D. Karnopp, D. Margolis, R. Rosenberg, System Dynamics : Modeling and Simulation of Mechatronic Systems, John Wiley & sons, 2000 (3rd edition).

[13] : Busson F., « Les bond graphs multi-énergies pour la modélisation et la surveillance en génie des procédés », Thèse de doctorat, Université des sciences et technologies de Lille, France, 2002

[14] : Dauphin-Tanguy G., « Les bond graphs », Hermes Science Publications, Paris, France, 2000.

[15] :] Document Internet (Traitement des signaux numériques -II filtrage adaptatif et analyse spectrale) Auteur : Jacob Benesty. (Avril 2005).

[16] : Maurice Rivoire /Jean-louis Ferrier : « Commande par ordinateur identification (tome3) ».Edition Eyrolles 1997.

[17] : Ioan Dore Landau : « Identification et commande des systèmes », Edition Hermès Paris – Londres – Lausanne 1993

[18] : Projet de fin d'études : Etude de deux algorithmes d'identification (l'algorithme des moindres carrés récursifs et l'algorithme de Levy (année 2006)).

Proposé et dirigé par : Mr. Mansouri

[19] : Document Internet (Identification des systèmes linéaires Polytech'Tours Département Productique « Novembre 2004 »).

[20] : Michael I. Jordan. "Serial order: A parallel distributed processing approach". Technical report, Department of Brain and Cognitive Science, Massachusetts Institute of Technology, University of California, San Diego, USA, May, 1986

[21] :] Paul. J. Werbos. "Backpropagation through Time : what it does and how it do ". Proceeding of the IEEE, October, 1990.

[22] Ryotaro Kamimura. "Application of Temporal supervised Learning Algorithm to generation of natural language". IJCNN '90, P: 1-201 -1-207, 1990.

[23] Ryotaro Kamimura. "Recognition And Restoration of Periodic Patterns with Recurrent Neural Net". IJCNN 90, P:689-698, 1990.

[24] E.B. Baum. " On the Capabilities of Multilayer Perceptrons". /. Complexity, Vol 4, P: 193-215, 1988

[25] : Hinton D.E. Rumelhart and R.J. Williams. "Learning Internal Representations by Error Propagation". D.E. Rumelhart, McClelland and The PDP Research Group, Parallel Distributed Processing: Explorations in The Microstructures of Cognition, Vol. 1, Cambridge: MIT Press, pp. 318-362, 1986

[26] : J.S Baras and A. La Vigna. "Convergence of Kohonen's Learning Vector Quantization". IJCNN International Joint Conference On Neural Networks, New York: IEEE Press, 111:17-20, 1990.

