

3/99

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

Ecole Nationale Polytechnique

D.E.R. de Génie Electrique et Informatique
Département d'Electrotechnique



Projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état
en Electrotechnique

Thème

*Identification des défauts dans un réseau électrique
par la technique des réseaux de neurones*

Proposé et dirigé par :

Mr A.Hellal
Mr L.Nezli

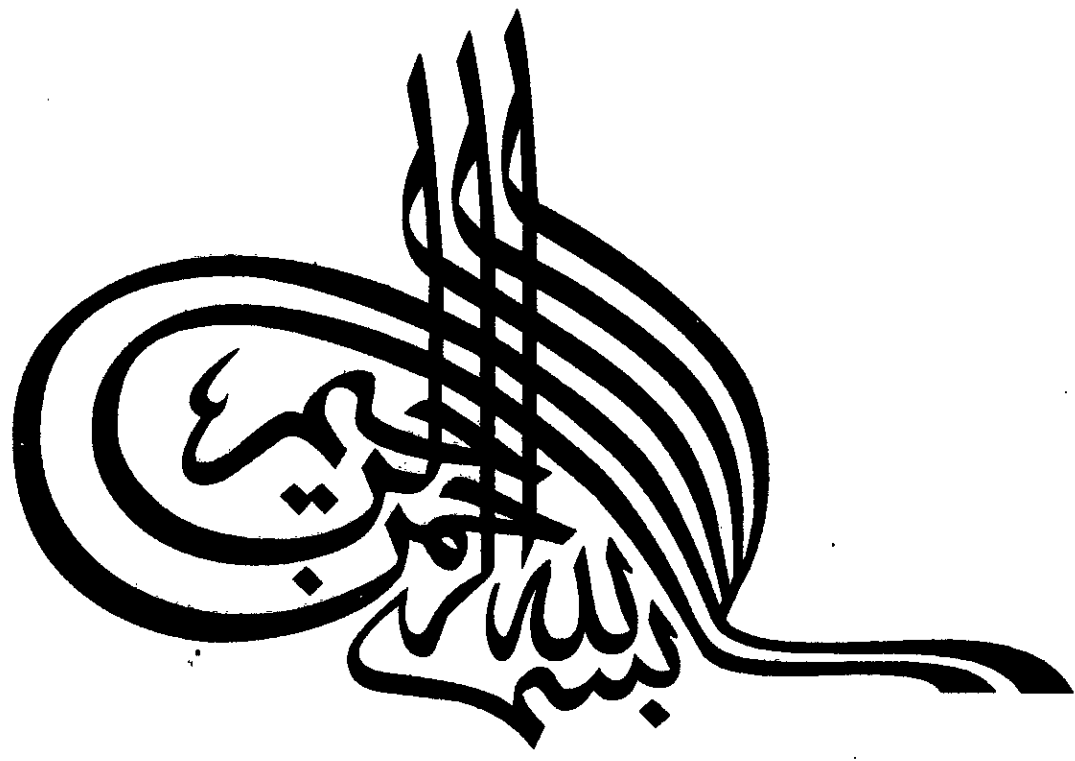
Etudié par :

Ailam El-hadj
Benzaïoua Ammar

Promotion : juin 99

E.N.P.10, Avenue Hassen-Badi, EL-HARRACH, ALGER.

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique



Dédicace

Je dédie ce modeste travail:

A mes très chers parents ;

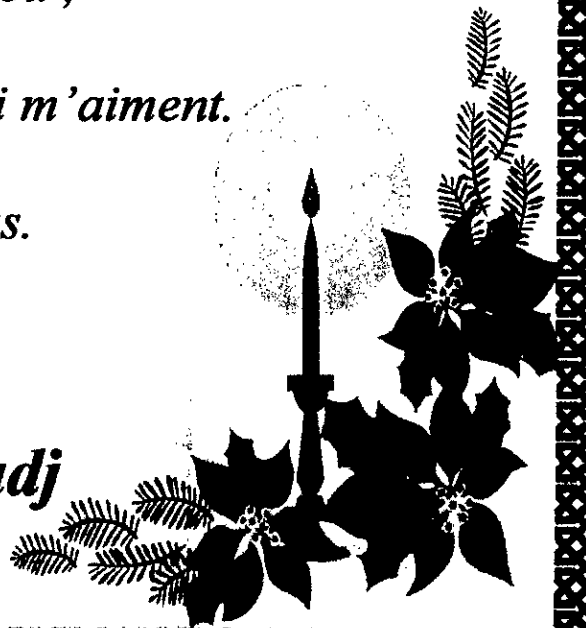
*A mes frères et sœurs, en particulier mon frère
Said et mon neveu Toufik ;*

*A tous mes amis de Bouraoui, l'ENP et
B.E.K, en particulier Ammar, Fateh, Yacine et
Mohamed ;*

A tous ceux qui m'aiment.

A vous.

El hadj



Dédicace

Je dédie ce modeste travail,

*Aux êtres les plus tendres à mes yeux et les plus
chers à mon cœur, à qui je dois énormément
et que je ne remerciais jamais assez :*

Mes parents.

*A mes frères et sœurs, en particulier
ABDENOUR, MOHAMMED et sa femme pour leur
soutien moral et matériel.*

A mon neveu NABIL et ma nièce ZAZA

*A toute ma famille, à ceux que j'aime et qui sauront
se reconnaître.*

*A tous mes amis de Bouraoui, l'ENP et
Taher, en particulier EL-HADJ.*

Ammar.



REMERCIEMENTS



Que tous les enseignants qui ont contribué à notre formation trouvent en ce modeste travail le témoignage de notre profonde et sincère gratitude et l'expression de notre vive amitié.

On tient particulièrement à assurer Mrs A.Hellal et L.Nezli de toute notre reconnaissance pour les conseils qu'ils n'ont cessé de nous prodiguer tout au long de la réalisation de cette thèse.

Nous tenons également à remercier messieurs les membres de jury pour l'honneur qu'ils nous ont fait en acceptant de siéger à notre soutenance, sans oublier tout le personnel de la bibliothèque, en particulier KRIMO.

Nos remerciements vont également à nos confrères et amis de la promotion.

قمنا في هذه الدراسة باستخدام شبكة الأعصاب الاصطناعية لتحديد الخلل في الشبكات الكهربائية وقد استخدمنا طريقتين. حيث نستعمل في الأولى القيم العظمى للتوتر كشعاع الدخول و في الثانية هذه القيم بالإضافة إلى فرق الطور بين التوترات.

الكلمات المفتاحية :

الشبكة الكهربائية، الدارة القصرى، شبكة الأعصاب، عداد الانتشار.

Résumé :

Dans cette étude, nous avons utilisé les réseaux de neurones artificiels pour l'identification des défauts dans un réseau électrique. Deux méthodes sont présentées; dans la première, les entrées du réseau de neurones sont les amplitudes de tension de trois phases seules, et dans la deuxième ces amplitudes plus le déphasage entre les tensions.

Mots clés :

Réseau électrique, court-circuit, réseau de neurones, compteur de propagation.

Abstract :

In this study, we used the artificial neurons networks for identifying the faults in the electric networks. Two methods had been used; first the networks inputs are the amplitude of tension in three phases, and in the second amplitudes and phase angle between tensions.

Key words :

Electric networks, short-circuit, neurons network, counter propagation network.

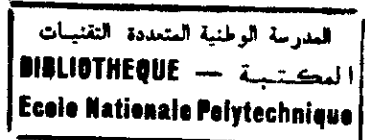
SOMMAIRE

Chapitre I : Introduction générale	1
Chapitre II : Rappels de calculs de courts-circuits	4
II.1. Introduction.....	4
II.2. Définition de courant de courts-circuits.....	5
II.3. Calcul de courts-circuits en utilisant Z_{BUS}	6
II.4. Calcul de courts-circuits par la méthode des composantes symétriques.....	12
II.5. Organigramme de calculs de courts-circuits.....	16
II.6. Exemple de calcul de courts-circuits.....	18
II.7. Régimes transitoires.....	22
II.8. Conclusion.....	26
Chapitre III : Réseaux de neurones : CPN	27
III.1. Introduction.....	27
III.2. Structure et fonctionnement d'un neurone artificiel.....	29
III.3. Eléments constitutifs d'un réseaux de neurones.....	30
III.4. Apprentissage des réseaux de neurones.....	33
III.4.1. Types d'apprentissage.....	34
III.4.2. Problèmes d'apprentissage.....	34
III.5. Compteur de propagation (CPN).....	36
III.5.1. Eléments constitutifs du CPN.....	38
III.5.2. Procédure des données du CPN.....	51
III.5.3. Apprentissage du CPN.....	54
III.5.4. CPN complet.....	60
III.6. Conclusion.....	62

Chapitre VI : Identification des courts-circuits par le CPN.....	63
VI.1. Introduction.....	63
VI.2. Identification des défauts.....	63
VI.2.1. Méthode 1.....	64
VI.2.2. Méthode 2.....	72
VI.3. Conclusion.....	84
Conclusion Générale.....	85

Chapitre I

Introduction générale



Les réseaux électriques représentent des investissements considérables consentis par les compagnies d'électricité pour alimenter leurs clients aux meilleures conditions de coûts et de qualité de service.

Pour des raisons techniques et économiques évidentes, il n'est pas possible de construire des réseaux exempts de défauts de fonctionnement ; ils sont, en particulier, exposés aux agressions naturelles comme la foudre. Les réseaux sont donc soumis à des perturbations qui peuvent mettre en cause la pérennité du matériel et la qualité de service rendu et dont il faut chercher à minimiser les conséquences.

Tout défaut doit donc être identifié immédiatement et l'ouvrage affecté séparé du réseau sans délai ; c'est l'objet de la protection des réseaux. Les différents composants des réseaux sont conçus, construits et entretenus de façon à réaliser le meilleur compromis entre coût et risque de défaillance [1]. Ce risque n'est donc pas nul et des incidents ou défauts de courts-circuits viennent perturber le fonctionnement des installations.

Les courts-circuits sont caractérisés par leur durée et l'intensité du courant. Les courts-circuits ont des effets néfastes [1] :

- sur le fonctionnement des réseaux ; les effets néfastes des courts-circuits sont surtout à redouter sur les réseaux THT et MT sur lesquels débitent des groupes générateurs de fortes puissances. Les courts-circuits lorsqu'ils sont proches des centrales, entraînent une diminution du couple résistant des machines et donc une rupture de l'équilibre entre celui-ci et le couple moteur ; s'ils ne sont pas éliminés rapidement, ils peuvent conduire à la perte de stabilité de groupes générateurs et à des fonctionnements hors synchronisme. Des temps d'élimination des courts-circuits de l'ordre de 100 à 150 ms sont en général considérés comme des valeurs à ne pas dépasser sur les réseaux électriques.
- tenue des matériels ; les courts-circuits provoquent des surintensités violentes, dans le cas de défaut triphasé, pouvant dépasser 20 à 30 fois le courant de service normal. Ces

surintensités amènent divers types de contraintes : des contraintes thermiques dues au dégagement de chaleur par effet de Joule dans les conducteurs, des contraintes mécaniques dues aux efforts électrodynamiques ; ceux-ci entraînent notamment le balancement des conducteurs aériens et le déplacement des bobinages des transformateurs ; ces efforts, s'ils dépassent les limites admises lors de la construction, sont souvent à l'origine d'avaries graves. De plus, l'arc électrique consécutif à un court-circuit met en jeu un important dégagement local d'énergie pouvant provoquer des dégâts importants aux matériels et même être dangereux pour le personnel travaillant à proximité.

- qualité de la fourniture ; dans la plus part des cas, les courts circuits se traduisent pour les utilisateurs par des chutes de tension dont l'amplitude et la durée sont fonction de différents facteurs tel que nature de court-circuit, structure du réseau affecté.
- circuits de télécommunication ; la présence d'un court-circuit dissymétrique entre une ou deux phases d'une ligne d'énergie et la terre entraîne la circulation d'un courant homopolaire qui s'écoule à la terre par les points neutres du réseau. Une tension induite longitudinale, proportionnelle à ce courant, apparaît sur les lignes de télécommunication qui ont un trajet parallèle à la ligne d'énergie. Cette tension peut atteindre des valeurs dangereuses pour le personnel et les installations de télécommunications.
- sécurité des personnes ; la mise sous tension accidentelle des masses, les élévations de potentiel liées à l'écoulement des courants de défaut à la terre, les conducteurs tombés au sol, etc. sont autant de situations pouvant présenter des risques pour la sécurité des personnes ; le mode de mise à la terre des points neutres joue de ce fait un rôle essentiel.

Certains défauts sont difficiles à identifier et présentent d'autant plus de risques. Donc, la connaissance des courants et puissances apparentes de court-circuit est d'une importance capitale pour l'exploitation de réseaux électriques qui a en vue non seulement le dimensionnement correct de ses appareils de commutation, mais aussi la réalisation économique de systèmes de protection surs. Pour remplir leur rôle, les protections doivent :

- Détecter la présence d'un défaut ;
- Identifier l'ouvrage atteint ;
- Commander les organes de coupure, dont l'ouverture conduira à l'isolement de cet ouvrage par rapport au reste du réseau ;

Traditionnellement, telle détection de défauts est faite par la mesure des tensions dans les trois phases, ou les courants traversant les éléments du réseau. Dans notre travail nous avons présenté la possibilité d'utiliser les réseaux de neurones artificiels pour l'identification des défauts de courts-circuits pouvant apparaître dans un réseau électrique. Les caractéristiques importantes des réseaux de neurones artificiels poussent ces derniers d'être appliqués dans plusieurs domaines pratiques d'activité. Ils sont rapides parce qu'ils exécutent plusieurs opérations de calculs en parallèle. Une des plus importantes caractéristiques est la généralisation [2].

Notre travail est composé de quatre parties principales ;

- Dans un premier temps une étude générale sur le calcul de court-circuit est présentée avec les différentes étapes de calculs de tous les paramètres caractérisant le réseau après apparition du défaut, telle que le courant injecté, la tension de court-circuit au nœud de défaut, les tensions des nœuds autres que le nœud de défaut et les courants de lignes. Un programme en FORTRAN a été développé pour faire tous ces calculs dans le cas d'un réseau électrique quelconque.
- Dans le troisième chapitre, les réseaux de neurones artificiels sont présentés d'abord d'une façon générale : principe de fonctionnement, éléments constitutifs du réseau de neurones, architecture et types d'apprentissage existants. Ensuite nous nous sommes intéressés à un type particulier le CPN (Counter Propagation Network).
- Dans le dernier chapitre l'application du CPN est faite pour l'identification de courts-circuits, avec l'utilisation de deux différentes méthodes.

Nous terminerons notre travail par une conclusion générale.

Chapitre 2 Rappels de calculs de courts-circuits

II.1. Introduction :

Pour le dimensionnement du matériel, on cherche à obtenir rapidement une valeur maximale des courants de courts-circuits. En général, cette valeur correspond à des courts-circuits aux sommets, francs, triphasés ne nécessitant que la considération du système de composantes directes [3].

Dans ce travail, les approximations suivantes sont faites [3] :

- Le régime avant défaut est pris tel que le plan de tension est supposé uniforme (modules constants, phases nulles) sur l'ensemble du réseau. Les courants de transit sont donc négligés devant le courant de défaut ;
- Les résistances et susceptances des ouvrages sont négligées ainsi que les charges ;
- Les courants du défaut injectés pour les nœuds autres que le nœud de défaut sont négligeables devant le courant du défaut injecté au nœud de court-circuit.

Dans ces conditions, correspondant à des hypothèses extrêmes, on peut obtenir un dimensionnement plus que suffisant du matériel du réseau.

Ce chapitre est donc composé de trois parties principales :

- 1- Une première partie qui développe toutes les relations nécessaires pour le calcul du courant et la tension de court-circuit au nœud de défaut, ainsi que les tensions des autres nœuds et les courants de lignes traversant tous les éléments du réseau, en utilisant la méthode générale (méthode des composantes relatives aux phases).
- 2- Une autre partie qui utilise la méthode des composantes symétriques (calcul de courts-circuits pour les réseaux triphasés équilibrés).

3- Une troisième partie qui développe un programme en FORTRAN de calcul de court-circuit pour un réseau électrique quelconque. Ce programme fait le calcul du courant et de la tension après défaut au nœud de court-circuit, ainsi que les tensions des autres nœuds et les courants de lignes du réseau après défaut. Les résultats de calculs sont présentés en composantes symétriques et en composantes réelles.

II.2. Définition du courant de court-circuit

Lors de l'apparition d'un court-circuit, du fait de la nature inductive du réseau [3], la valeur à fréquence industrielle ne s'établit pas immédiatement. Le courant peut approximativement être considéré comme la somme de deux termes :

- Une composante transitoire apériodique qui décroît rapidement de la forme $I\sqrt{2}e^{-t/T} \cos\theta$. La constante du temps T est de l'ordre de 50 à 100 ms pour un réseau à très haute tension suivant les caractéristiques du réseau et la position du défaut. θ est l'écart angulaire entre le zéro de tension à fréquence industrielle et l'instant d'apparition du court-circuit ;
- Une composante sinusoïdale à fréquence industrielle correspondant au régime établi de la forme $I\sqrt{2} \cos(\omega t + \theta)$. Lorsque le défaut est proche des machines, la variation de l'impédance interne de celles-ci avec le temps, fait que l'amplitude du courant de court-circuit varie également avec le temps. L'impédance interne des machines doit donc être choisie en fonction de l'instant considéré. Si le défaut est très proche des machines, il n'est pas possible de négliger l'effet des régulateurs de tension (surexcitation) [3]. Pour ce cas particulier, il est nécessaire de considérer une modélisation détaillée des générateurs comme celle utilisée pour la simulation des phénomènes de stabilité.

Par ailleurs, la discontinuité due à l'apparition du court-circuit génère des transitoires s'amortissant rapidement (quelques millisecondes) qui ne sont pas pris en compte par les calculs classiques de courants de court-circuit. Si on veut tenir compte de ces phénomènes, il faut utiliser une simulation de transitoires. C'est le cas notamment des études concernant les protections à temps de repense très court fonctionnant durant ces phénomènes transitoires.

II.3. Calcul de court-circuit en utilisant Z_{BUS}

II.3.1. Représentation du réseau:

La représentation triphasée d'un réseau électrique est montrée dans Fig.(2.1). En général, une suffisante précision dans les études du court-circuit peut être obtenue avec une représentation simplifiée. La représentation triphasée simplifiée est montrée dans Fig.(2.2) et est obtenu par :

- Représenter chaque machine par une source de tension constante derrière la réactance de la machine, transitoire ou subtransitoire
- Négliger les connections shunts, charges, etc.

En particulier pour les réseaux de haute tension, il est suffisant de représenter les lignes de transport et les impédances des transformateurs par leurs réactances correspondantes dans les études du court-circuit [4].

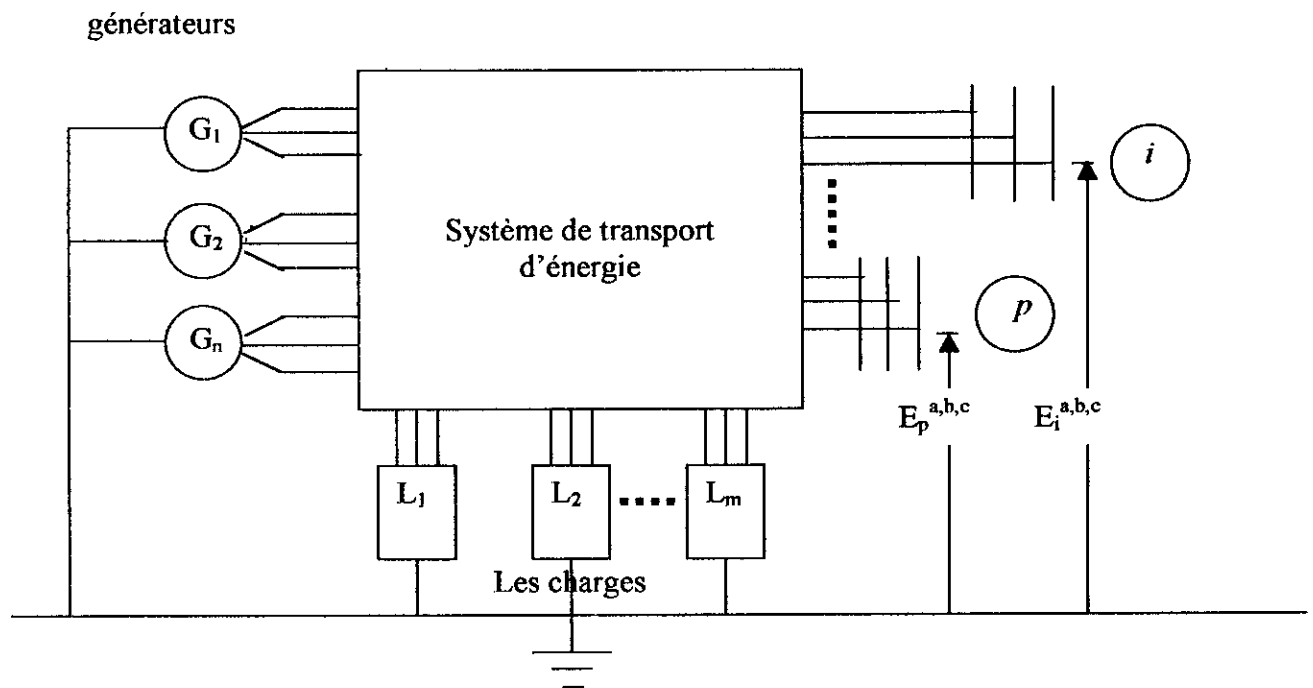


Figure (2.1)

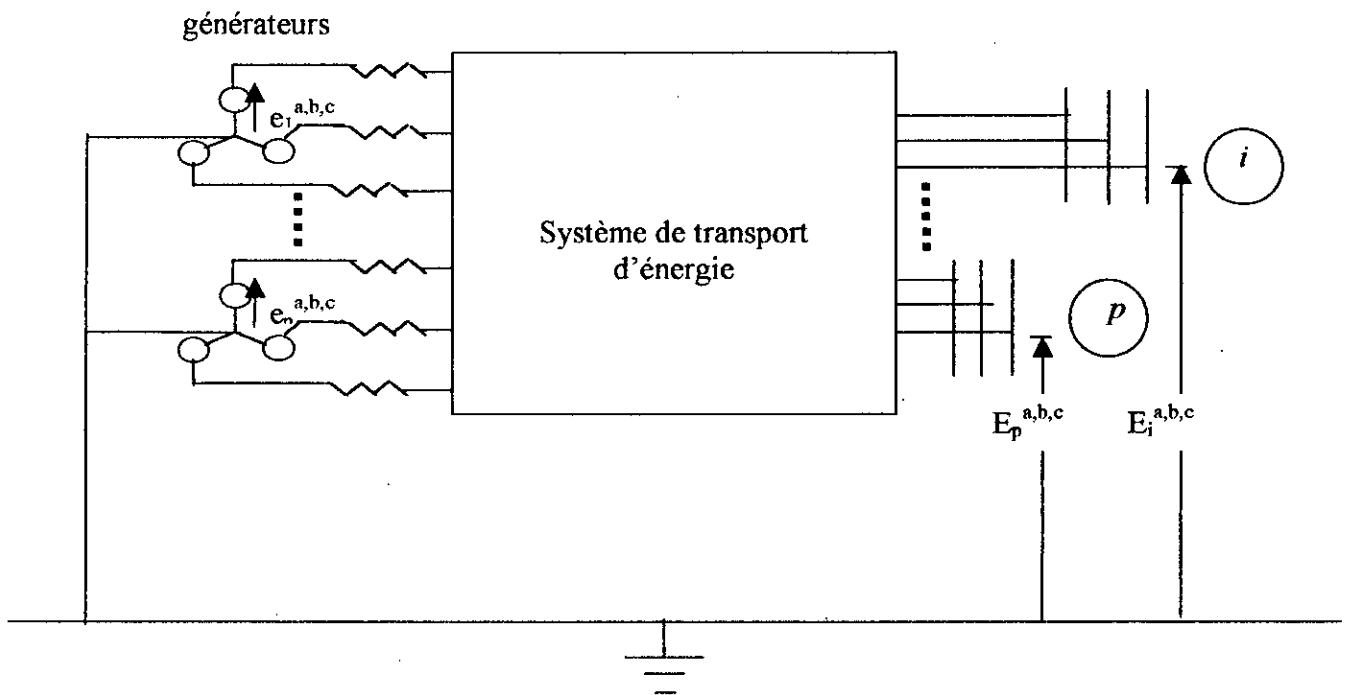


Figure (2.2)

II.3.2. Courants et tensions de défaut :

L'utilisation de la matrice impédance nodale fournit un moyen pratique pour le calcul des courants et les tensions du court-circuit quand la terre est sélectionnée comme référence [5]. Un des avantages distincts est que, une fois la matrice impédance des nœuds formée, les éléments de cette matrice peuvent être utilisés pour calculer les courants et tensions associés à différents défauts et leurs emplacements directement. La représentation du réseau avec un défaut au nœud p est montré par Fig. (2.3). Dans cette représentation dérivée du théorème de Thevenin, l'impédance interne est représentée par la matrice impédance des nœuds y compris les réactances des machines, et les tensions des circuits ouverts sont représentées par les tensions des nœuds autres que celui de défaut.

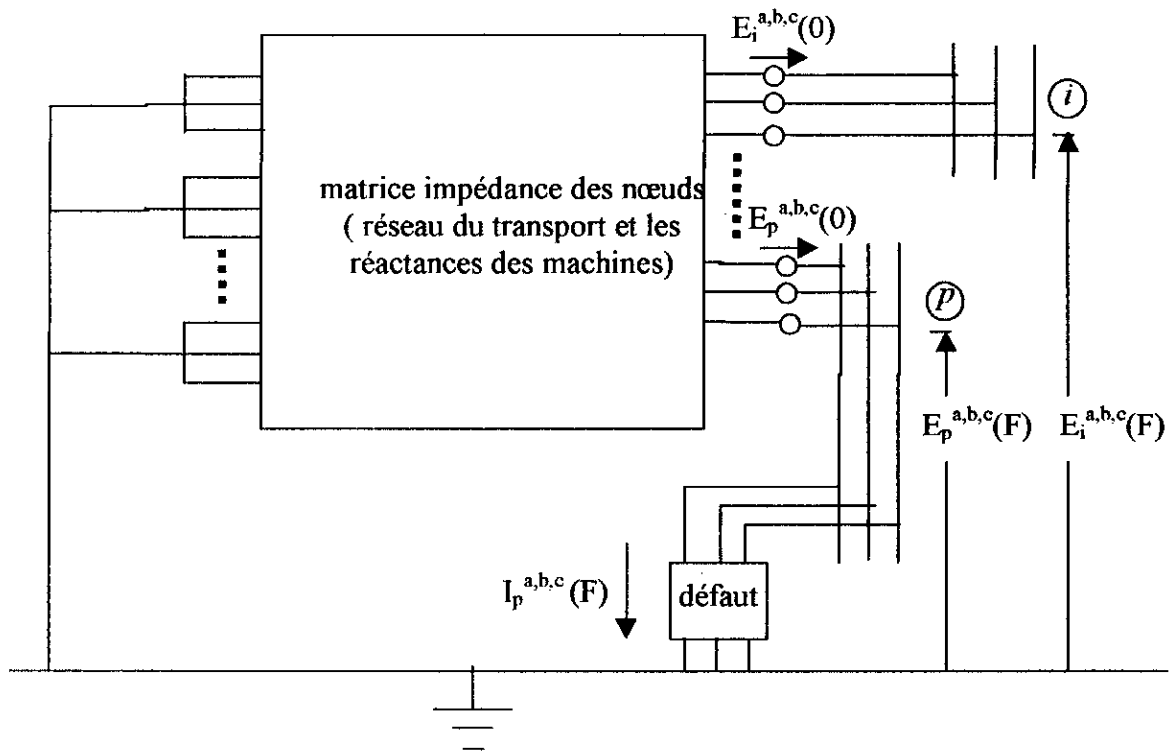


Figure (2.3)

L'équation (2.1) caractérisant le réseau pendant le défaut s'écrit :

$$\bar{E}_{BUS}^{a,b,c}(F) = \bar{E}_{BUS}^{a,b,c}(0) - Z_{BUS}^{a,b,c} I_{BUS}^{a,b,c}(F) \tag{2.1}$$

Les tensions après défaut de tous les nœuds ont pour expression :

$$E_{BUS}^{a,b,c}(F) = (E_1^{a,b,c}(F) \dots E_p^{a,b,c}(F) \dots E_n^{a,b,c}(F))$$

Où $E_{BUS}^{a,b,c}(F)$ vecteur des tensions triphasées $E_i^{a,b,c}(F)$ $i=1, \dots, n$.

et $E_{BUS}^{a,b,c}(0)$ vecteur des tensions triphasées avant défaut.

$$Z_{BUS}^{a,b,c} = \begin{bmatrix} Z_{11}^{a,b,c} & \dots & \dots & Z_{1n}^{a,b,c} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ Z_{n1}^{a,b,c} & \dots & \dots & Z_{nn}^{a,b,c} \end{bmatrix}$$

Le vecteur des courants de défaut au nœud p s'écrit :

$$I_{BUS}^{a,b,c}(F) = (0 \dots 0 I_p^{a,b,c}(F) 0 \dots 0)$$

L'équation (1.1), développée donne le système suivant :

$$\begin{cases} E_1^{a,b,c}(F) = E_1^{a,b,c}(0) - Z_{1p}^{a,b,c} I_p^{a,b,c}(F) \\ E_2^{a,b,c}(F) = E_2^{a,b,c}(0) - Z_{2p}^{a,b,c} I_p^{a,b,c}(F) \\ E_p^{a,b,c}(F) = E_p^{a,b,c}(0) - Z_{pp}^{a,b,c} I_p^{a,b,c}(F) \end{cases} \quad (2.2)$$

A partir de la figure(2.3), le vecteur de tension au nœud de défaut p s'écrit :

$$E_p^{a,b,c}(F) = Z_F^{a,b,c} I_p^{a,b,c}(F) \quad (2.3)$$

où $Z_f^{a,b,c}$ est la matrice impédance du défaut triphasée. Les éléments de cette matrice de dimension 3×3 dépendent du type de défaut et de l'impédance du défaut. Après Substitution de l'équation (2.3) la p^{eme} équation du système (1.2) donne :

$$Z_F^{a,b,c} I_p^{a,b,c}(F) = E_p^{a,b,c}(0) - Z_{pp}^{a,b,c} I_p^{a,b,c}(F) \quad (2.4)$$

La résolution de l'équation (2.4) pour l'inconnu $I_p^{a,b,c}(F)$, donne :

$$I_p^{a,b,c}(F) = (Z_F^{a,b,c} + Z_{pp}^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad (2.5)$$

Après substitution de $I_p^{a,b,c}(F)$ dans l'équation(2.3), la tension triphasée au nœud de défaut devient :

$$E_p^{a,b,c}(F) = (Z_F^{a,b,c} + Z_{pp}^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad (2.6)$$

De manière similaire, les tensions triphasées des autres nœuds peuvent être obtenues par la substitution de $I_p^{a,b,c}(F)$ dans (2.5) :

$$E_i^{a,b,c}(F) = E_i^{a,b,c}(0) - Z_{ip}^{a,b,c} (Z_F^{a,b,c} + Z_{pp}^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad i \neq p \quad (2.7)$$

Connaissant les paramètres du circuit du défaut sous forme admittance (matrice admittance du défaut) le courant triphasé de court-circuit s'écrit :

$$J_p^{a,b,c}(F) = Y_F^{a,b,c} E_p^{a,b,c}(F) \quad (2.8)$$

où $Y_F^{a,b,c}$ la matrice admittance du défaut, triphasée.

Substituant $J_p^{a,b,c}(F)$ de l'équation (2.8) dans l'équation (2.5), la $p^{\text{ème}}$ équation du système (2.2) devient :

$$E_p^{a,b,c}(F) = E_p^{a,b,c}(0) - Z_{pp}^{a,b,c} Y_F^{a,b,c} E_p^{a,b,c}(F) \quad (2.9)$$

La résolution de l'équation (2.9) $E_p^{a,b,c}(F)$ donne :

$$E_p^{a,b,c}(F) = (U + Z_{pp}^{a,b,c} Y_F^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad (2.10)$$

U : matrice identité.

Substituant $E_p^{a,b,c}(F)$ dans l'équation (2.9), la tension triphasé au nœud de défaut p s'écrit :

$$E_p^{a,b,c}(F) = (U + Z_{pp}^{a,b,c} Y_F^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad (2.11)$$

De même, les tensions triphasées des autres nœuds peuvent être obtenues par la substitution de $J_p^{a,b,c}(F)$ de l'équation (2.11) :

$$E_i^{a,b,c}(F) = E_i^{a,b,c} - Z_{ip}^{a,b,c} Y_F^{a,b,c} (U + Z_{pp}^{a,b,c} Y_F^{a,b,c})^{-1} E_p^{a,b,c}(0) \quad i \neq p \quad (2.12)$$

Les courants de défaut traversant les éléments ij du réseau peuvent être calculés à partir des tensions obtenues dans les équations (2.6) et (2.7) ou (2.10) et (2.12). Ces courants sont fonction des tensions aux bornes de l'élément considéré. On peut écrire :

$$\bar{i}^{a,b,c}(F) = [y^{a,b,c}] \bar{v}^{a,b,c}(F)$$

$$\text{où: } i_{ij}^{a,b,c}(F) = \begin{bmatrix} i_{ij}^a(F) \\ i_{ij}^b(F) \\ i_{ij}^c(F) \end{bmatrix}$$

$$v_{ij}^{a,b,c} = \begin{bmatrix} v_{ij}^a(F) \\ v_{ij}^b(F) \\ v_{ij}^c(F) \end{bmatrix}$$

$$\text{et } y_{ij,kl}^{a,b,c} = \begin{bmatrix} y_{ij,kl}^{aa} & y_{ij,kl}^{ab} & y_{ij,kl}^{ac} \\ y_{ij,kl}^{ba} & y_{ij,kl}^{bb} & y_{ij,kl}^{bc} \\ y_{ij,kl}^{ca} & y_{ij,kl}^{cb} & y_{ij,kl}^{cc} \end{bmatrix}$$

Avec $y_{ij,kl}^{bc}$ admittance mutuelle entre la phase b de l'élément ij et la phase c de l'élément kl du réseau.

Le courant triphasé dans l'élément ij du réseau est calculé à partir de l'équation suivante :

$$i_{ij}^{a,b,c}(F) = \bar{y}_{ij,\rho\sigma}^{a,b,c} v_{\rho\sigma}^{a,b,c}(F) \quad (2.13)$$

Dans le cas couplage mutuel entre l'élément $i-j$ et l'élément $\rho\sigma$ du réseau, nous avons :

$$\bar{v}_{\rho\sigma}^{a,b,c}(F) = \bar{E}_{\rho}^{a,b,c} - \bar{E}_{\sigma}^{a,b,c}$$

et l'équation (2.13) devient :

$$i_{ij}^{a,b,c}(F) = y_{ij,\rho\sigma} \left(\bar{E}_{\rho}^{a,b,c}(F) - \bar{E}_{\sigma}^{a,b,c}(F) \right) \quad (2.14)$$

II.4. Calcul de court-circuit par la méthode des composantes symétriques :

Les relations présentées dans la section précédente utilisent les composantes par phase des éléments. En utilisant ces relations, le calcul de court-circuit des régimes déséquilibrés devient assez compliqué. Certes on peut toujours appliquer les lois de Kirchoff et d'Ohm, mais il faut écrire les équations correspondantes pour chacun des conducteurs de phases en faisant intervenir, non seulement les variations de tensions dues aux résistances et aux inductances propres d'une phase, mais aussi celles dues aux mutuelles inductances entre cette phase et les autres. Chaque équation contient alors les courants des autres phases. Pour simplifier ce genre de calculs, on utilise la méthode des composantes symétriques, qui généralement permet une mise en équation et une résolution bien plus aisée que la méthode précédente. Cette méthode repose essentiellement sur la propriété suivante [6] :

« Tout système de grandeurs sinusoïdales q -phasés non équilibrées peut être décomposé en q systèmes q -phasés équilibrés, que l'on appelle les composantes symétriques du système non équilibré ».

II.4. 1. Transformation par les composantes symétriques :

La matrice impédance primitive pour un réseau triphasé équilibré est donnée par :

$$Z_{pq}^{a,b,c} = \begin{bmatrix} Z_{pq}^s & Z_{pq}^{ms} & Z_{pq}^m \\ Z_{pq}^{ms} & Z_{pq}^s & Z_{pq}^{ms} \\ Z_{pq}^m & Z_{pq}^{ms} & Z_{pq}^s \end{bmatrix}$$

On peut diagonaliser cette matrice par la transformation $(T_s^*)^Y Z_{pq}^{a,b,c} T_s$ pour obtenir :

$$Z_{pq}^{0,1,2} = \begin{bmatrix} Z_{pq}^{(0)} & & \\ & Z_{pq}^{(1)} & \\ & & Z_{pq}^{(2)} \end{bmatrix}$$

où $z_{pq}^{(0)}$, $z_{pq}^{(1)}$ et $z_{pq}^{(2)}$ sont les séquences impédances homopolaire, directe et inverse, respectivement. Les séquences impédances directe et inverse sont égales pour un réseau triphasé équilibré stationnaire [4]. De plus, on admet que les séquences impédances directe et inverse sont égales dans le cas des éléments rotationnels pour les calculs de courts-circuits. Chaque matrice primitive $y_{ij,k}^{a,b,c}$ ou $Z_{ij}^{a,b,c}$ peut aussi être diagonalisée par la matrice de transformation T_s pour obtenir respectivement :

$$y_{ij,k}^{0,1,2} = \begin{bmatrix} y_{ij,k}^{(0)} & & \\ & y_{ij,k}^{(1)} & \\ & & y_{ij,k}^{(2)} \end{bmatrix} \quad \text{ou} \quad Z_{ij}^{0,1,2} = \begin{bmatrix} Z_{ij}^{(0)} & & \\ & Z_{ij}^{(1)} & \\ & & Z_{ij}^{(2)} \end{bmatrix}$$

On admet aussi que les tensions de tous les nœuds avant le court-circuit sont égales en amplitude et en phase. On suppose que l'amplitude de tension avant défaut est de 1 pu .

La tension avant défaut au nœud i est :

$$E_i^{a,b,c} = \begin{bmatrix} 1 \\ a^2 \\ a \end{bmatrix}$$

La transformation en composantes symétriques nous donne :

$$E_i^{0,1,2}(0) = (T_s^*) E_i^{a,b,c}(0) = \begin{bmatrix} 0 \\ \sqrt{3} \\ 0 \end{bmatrix}$$

La matrice impédance de défaut $Z_F^{a,b,c}$ peut se transformer en utilisant T_s à $Z_F^{0,1,2}$. La matrice résultante est une matrice diagonale si le défaut est équilibré. Les matrices impédance et admittance de défaut pour les différents types de courts-circuits sont données par le tableau (2.1) [4]. Les équations des courants et des tensions de court-circuit peuvent également s'écrire sous forme de composantes symétriques.

Le courant au nœud de défaut p est :

$$I_p^{0,1,2}(F) = (Z_F^{0,1,2} + Z_{pp}^{0,1,2})^{-1} E_p^{0,1,2}(0) \quad (2.15)$$

ou bien

$$I_p^{0,1,2}(F) = Y_F^{0,1,2} \left(U + Z_{pp}^{0,1,2} Y_F^{0,1,2} \right)^{-1} E_p^{0,1,2} \quad (2.16)$$

La tension au nœud de défaut p est

$$E_p^{0,1,2}(F) = Z_F^{0,1,2} \left(Z_F^{0,1,2} + Z_{pp}^{0,1,2} \right)^{-1} E_p^{0,1,2}(0) \quad (2.17)$$

ou bien

$$E_p^{0,1,2}(F) = \left(U + Z_{pp}^{0,1,2} Y_F^{0,1,2} \right)^{-1} E_p^{0,1,2}(0) \quad (2.18)$$

Les tensions aux autres nœuds sont données par :

$$E_i^{0,1,2}(F) = E_i^{0,1,2}(0) - Z_{ip}^{0,1,2} \left(Z_F^{0,1,2} + Z_{pp}^{0,1,2} \right)^{-1} E_p^{0,1,2} \quad (2.19)$$

ou bien

$$E_i^{0,1,2}(F) = E_i^{0,1,2}(0) - Z_{ip}^{0,1,2} Y_F^{0,1,2} \left(U + Z_{pp}^{0,1,2} Y_F^{0,1,2} \right)^{-1} E_p^{0,1,2}(0) \quad (2.20)$$

Le courant triphasé de défaut traversant l'élément ij est

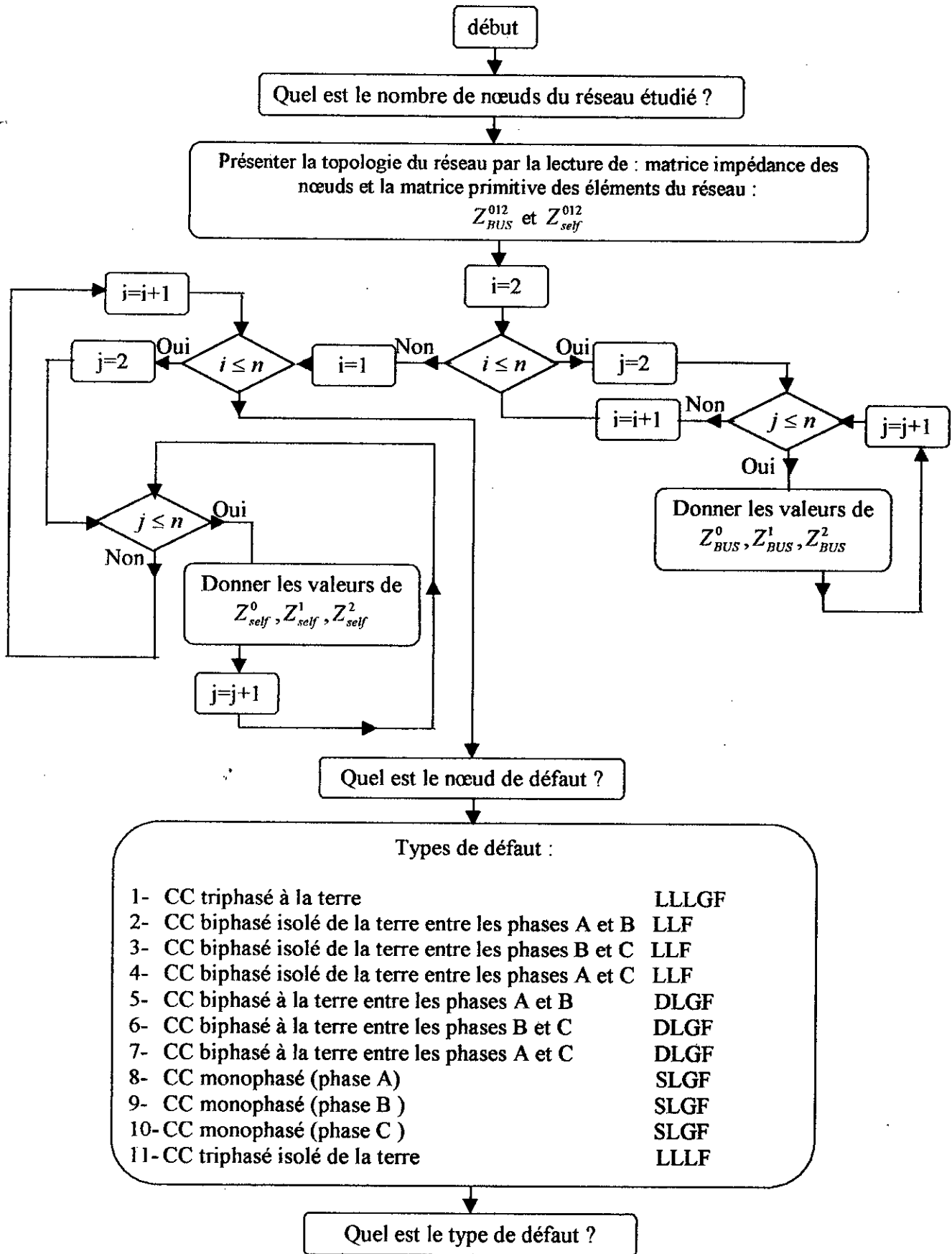
$$i_{ij}^{0,1,2}(F) = \bar{y}_{ij,\rho\sigma}^{0,1,2} \left(\bar{E}_\rho^{0,1,2}(F) - \bar{E}_\sigma^{0,1,2}(F) \right) \quad (2.21)$$

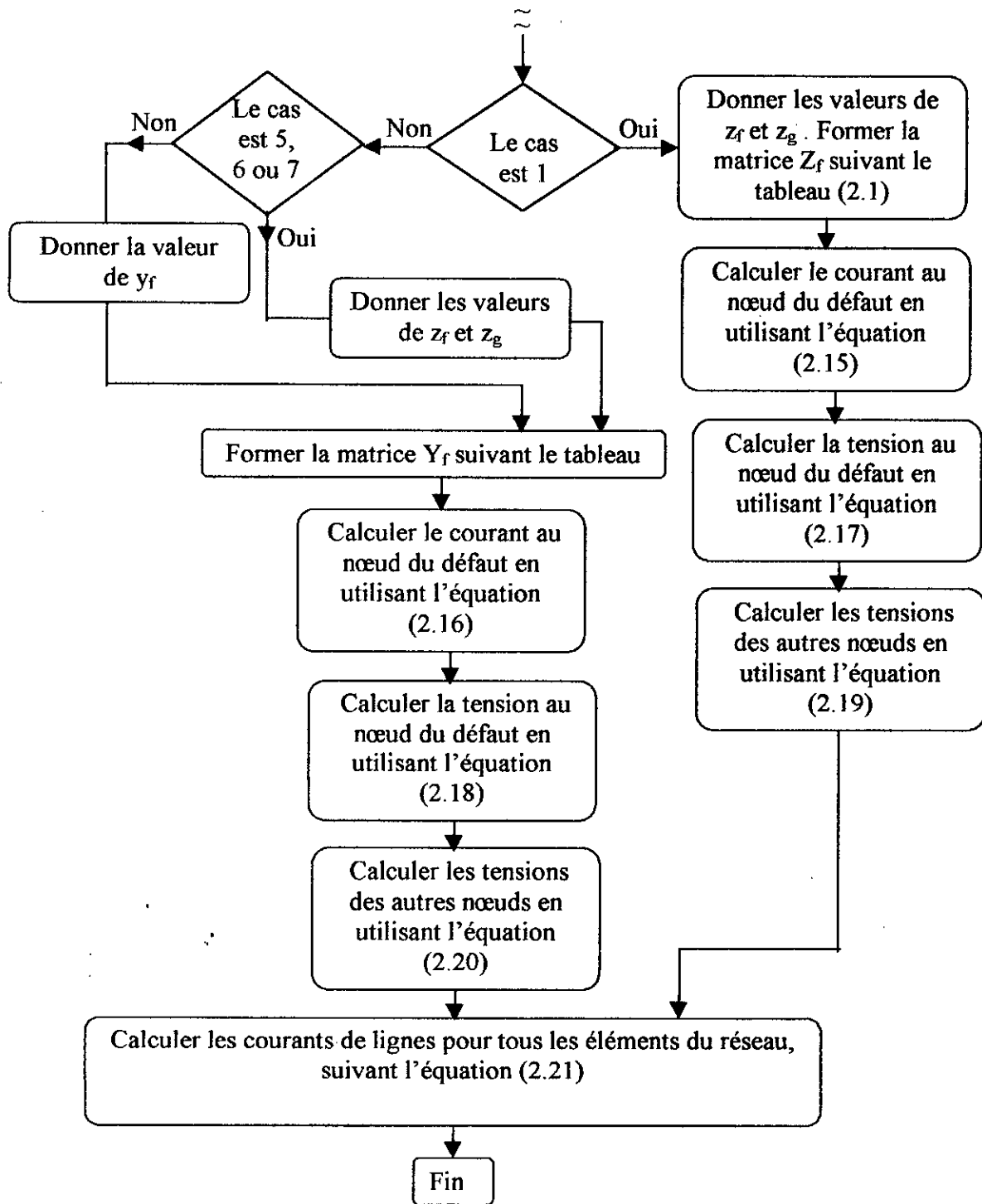
type de défaut	$Z_F^{0,1,2}$	$Y_F^{0,1,2}$
	$\begin{bmatrix} Z_F + 3z_g & 0 & 0 \\ 0 & Z_F & 0 \\ 0 & 0 & Z_F \end{bmatrix}$	
		$\frac{y_F}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -a \\ 0 & -a^2 & 1 \end{bmatrix}$
		$\frac{y_F}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$
		$\frac{y_F}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -a^2 \\ 0 & -a & 1 \end{bmatrix}$
		$\frac{1}{3k} \begin{bmatrix} 2z_F & -az_F & -a^2z_F \\ 2z_F & -az_F + 3z_g & -a^2z_F - 3a^2z_g \\ 2z_F & -3a^2z_g - az_F & 3z_g - a^2z_F \end{bmatrix}$
		$\frac{1}{3k} \begin{bmatrix} 2z_F & -z_F & -z_F \\ -z_F & 2z_F + 3z_g & -a^2z_F - 3z_g \\ -z_F & -3z_g - z_F & 3z_g - z_F \end{bmatrix}$
		$\frac{1}{3k} \begin{bmatrix} 2z_F & -a^2z_F & -az_F \\ 2z_F & -a^2z_F + 3z_g & -az_F - 3a^2z_g \\ 2z_F & -3az_g - a^2z_F & 3z_g - az_F \end{bmatrix}$
		$\frac{y_F}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
		$\frac{y_F}{3} \begin{bmatrix} 1 & a^2 & a \\ a & 1 & a^2 \\ a^2 & a & 1 \end{bmatrix}$
		$\frac{y_F}{3} \begin{bmatrix} 1 & a & a^2 \\ a^2 & 1 & a \\ a & a^2 & 1 \end{bmatrix}$
		$y_F \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

avec $k = z_F(z_F + 2z_g)$

Tableau 2.1 : Les matrices admittance et impédance du défaut.

II.5. Organigramme de calcul de court-circuit :





II.6. Exemple de calculs de court-circuit :

A titre d'exemple, et pour présenter la fiabilité de calcul de notre méthode on peut faire le calcul pour un réseau donné.

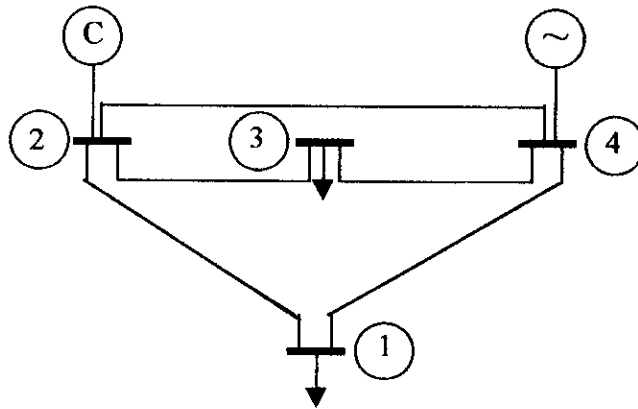


Figure (2.4)

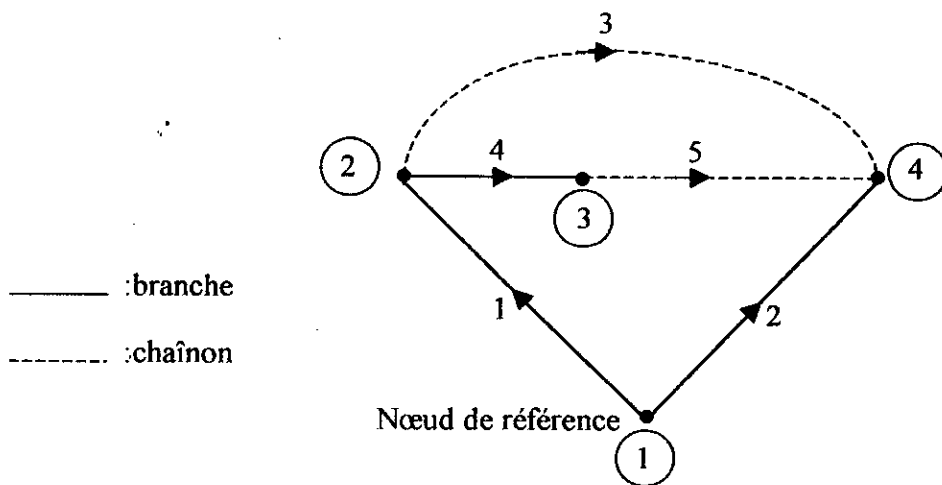


Figure (2. 5)

Les données :

		élément <i>ij</i>	Impédance Z_{self}^{012}		
1	1-2		0.035		
				0.1025	
					0.1025
2	1-3		0.035		
				0.1025	
					0.1025
3	2-3		2.50		
				1.00	
					1.00
4	2-4		1.00		
				0.40	
					0.40
5	4-3		1.50		
				0.60	
					0.60

	.0344			0.0006			0.0209		
		0.0876				0.0149			0.0586
			0.0876				0.0149		
$Z_{BUS}^{012} =$	0.0006			0.0344				0.0141	
		0.0149				0.0876			0.0439
			0.0149				0.0876		
	0.0209				0.0141			0.6182	
		0.0586				0.0439			0.2928
			0.0586				0.0439		
								0.0439	0.2928

On a fait le calcul de court-circuit pour deux cas :

- 1- Un court-circuit triphasé au nœud 4 (LLLF) ;
- 2- Un court-circuit monophasé à la terre de la phase A du nœud 4 (SLGF) ;

Résultats de calcul :

1^{er} cas :

Le choix est 11 : court circuit triphasé au nœud 4.

Le courant au nœud de défaut :

	Les composantes symétriques	Les composantes réelles
0,A	(0.000000,0.000000)	(3.415289,0.000000)
1,B	(5.915453,0.000000)	(0.000000E,0.000000)
2,C	(0.000000,0.000000)	(-1.707644,-2.957727)

Les tensions des différents nœuds :

		Les composantes symétriques	Les composantes réelles
Nœud 2	0,A	(0.000000,0.000000)	(7.998641E-01,0.000000)
	1,B	(1.385405,0.000000)	(-3.9993 ^E -01,-6.927026 ^E -01)
	2,C	(0.000000,0.000000)	(-3.99932 ^E -01,6.927026 ^E -01)
Nœud 3	0,A	(0.000000,0.000000)	(8.500689 ^E -01,0.000000)
	1,B	(1.472362,0.000000)	(-4.25034 ^E -01,-7.36181 ^E -01)
	2,C	(0.000000,0.000000)	(-4.25034 ^E -01,7.36181 ^E -01)
Nœud 4	0,A	(0.000000,0.000000)	(3.415289 ^E -06,0.000000)
	1,B	(5.915454 ^E -06,0.000000)	(-1.70764 ^E -06,-2.95772 ^E -06)
	2,C	(0.000000,0.000000)	(-1.707644 ^E -06,2.95772 ^E -06)

Calcul des courants de lignes :

		Les composantes symétriques	Les composantes réelles
I=2	0,A	(0.000000,0.000000)	(4.34786 ^E -02,-7.530716 ^E -02)
	J=3		
	1,B	(-8.695722 ^E -02,0.000000)	(4.34786 ^E -02,7.530716 ^E -02)
	2,C	(0.000000,0.000000)	(-8.695722 ^E -02,0.000000)
I=2	0,A	(0.000000,0.000000)	(-1.731749,2.999477)
	J=4		
	1,B	(3.463498,0.000000)	(-1.731749,-2.999477)
	2,C	(0.000000,0.000000)	(3.463498,0.000000)
I=4	0,A	(0.000000,0.000000)	(1.226964,-2.125163)
	J=3		
	1,B	(-2.453927,0.000000)	(1.226964,2.125163)
	2,C	(0.000000,0.000000)	(-2.453927,0.000000)

2^{eme} cas :

Le choix est 8 : court circuit monophasé dans la phase A au nœud 4.

Le courant au nœud de défaut :

	Les composantes symétriques	Les composantes réelles
0,A	(1.438372,0.000000)	(2.491333,0.000000)
1,B	(1.438372,0.000000)	(2.474917 ^E -08,0.000000)
2,C	(1.438372,0.000000)	(3.441276 ^E -08,0.000000)

Les tensions des différents nœuds :

		Les composantes symétriques	Les composantes réelles
Nœud 2	0,A	(-3.006197 ^E -02,0.000000)	(8.853157 ^E -01,0.000000)
	1,B	(1.647762,0.000000)	(-4.68692 ^E -01,-8.66025 ^E -01)
	2,C	(-8.428859 ^E -02,0.000000)	(4.68692 ^E -01,8.660253 ^E -01)
Nœud 3	0,A	(-2.028104 ^E -02,0.000000)	(9.153777 ^E -01,0.000000)
	1,B	(1.668906,0.000000)	(-4.752527 ^E -01,8.66025 ^E -01)
	2,C	(-6.314452 ^E -02,0.000000)	(-4.75252 ^E -01,8.660253 ^E -01)
Nœud 4	0,A	(-8.892142 ^E -01,0.000000)	(2.491656 ^E -04,0.000000)
	1,B	(1.310848,0.000000)	(-7.70206 ^E -01,8.660253 ^E -01)
	2,C	(-4.212025 ^E -01,0.000000)	(-7.70206 ^E -01,8.660253 ^E -01)

Calcul des courants de lignes :

		Composantes symétriques	Composantes réelles
I=2	0,A	(-2.114407 ^{E-02} ,0.000000)	(1.723168 ^{E-02} ,3.16649 ^{E-08})
J=3	1,B	(-2.114403 ^{E-02} ,0.000000)	(1.723168 ^{E-02} ,3.16649 ^{E-08})
	2,C	(-3.912317 ^{E-03} ,0.000000)	(-4.620047 ^{E-02} ,0.000000)
I=2	0,A	(8.422848 ^{E-01} ,0.000000)	(1.68674 ^{E-02} ,5.96046 ^{E-08})
J=4	1,B	(8.422848 ^{E-01} ,0.000000)	(1.68674 ^{E-02} ,5.96046 ^{E-08})
	2,C	(8.591522 ^{E-01} ,0.000000)	(2.543722,0.000000)
I= 4	0,A	(-5.967633 ^{E-01} ,0.000000)	(1.477444 ^{E-02} , 0.000000)
J=3	1,B	(-5.967633 ^{E-01} ,0.000000)	(1.477444 ^{E-02} , 0.000000)
	2,C	(-5.792888 ^{E-01} ,0.000000)	(-1.772815,0.000000)

II.7. Régimes transitoires :

Nous nous proposons ici d'étudier les régimes transitoires des courants de courts-circuits pour les différents types de défauts.

La transformation de Park rend assez aisée l'étude des régimes transitoires conservant la symétrie de construction des enroulements statoriques (les trois phases du stator sont soit ouvertes ou en court-circuit ou débitent sur trois impédances égales ou sont reliées à un réseau puissant soit directement soit par trois impédances égales).

Quand la symétrie de construction du stator n'est pas conservée, l'étude des régimes transitoires est beaucoup plus compliquée [7].

II.7.1. Court-circuit triphasé (LLLF) :

Un alternateur bipolaire à vide, tournant à la vitesse ω , ayant son inducteur alimenté par une tension constante V_F origine d'une *f.e.m* E égale à $M_F \omega V_F / R_F \sqrt{3}$, a ses trois bornes statoriques brusquement réunies à l'instant $t=0$.

Le courant dans la phase A s'écrit [7] :

$$i_A = -E_m \left[\frac{1}{X_d} + \left(\frac{1}{X'_d} - \frac{1}{X_d} \right) e^{-t/T'_a} + \left(\frac{1}{X''_d} - \frac{1}{X'_d} \right) e^{-t/T''_a} \right] \cos(\omega t + \varphi) + E_m e^{-t/T_a} \left[\frac{1}{2} \left(\frac{1}{X'_d} + \frac{1}{X''_d} \right) \cos \varphi + \frac{1}{2} \left(\frac{1}{X''_d} - \frac{1}{X'_d} \right) \cos(2\omega t + \varphi) \right] \quad (2.22)$$

Les expressions des courants i_B et i_C dans les deux autres phases se déduisent de celle de i_A en remplaçant φ par $\varphi - \frac{2\pi}{3}$ et par $\varphi - \frac{4\pi}{3}$.

L'expression du courant de court-circuit est la somme de cinq termes qu'on peut regrouper en trois composantes :

- La composante alternative fondamentale de pulsation ω .
- La composante asymétrique, terme aperiodique de valeur initiale $E_m \cos \varphi (1/X'_d + 1/X''_d)$, amortie avec la constante de temps T_a .
- La composante du second harmonique de pulsation 2ω , d'amplitude initiale $E_m (1/X''_d - 1/X'_d)$ amortie avec la constante de temps T_a .

II.7.2. Court-circuit biphasé isolé (LLF) :

L'alternateur à vide ayant ses phases sièges de *f.e.m* d'amplitude E_m et de pulsation ω , à l'instant $t=0$, où l'axe de l'inducteur fait un angle φ avec celui de la phase A , on réunit brusquement les phases B et C .

Le courant de court-circuit s'écrit [7] :

$$i_B = -i_C = \frac{-\sqrt{3}E_m \left[g_1(t) \sin(\omega t + \varphi) - \sin \varphi e^{-t/T_{a1}} \right]}{X''_d + X''_q + (X''_q - X''_d) \cos 2(\omega t + \varphi)} \quad (2.23)$$

avec

$$g_1(t) = \frac{X''_d + X_{m1}}{X_d + X_{m1}} + \left(\frac{X''_d + X_{m1}}{X'_d + X_{m1}} - \frac{X''_d + X_{m1}}{X_d + X_{m1}} \right) e^{-t/T'_{d1}} + \left(1 - \frac{X''_d + X_{m1}}{X'_d + X_{m1}} \right) e^{-t/T''_{d1}}$$

$$\text{où } X_{m1} = \sqrt{X_d'' X_q''}$$

$$T_{d1}' = \frac{X_d' + X_{m1}}{X_d' + X_{m1}} T_{d0}', \quad T_{d1}'' = \frac{X_d'' + X_{m1}}{X_d' + X_{m1}} T_{d0}'', \quad T_{d1} = \frac{X_{m1}}{\omega R_s}$$

II.7.3. Court-circuit monophasé (SLGF):

L'alternateur à vide produisant des *f.e.m* d'amplitude E_m et de pulsation ω , à l'instant $t=0$ tel que $\theta_0 = \varphi$, on relie la phase A à la terre.

Le courant de court-circuit i_A est donné par [7] :

$$i_A = \frac{-3E_m \left[g_2(t) \cos(\omega t + \varphi) - \cos \varphi e^{-t/T_{d2}} \right]}{X_d'' + X_q'' - (X_q'' - X_d'') \cos 2(\omega t + \varphi)} \quad (2.24)$$

avec

$$g_2(t) = \frac{X_d'' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} + \left(\frac{X_d'' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} - \frac{X_d'' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} \right) e^{-t/T_{d2}} \\ + \left(1 - \frac{X_d'' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} \right) e^{-t/T_{d2}}$$

où

$$X_{m2} = \sqrt{\left(X_d'' + \frac{1}{2} X_0' \right) \left(X_q'' + \frac{1}{3} X_0 \right)} - \frac{1}{2} X_0$$

$$X_0 = \omega(L_{s20} + 2M_{s0})$$

$$T_{d2}' = \frac{X_d' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} T_{d0}'$$

$$T_{d2}'' = \frac{X_d'' + X_{m2} + X_0}{X_d' + X_{m2} + X_0} T_{d0}''$$

$$T_{A2} = \frac{2X_{m2} + X_0}{\omega(2R_s + R_0)}$$

II.7.4. Court-circuit biphasé à la terre (LLGF):

Si, avec les mêmes conditions initiales que précédemment, à $t=0$, on relie simultanément les bornes B et C à la terre, on obtient [7] :

$$\begin{aligned}
 i_b = & \frac{\sqrt{3} E}{2 D} g_3(t) \left[\sqrt{3} X_q'' \cos \theta - (X_q'' + 2X_0) \sin \theta \right] \\
 & - \frac{\sqrt{3} E}{4 D} \cos \varphi e^{\frac{-t}{T_{d3}}} \left\{ \sqrt{3} [X_d'' + X_q'' - (X_q'' - X_d'') \cos 2\theta] + (X_q'' - X_d'') \sin 2\theta \right\} \\
 & + \frac{\sqrt{3} E}{4 D} \sin \varphi e^{\frac{-t}{T_{d4}}} \left[X_d'' + X_q'' + 4X_0 - (X_q'' - X_d'') \cos 2\theta - \sqrt{3} (X_q'' - X_d'') \sin 2\theta \right]
 \end{aligned} \quad (2.25)$$

$$\begin{aligned}
 i_c = & \frac{\sqrt{3} E}{2 D} g_3(t) \left[\sqrt{3} X_q'' \cos \theta + (X_q'' + 2X_0) \sin \theta \right] \\
 & - \frac{\sqrt{3} E}{4 D} \cos \varphi e^{\frac{-t}{T_{d3}}} \left\{ \sqrt{3} [X_d'' + X_q'' + (X_q'' - X_d'') \cos 2\theta] - (X_q'' - X_d'') \sin 2\theta \right\} \\
 & + \frac{\sqrt{3} E}{4 D} \sin \varphi e^{\frac{-t}{T_{d4}}} \left[X_d'' + X_q'' + 4X_0 - (X_q'' - X_d'') \cos 2\theta + \sqrt{3} (X_q'' - X_d'') \sin 2\theta \right]
 \end{aligned} \quad (2.26)$$

En posant :

$$\theta = \omega t + \varphi ; D = X_d'' X_q'' + X_0 (X_d'' + X_q'') + X_0 (X_q'' + X_d'') \cos 2\theta$$

avec

$$g_3(t) = \frac{X_d' + X_e}{X_d' + X_e} + \left(\frac{X_d'' + X_e}{X_d' + X_e} - \frac{X_d'' + X_e}{X_d' + X_e} \right) e^{-t/T_{d3}'} + \left(1 - \frac{X_d'' + X_e}{X_d' + X_e} \right) e^{-t/T_{d3}''}$$

$$\text{où } X_e = \frac{X_0 X_{m3}}{X_0 + X_{m3}} ; X_{m3} = \frac{X_d'' X_q'' + \sqrt{X_d'' X_q'' (X_d'' + 2X_0)(X_q'' + 2X_0)}}{X_d'' + X_q'' + 2X_0}$$

et

$$T_{d3}' = \frac{X_d' + X_e}{X_d' + X_e} T_{d0}' ; T_{d3}'' = \frac{X_d'' + X_e}{X_d' + X_e} T_{d0}'' ; T_{A3} = \frac{X_{m3} + 2X_0}{\omega(R_s + 2R_0)} ; T_{A4} = \frac{X_{m3}}{\omega R_s}$$

II.8. Conclusion :

Les courts-circuits sont des incidents qu'il faut éliminer le plus vite possible ; c'est le rôle des protections dont la rapidité de fonctionnement est une des performances prioritaires.

Le temps d'élimination des défauts comprend deux composantes principales :

- Le temps de fonctionnement des protections ;
- Le temps d'ouverture des disjoncteurs.

L'utilisation de la méthode des composantes symétriques dans les calculs de courts-circuits, rend ce dernier plus simple, par rapport à l'utilisation de la méthode générale (méthode des composantes par phases).

Les calculs que nous avons fait, ont montré que le court-circuit triphasé est le cas le plus sévère (plus grand courant de court-circuit), pour dimensionner les différents éléments constitutifs d'un réseau électrique.

Chapitre 3

CPN

III.1. Introduction :

L'étude des réseaux de neurones date des années 1940-50 [2]. Héritière de la révolution scientifique qui mobilisait les esprits à cette période, cette discipline trouve ces sources à la fois dans les connaissances neurobiologiques de l'époque et dans un outil permettant d'explorer par simulation le comportement des modèles théoriques proposés, l'ordinateur. Cette décennie vit aussi plusieurs travaux importants (en particulier les réflexions sur la nature du raisonnement humain de John von Newomann, d'Alan Turing, de Alonzo Church, d'Allan newell et de Herbert Simon) qui galvanisèrent les recherches sur le cerveau et l'esprit humain et posèrent les fondations de ce qui allait devenir l'intelligence artificielle [2].

Les réseaux de neurones sont des assemblages fortement connectés d'unités de calcul, les neurones formels. Ces derniers ont pour origine un modèle du neurone biologique, dont ils ne retiennent d'ailleurs qu'une vision fort simplifiée. Le neurone comme toute cellule, est composé d'un corps (ou sagma), qui contient son noyau et où se déroulent les activités propres à sa vie cellulaire [8]. Cependant, il est aussi généralement doté d'un axone et de dendrites, structures spécialisées dans la communication avec les autres neurones.

Cette communication entre cellules nerveuses revêt plusieurs formes, et peut être fort complexe. L'impulsion nerveuse en est une des manifestations les plus spectaculaires : c'est une décharge électrique qui prend sa source à l'extrémité somatique de l'axone (le segment initial), et court le long de l'axone vers les terminaisons axonales. Là, elle affectera tous les neurones reliés au neurone générateur, par l'intermédiaire de jonction entre les terminaisons axonales et les autres cellules. Cette jonction est appelée synapse [2].

La synapse est un lieu d'interactions fort complexes où le signal électrique de l'impulsion nerveuse est converti en un signal biochimique. Des substances biochimiques appelées neurotransmetteurs [2] sont libérés par le terminal axonal pré-synaptique du neurone

émetteur et traversent rapidement la courte distance qui le sépare de la membrane cellulaire post-synaptique du neurone récepteur. Un des effets possibles des neurotransmetteurs est d'exciter les neurones qui la reçoivent ; si suffisamment d'impulsions excitatrices convergent en un temps suffisamment court vers un même neurone, celui-ci produira généralement à son tour une impulsion nerveuse et affectera ses voisins. L'effet inverse excite aussi : c'est l'inhibition [2]. Un neurone inhibant un autre neurone réduira sa tendance à produire des impulsions nerveuses.

Le premier modèle neuromimétique est dû à deux neurobiologistes américains, Warren McCulloch et Walter Pitts [2]. Ceux-ci proposèrent pour la première fois un modèle formel de la cellule nerveuse qui résumait en une formule simple l'effet d'une impulsion nerveuse sur un neurone. Mc Culloch et Pitts supposèrent que l'impulsion nerveuse était l'expression ou le résultat d'un calcul simple effectué par chaque neurone, et que c'est grâce à l'effet collectif d'un réseau de neurones interconnectés que naît la pensée. Au même moment, Donald Hebb proposait une hypothèse qui associe l'activité des neurones à l'efficacité des synapses qui les relient [2]. Ces deux théories, en décrivant formellement d'une part de comportement à court terme d'un réseau connexionniste en fonction des neurones qui le constituent et d'autre part son comportement à long terme en fonction de ses interconnexions, constituent les fondements du neuromimétisme.

Peu de temps après, Franck Rosenblatt et Bernard Widrow proposèrent indépendamment des modèles neuromimétiques (le perceptron et l'Adaline, respectivement) capables d'apprendre à résoudre des problèmes posés par l'expérimentateur [2]. Ces travaux donnèrent lieu à une intense activité de recherche.

Ce chapitre est composé de deux parties. Dans la première sont présentées les définitions principales et les notions de base des réseaux de neurones, ainsi que les différents types d'apprentissage.

Dans la deuxième partie, le réseau de neurones utilisé dans cette étude (le CPN) est présenté plus en détail.

Première partie :**III.2. Structure et fonctionnement d'un neurone artificiel :***Neurone formel :*

C'est le résultat d'analogie directe, qu'ont faite Mc Culloch et Pitts avec le neurone biologique. Il s'agit d'un corps cellulaire qui exécute une somme pondérée des entrées qui lui parviennent. Si cette somme dépasse un certain seuil, le neurone est activé, sinon le neurone est dit désactivé ou au niveau bas [8].

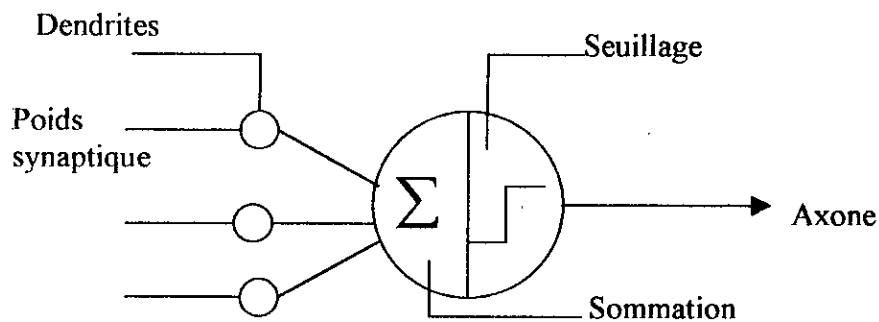


Figure (3.1) : Neurone formel

Élément linéaire adaptatif (Adaline) :

L'élément linéaire adaptatif, est le bloc de base dans l'architecture d'un réseau de neurone, voir Fig.(3.2), il est appelé ainsi car il permet une adaptation de ses poids synaptiques au vu d'un certain comportement.

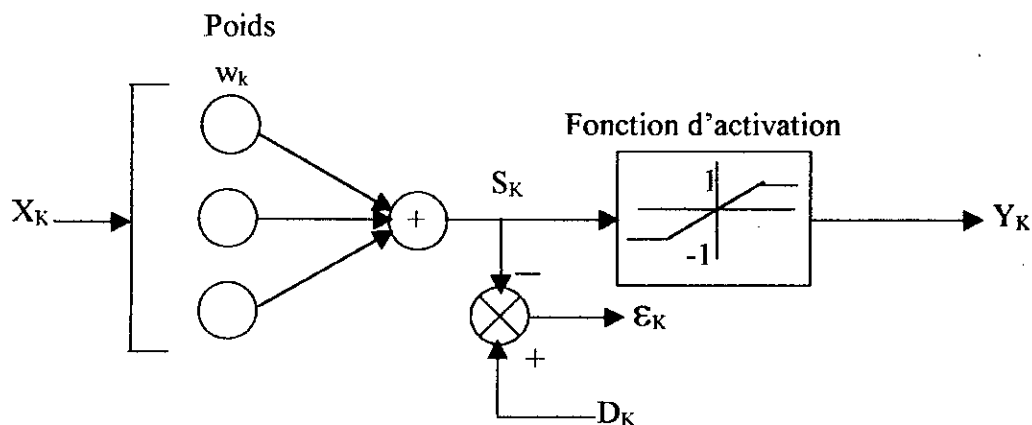


Figure (3.2) : élément linéaire adaptatif

Cet élément reçoit à l'instant K un vecteur X_k et une réponse désirée D_k dont on Véra le rôle dans le paragraphe de l'apprentissage, les composantes du vecteur d'entrée X_k sont pondérées à l'aide d'un vecteur poids W_k .

Après avoir calculer la somme pondérée, on applique à celle-ci une fonction dite d'activation. le vecteur d'entrée X_k peut contenir des composantes à valeurs continues ou binaires.

Dans tous les cas, la fonction d'activation $F(x)$ doit être saturable, pour éviter des valeurs en sortie trop élevées qui peuvent déstabiliser le réseau .

La remarque importante qu'on peut faire sur la fonction d'activation est son caractère non linéaire, ce caractère avantageux donne la possibilité au réseau de simuler des fonctions complexes, non linéaires.

III.3. Eléments constitutifs d'un réseau de neurones :

Un réseau de neurones est un graphe orienté et pondéré. Les nœuds de ce graphe sont des automates simples nommés neurones formels (ou unités connectionnistes) et dotés d'un état interne, l'activation, par lequel ils influencent les autres neurones du réseau. Cette activité se propage dans le graphe le long d'arcs pondérés appelés liens synaptiques ; par

extension, on appelle poids synaptiques la pondération d'un lien synaptique [3]. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ses pairs est appelée règle ou fonction d'activation. L'état du réseau entier est composé de l'activation de ses neurones constitutifs. Si l'on suppose que les N neurones d'un réseau sont indexés, cet état global peut être exprimé sous la forme d'un vecteur d'activation, dont les composantes sont les valeurs d'activation des neurones individuels :

$$A=(a_1 \ a_2 \ \dots \ a_N)$$

De même les liens synaptiques du réseau sont utilement décrits par la matrice des poids synaptiques, matrice carrée dont laquelle s'inscrivent les poids des liens :

$$W = \begin{bmatrix} w_{11} & \dots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} & \dots & w_{NN} \end{bmatrix}$$

En général, un neurone n'est relié qu'à un sous-ensemble d'autres neurones. En effet, la connectivité d'un réseau est rarement complète, mais obéit plutôt à des contraintes imposées lors de sa construction. Les liens manquants sont alors représentés dans la matrices des poids par un élément nul ; quand les contraintes de connectivité sont sévères, la matrice des poids résultante est creuse. Ainsi, les caractéristiques de la matrice résument bien la structure de connectivité du réseau. On nommera architecture la structure (nombre et types des neurones, connectivité) du réseau [2].

Choix des poids synaptiques :

Une fois l'architecture du réseau déterminée, il reste encore à choisir les valeurs des poids synaptiques de connexions que cette architecture a prescrites.

Souvent, ce choix est implicite dans les contraintes de connectivité choisies ; ainsi ; les neurones des modèles compétitifs doivent interagir par des liens essentiellement inhibiteurs. Dans d'autres cas, le réseau est suffisamment simple pour les déterminer directement, selon l'effet désiré. Cependant, les cas simples sont l'exception plutôt que la règle : un réseau de neurone comporte en général un grand nombre de connexions, qui ont de surcroît un rapport fort complexe avec le comportement du réseau.

En effet, la plupart des modèles neuromimétiques disposent de mécanismes capables de modifier leurs poids synaptiques automatiquement ; ils sont dotés de règles

d'apprentissage. Cette règle, ainsi que le choix de fonctions d'activations et de contraintes de connectivité, constituent les principaux éléments qui définissent un modèle neuromimétique.

Il résulte que le réseau possède deux dynamiques : celle de l'activité de ses neurones et celles des poids de ses liens. En général le changement d'activation du réseau est beaucoup plus rapide que l'évolution de ses poids synaptiques, et on suppose que la dynamique de l'activation est indépendante de la dynamique des poids (on dira que le changement de l'activation est adiabatique par rapport à celui des poids) [5].

Entrées et sorties du réseau :

Tout l'intérêt d'un réseau de neurones réside en sa capacité de résoudre des problèmes pratiques. Pour l'utiliser, il est donc nécessaire de communiquer au réseau les données du problème en question, et de lui soutirer le résultat de ses calculs.

Répartissons donc les neurones du réseau en trois ensembles E, S et C (voir la figure (3.3)) :

- L'ensemble E des neurones d'entrée. Appelons neurones d'entrée les neurones du réseau qui perçoivent les données du problème.
- L'ensemble S des neurones de sortie. Choisissons un sous-ensemble de neurones dont l'activation sera interprétée comme la réponse du réseau : ce sont les neurones de sortie.
- L'ensemble C des neurones cachés. Nommons enfin neurones cachés ceux qui ne sont ni à l'entrée ni à la sortie du réseau. Les neurones cachés n'ont aucun lien direct avec l'utilisateur et doivent donc agir par l'intermédiaire d'autres neurones.

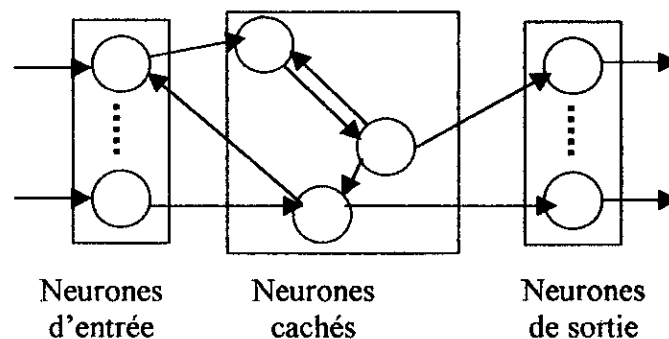


Figure (3.3) Un réseau de neurones

III.4. Apprentissage des réseaux de neurones :

On entend par apprentissage d'un réseau, l'opération qui consiste à modifier les connexions dans le réseau lors de la présentation d'un vecteur d'entrée à celui-ci, la modification des poids se poursuit jusqu'à ce que ces derniers ne varient que d'une façon infime. En effet, dans cette partie du travail, le réseau est entraîné pour être adapté à un type d'application. Ce travail d'apprentissage est réalisé à l'aide, d'algorithmes appropriés qu'on verra par la suite.

L'apprentissage d'un réseau se fait généralement dans le contexte d'une tâche ou d'un comportement à apprendre. Les informations à traiter sont codées sous la forme d'un vecteur appelé "patron d'entrée" [2], qui est communiqué aux neurones d'entrée du réseau. La réponse du réseau s'interprète à partir de la valeur d'activation de ces neurones de sortie, dont le vecteur s'appelle "patron de sortie" [2]. Lors d'un apprentissage supervisé, on dispose aussi du comportement de référence que doit apprendre le réseau, exprimé sous la forme de "patron de référence", ou "patron de sorties désirées" [2].

En général, l'apprentissage se fait sur une période relativement longue, durant laquelle les patrons d'entrées (et éventuellement de sorties désirées) peuvent être présentés au réseau un grand nombre de fois chacun. Cet apprentissage comprend quatre étapes de calcul :

- 1- Initialisation des poids synaptiques du réseau. En général, les poids trouvés par le réseau à la fin de l'apprentissage dépendent en partie de l'ensemble des poids dont il disposait au départ. La pratique courante est d'initialiser les poids du réseau à des (petites) valeurs aléatoires au début de l'apprentissage.
- 2- Présentation du patron d'entrée et propagation d'activation.
- 3- Calcul de l'erreur. Pour chaque neurone du réseau, une valeur d'erreur est calculée à partir de son activation et de celui des neurones qui lui sont reliés. Dans le cas d'un apprentissage supervisé, l'erreur tient aussi compte de la différence entre l'activation des neurones de sortie et de "patron de référence".
- 4- Calcul du vecteur de correction. A partir des valeurs d'erreur, on détermine alors la correction à apporter aux poids synaptiques des connexions et aux seuils des neurones. La correction effective des poids peut se faire après chaque présentation de patron. Le

nombre de patrons à présenter au réseau avant d'effectuer la correction s'appelle "fenêtre de mise à jour" [2].

Les étapes 2, 3 et 4 sont répétées jusqu'à la fin de l'apprentissage.

III.4.1. Types d'apprentissage :

Les techniques d'apprentissage des réseaux de neurones se repartissent en trois grandes familles [2] :

- L'apprentissage non supervisé ; modifie les poids du réseau en fonction d'un critère interne, indépendant de l'adéquation entre le comportement de réseau et la tâche qu'il doit effectuer.
- L'apprentissage supervisé ; au contraire, dispose d'un comportement de référence vers lequel il tente de faire converger le réseau.
- L'apprentissage semi-supervisé suppose qu'un comportement de référence précis n'est pas disponible, mais qu'en revanche il est possible d'obtenir des indications qualitatives (correct/incorrect ; par ex), ou lacunaires (une indication tout les N patrons par ex) sur les performances du réseau.

III.4.2. Problèmes d'apprentissage :

Ainsi, il est clair qu'un grand nombre de choix s'impose à celui qui voudrait entraîner un réseau de neurones. Chaque choix compte plus d'un écueil potentiel, dont par exemple :

- Insuffisance de la règle d'apprentissage. Plusieurs réseaux sont capables en principe de calcul élaboré. En revanche, rien nous garantit que la règle d'apprentissage soit capable de tirer profit de plein potentiel du réseau.
- Minima locaux. La majorités des apprentissages supervisés procédant par descente du gradient[8]. Or, nous avons vu que ce type de technique peut conduire à une solution sous optimale. En pratique, ce type du problèmes est particulièrement difficiles à éviter, notamment parce que la surface d'erreur n'est généralement pas connue [2]. Sachant que

les poids initiaux du réseau conditionnent en partie les résultats de l'apprentissage, plusieurs chercheurs relancent l'apprentissage à partir du choix des poids initiaux différents pour se préserver de ce problème.

- Mauvais choix de paramètres. Le comportement du réseau est gouverné par un ensemble de paramètres architecturant, pour lesquels un mauvais choix de valeurs peut compromettre l'apprentissage. Parmi ceci considérant par exemple le nombre de neurones cachés du réseau. Cette valeur est particulièrement importante parce qu'elle détermine le nombre de variables libres [2] (c'est à dire de poids) du réseau, et parce qu'elle détermine les capacités de calcul de réseau. Ainsi, un nombre insuffisant de neurones cachés peut compromettre la capacité du réseau à résoudre la tâche donnée. Inversement, trop de neurones cachés permettent au réseau d'apprendre les particularités de corpus d'apprentissage (d'apprendre les corpus par cœur), au détriment des performances de généralisation.

Les paramètres propres à la règle d'apprentissage sont souvent aussi difficiles à choisir que ceux de réseau [2]. Par exemple, la plupart des règles d'apprentissage comprennent un pas d'apprentissage, valeur qui affecte la taille du vecteur de correction. Une fois de plus, le bon choix de ce paramètre peut relever de l'exploit : un pas trop petit empêche le réseau de se stabiliser sur une solution, tandis qu'un pas trop grand augmente inutilement la quantité de calculs à effectuer.

- Surapprentissage. Quand l'apprentissage d'un réseau est prolongé, son choix de poids reflètent de trop près les particularités de corpus d'apprentissage, au détriment de la tâche réelle.
- Mauvais échantillonnage. Les corpus d'apprentissage ne reflètent pas toujours adéquatement la tâche, soit par mauvais échantillonnage, soit tout simplement parce qu'il est trop petits. Il en résulte que le réseau généralise mal et se comporte incorrectement quand il est mis en situation réelle.
- Oubli et interférence. Certains patrons peuvent interférer avec l'apprentissage d'autres patrons. Cela est particulièrement vrai lorsque le corpus d'apprentissage est grand, ou que le comportement à apprendre comprend plusieurs cas exceptionnels. Ces derniers sont alors difficiles à apprendre par le réseau et peuvent nécessiter un traitement particulier.
- Mauvais codage. Il se peut que la technique de codage choisie accule ou rende inutilement obscure l'information nécessaire à la résolution du problème.

Deuxième partie :

III.5. Compteur de propagation (CPN) :

Le CPN est une combinaison des différentes structures des réseaux de neurones artificiels (RNA). HECHT-NILSON a fait la synthèse de l'architecture du CPN par une combinaison des structures connues comme : les réseaux compétitifs et la structure de GROSSBERG de l'outstar [5]. L'architecture du réseau comme elle a publiée la première fois est montrée à la figure (3.4) [5].

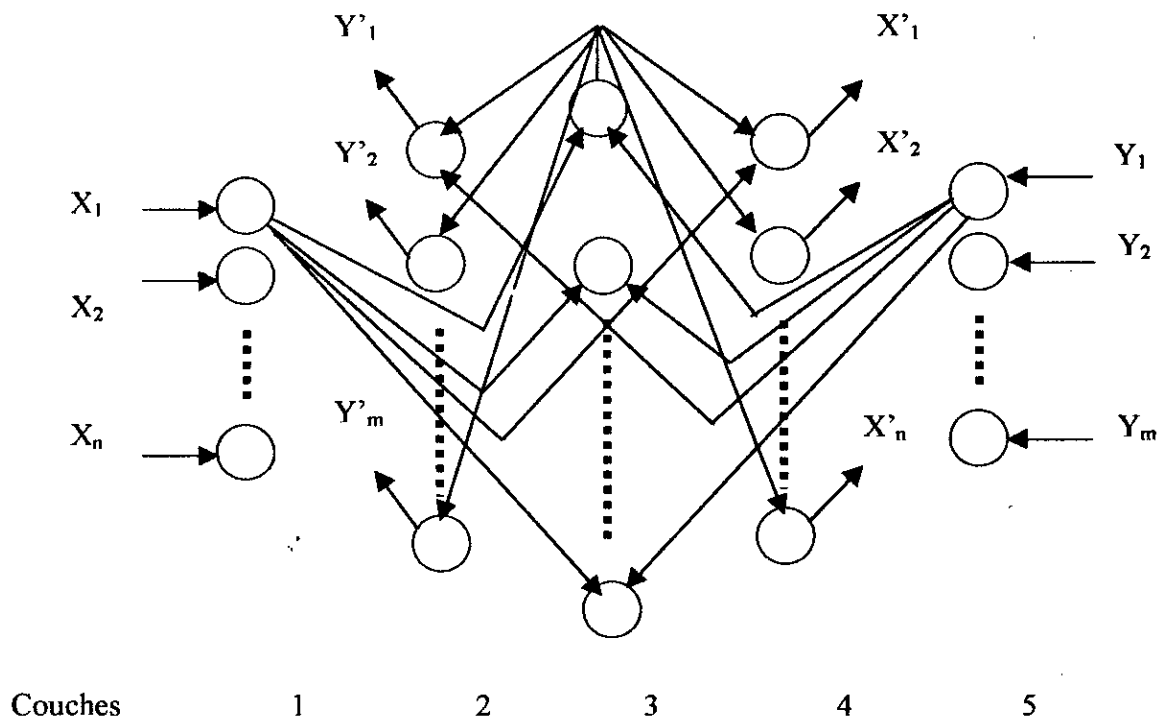


Figure (3.4)

Pour un vecteur des couples $(X_1, Y_1) \dots (X_L, Y_L)$ donnés, le CPN peut apprendre à associer pour chaque vecteur d'entrée X un vecteur de sortie Y . Si la fonction qui lie X à Y peut être représentée par ϕ , où $Y = \phi(X)$, le CPN apprend à approximer une sortie Y pour chaque valeur de X dans une classe spécifiée par des vecteurs de formation donnés. Si l'inverse de ϕ existe, où X est une fonction de Y , le CPN peut aussi apprendre l'approximation inverse $X = \phi^{-1}(Y)$.

Dans la pratique et pour plusieurs cas, la fonction inverse n'existe pas [5]. Dans ce cas le rôle du CPN peut se simplifier seulement à la formation de $Y = \phi(X)$.

Dans la figure (3.5) nous avons réorganiser l'architecture du CPN et nous avons considéré le simple cas de formation : le plan d'action directe « **the forward mapping** » [5].

Le réseau considéré maintenant est composé de trois couches. Un vecteur d'entrée est appliqué aux unités de la première couche. Chaque unité dans la couche cachée calcule son activation, puis une compétition entre eux permet de déterminer laquelle des unités a la plus grande sortie. Cette dernière unité est la seule qui va garder une sortie non nulle.

Par la combinaison des différents types des RNA déjà existant pour obtenir un nouveau type, le CPN utilise une différente procédure d'apprentissage pour chaque couche. Les algorithmes d'apprentissages du CPN tendent à un comportement très rapide par rapport aux autres réseaux.

Le CPN est une technique efficace pour beaucoup d'applications et l'est moins pour certaines [1].

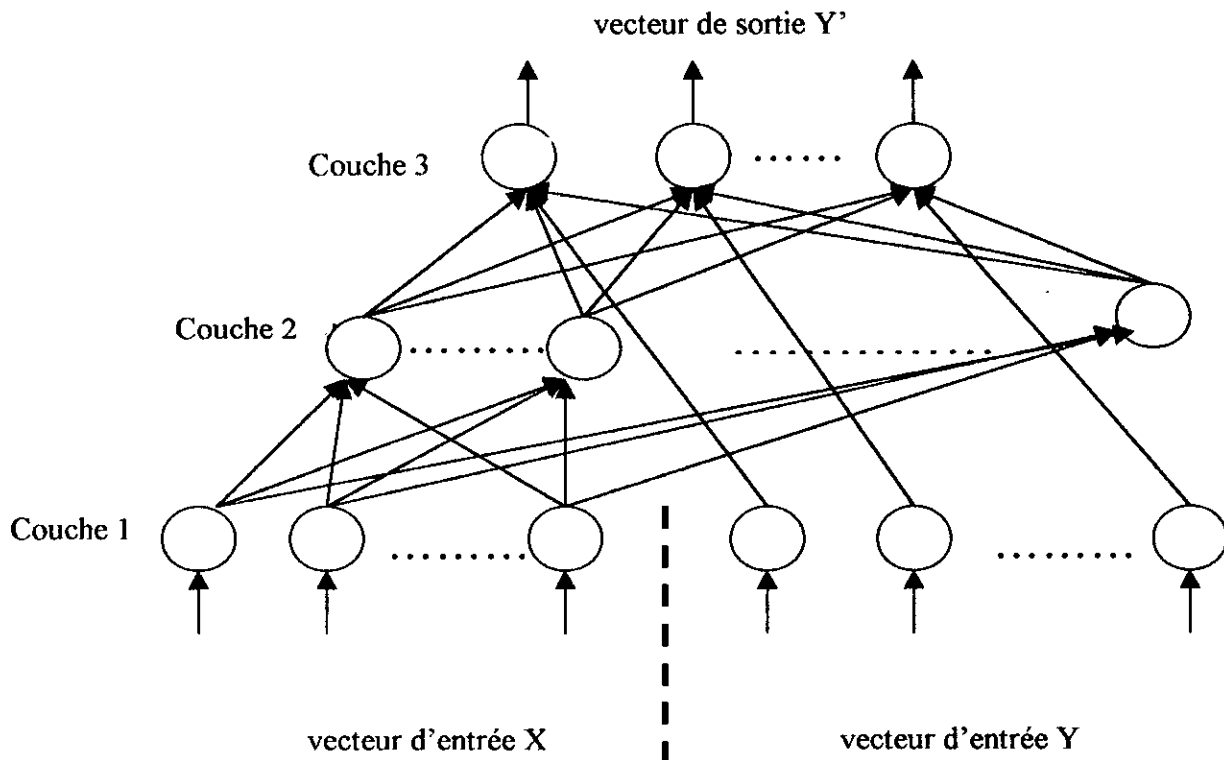


Figure (3.5)

III.5.1. Eléments constitutifs du CPN :

Les éléments présentés dans cette section jouent un rôle important dans l'étude et la compréhension du CPN. On distingue quatre principaux éléments :

- une couche d'entrée responsable seulement de la distribution des entrées pour les unités des autres couches.
- Un élément appelé "instar".
- Une couche composée de plusieurs instars appelée "couche compétitive".
- Une structure de sortie appelée "outstar".

La combinaison de tous ces éléments forme le CPN.

Couche d'entrée :

L'étude des réseaux de neurones ignore les éléments d'entrée ou les considère simples, ces éléments étant responsables seulement de la distribution des vecteurs d'entrée pour les éléments des autres couches. Les vecteurs d'entrée doivent être normalisés.

Nous commencerons par l'étude de la couche des éléments montrés par la figure (3.6), où une valeur d'entrée I_i est présentée pour chacune des n unités de la couche.

L'intensité totale du patron d'entrée est donnée par [5] : $I = \sum_i I_i$.

Pour chaque unité on va définir la fonction suivante :

$$\theta_i = I_i \left(\sum_i I_i \right)^{-1} \quad (3.1)$$

Le vecteur $(\theta_1, \dots, \theta_n)^T$ est le patron réfléchi [5].

Les sorties de cette couche sont proportionnelles au patron normalisé réfléchi.

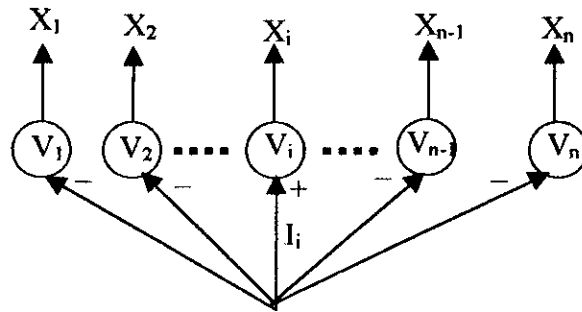


Figure (3.6)

On note que le vecteur $(\theta_1, \dots, \theta_n)$ est normalisé, où $\sum_i \theta_i = 1$. Le patron réfléchi est indépendant de l'intensité totale du patron d'entrée correspondant.

Les sorties des éléments de la figure (3.5) vérifient les équations différentielles suivantes :

$$\dot{x}_i = -Ax_i + (B - x_i)I_i - x_i \sum_{k \neq i} I_k \quad (3.2)$$

où $0 < x_i(0) < B$ et $A > 0, B > 0$.

Après l'application d'un patron d'entrée, les éléments tendent à l'état d'équilibre :

$$x_i^{eq} = \theta_i \frac{BI}{A + I} \quad (3.3)$$

Le patron de sortie est donc normalisé, et est toujours inférieur à B .

La couche étudiée dans cette section est utilisée pour construire les deux portions X et Y de la couche d'entrée du CPN présenté par la figure (3.4).

Instar :

La couche cachée du CPN contient un processus d'élément connu sous le nom d'instar.

Il faut noter que les vecteurs d'entrée et de poids sont normalisés à la valeur 1.

La sortie de l'instar est exprimée par l'équation différentielle suivante :

$$\dot{y} = -ay + b.net \quad (3.4)$$

où $a, b > 0$.

Il faut ensuite résoudre l'équation (3.4) pour obtenir la sortie en fonction du temps.

Considérant une sortie initiale nulle, si on applique un vecteur d'entrée durant l'intervalle $[0, t]$ on obtient :

$$y(t) = \frac{b}{a} net (1 - e^{-at}) \quad (3.5)$$

La valeur d'équilibre de $y(t)$ est :

$$y^{eq} = \frac{b}{a} net \quad (3.6)$$

Si le vecteur d'entrée est enlevé au temps t' , pour $t > t'$ la valeur de $y(t)$ est :

$$y(t) = y^{eq} e^{-a(t-t')} \quad (3.7)$$

On note que pour a et b données, la sortie d'équilibre augmente si *net-input* augmente.

La figure (3.7) montre un diagramme du poids w d'un instar, et un vecteur d'entre I . La valeur du *net-input* détermine comment les vecteurs d'entrée et poids sont ils, par la mesure de l'angle de déphasage entre w, I . La plus grande sortie d'équilibre est obtenue si $\theta = 0$: w et I sont alignés. Si on veut que l'instar apprend au maximum une entrée particulière, il faut un vecteur poids identique au vecteur d'entrée désire.

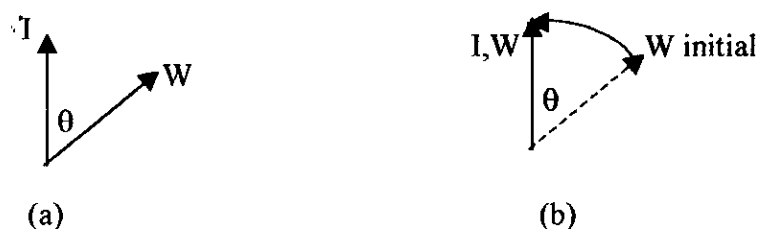


Figure (3.7)

L'instar peut apprendre à répondre à un vecteur d'entrée, si le vecteur poids varie en vérifiant l'équation suivante :

$$\dot{w} = -cw + dIy \quad (3.8)$$

ou y est la sortie de l'instar et $c, d > 0$.

Le deuxième terme de droite de l'équation (3.6) est un produit de l'entrée et la sortie des neurones. Ainsi, si l'entrée et la sortie sont grandes, les poids pour la connexion d'entrée sont renforcés, comme décrit par la théorie de **Hebb** [5].

L'équation (3.8) est difficile à résoudre parce que y est une fonction de temps très complexe. On peut simplifier le problème par considération que y atteint sa valeur d'équilibre rapidement devant la variation de w [5], donc $y = y^{eq} = \frac{b}{a} net$.

L'équation (3.8) devient :

$$\dot{w} = -cw + dI(w.I) \quad (3.9)$$

où le coefficient $\frac{b}{a}$ est compris dans la constante d .

L'absence d'un vecteur d'entrée I , diminue le vecteur poids vers zéro ; $\dot{w} = -cw$. Cet effet est connu sous le nom de "forget-fulness" du neurone [5]. Pour éviter ce phénomène, on peut modifier l'équation (3.8) en vérifiant que chaque changement du vecteur poids dépend du vecteur entrée qu'il doit mener à l'apprentissage.

La loi d'apprentissage utilisée s'écrit :

$$\dot{w} = (-cw + dI)U(net) \quad (3.10)$$

$$\text{tel que } U(net) = \begin{cases} 1 & net > 0 \\ 0 & net = 0 \end{cases}$$

où on peut intégrer l'équation (3.10) directement pour $U(net)=1$.

Notons que pour $c=d$ w tend vers une valeur exacte avec I [5]. On peut aussi réécrire l'équation (3.10) sous une forme plus souhaitable pour les simulation numériques.

$$\Delta w = \alpha(I - w) \quad (3.11)$$

Dans l'équation (3.11) l'approximation : $\frac{dw}{dt} = \frac{\Delta w}{\Delta t}$ a été utilisée, ainsi que $\alpha = c\Delta t$ a été considéré.

Une solution approximée de l'équation (3.10) devient :

$$w(t+1) = w(t) + \alpha(I - w(t)) \quad (3.12)$$

pour $\alpha > 1$.

La solution itérative de l'équation (3.12) est illustrée par la figure (3.8).

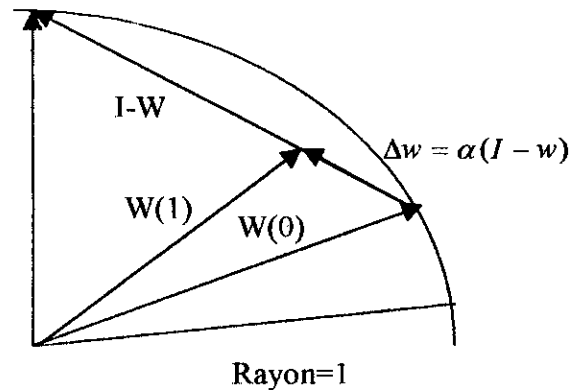


Figure (3.8)

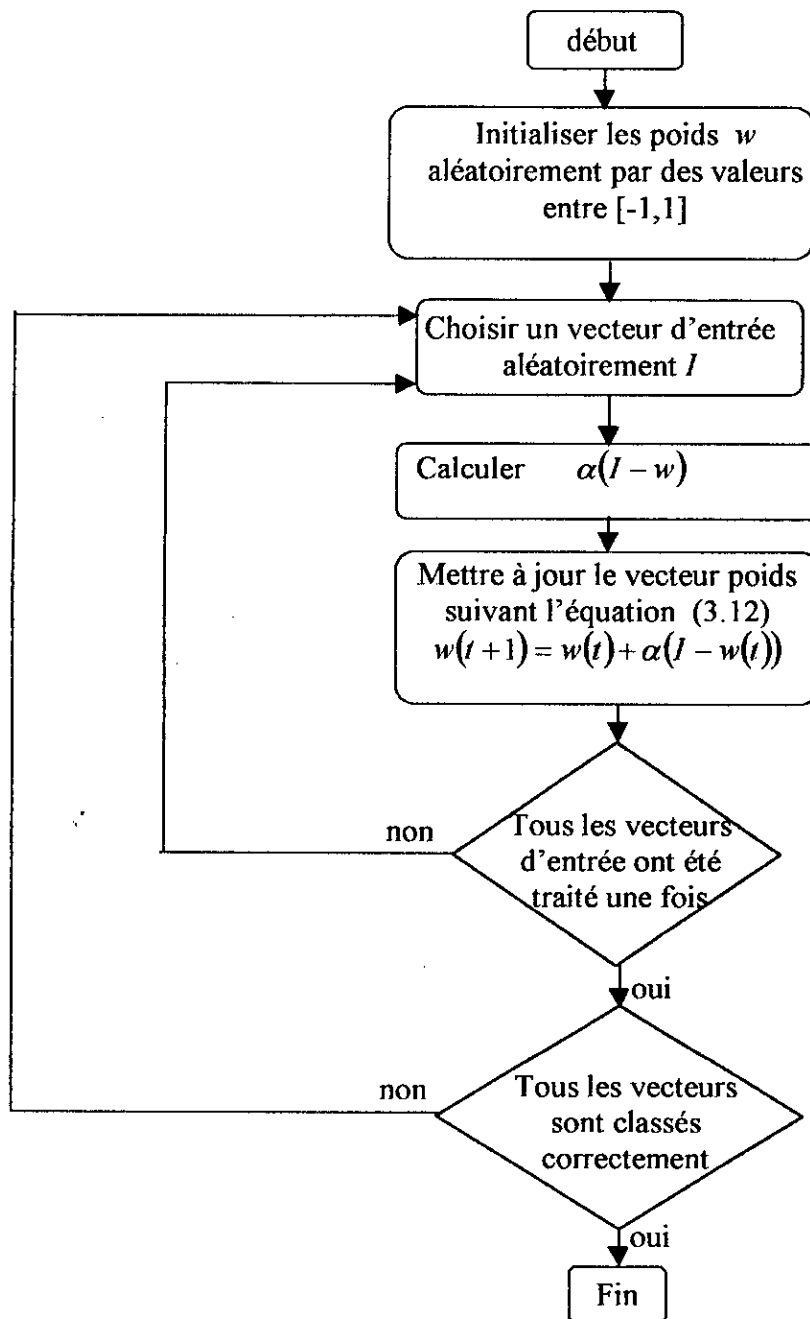
L'apprentissage d'un seul vecteur d'entrée par un seul instar n'est pas un cas intéressant. Considérant la situation où nous avons plusieurs vecteurs d'entrée tous relativement proches et rassemblés dans ce qu'on appelle "un groupe".

L'apprentissage se fait de la façon itérative suivante :

- 1- Sélectionner un vecteur d'entre I_i , aléatoirement, suivant la probabilité de distribution du groupe.
- 2- Calculer, et mettre à jour le vecteur poids suivant l'équation (3.12).
- 3- Répéter les étapes 1 et 2 pour un nombre de fois égal au nombre de vecteurs d'entrée du groupe.
- 4- Répéter l'étape 3 plusieurs fois.

On peut utiliser l'erreur moyenne comme critère d'arrêt. Dans la pratique ce critère est rarement utilisé. On utilise d'autres critères pour arrêter l'apprentissage.

Organigramme d'apprentissage de l'instar :



Les instars peuvent se regrouper pour former ce qu'on appelle un réseau compétitif [5]. La couche cachée du CPN est ce réseau compétitif.

Réseau compétitif :

Dans la section précédente, nous avons vu comment un instar individuel peut apprendre un groupe de vecteurs d'entrée regroupés dans une région de l'espace. Supposons que nous avons plusieurs instars regroupés dans une couche, comme illustré par la figure (3.9).

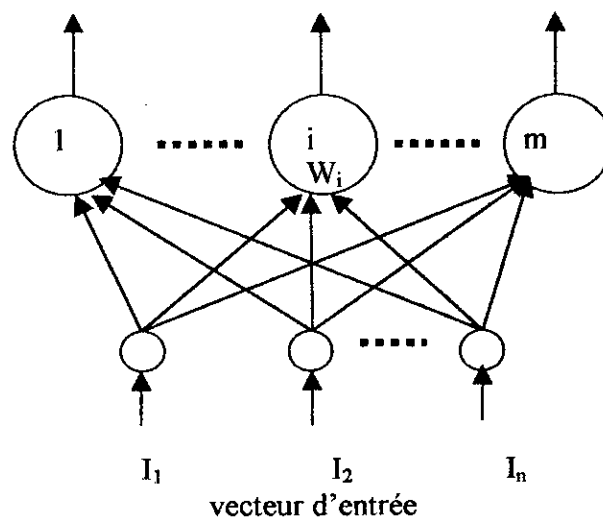


Figure (3.9)

On peut dire que cette couche d'instars classe les entrées présentées, car avec une réponse d'instar pour une entrée donnée il identifie où le vecteur d'entrée se trouve.

Pour que chaque instar se comporte correctement suivant le vecteur d'entrée présenté, il faut une vérification entre tous les neurones de la couche cachée. Cette vérification est simple, seul le neurone gagnant de la compétition a une sortie non nulle. Puisqu'il n'y a pas de juge externe pour choisir le neurone gagnant, les neurones doivent décider entre eux celui qui a la plus grande sortie. Ce processus de décision nécessite la communication entre tous les neurones de la couche cachée, ce qui complique l'étude.

Les activations des unités sont exprimées par des équations différentielles.

Ces équations peuvent prendre plusieurs formes. L'une des plus simples est :

$$\dot{x}_i = -Ax_i + (B - x_i)[f(x_i) + net_i] - x_i \left[\sum_{k \neq i} f(x_k) + \sum_{k \neq i} net_k \right] \quad (3.13)$$

où A et B sont des constantes positives, et $f(x_j)$ est une fonction d'activation. Si on compare l'équation (3.2) à l'équation (3.13), on peut convertir l'équation (3.2) à (3.13) en remplaçant chaque I_j dans l'équation (3.2) par $f(x_j + net_j)$ pour toutes les valeurs de j .

La relation entre les constantes A et B et la forme de la fonction $f(x_j)$ déterminent comment la solution de l'équation varie avec le temps [5].

L'étude de l'équation (3.13) devient simple si on la remplace par une paire d'équations : une qui décrit les variables réfléchies, $X_i = \frac{x_i}{\sum_k x_k}$, et l'autre décrit l'intensité

totale du patron, $x = \sum_k x_k$.

Réarrangerons d'abord l'équation (3.13) comme suit :

$$\dot{x}_i = -Ax_i + B[f(x_i) + net_i] - x_i \left[\sum_k f(x_k) + \sum_k net_k \right]$$

La somme par rapport à i donne :

$$\dot{x} = \sum_i x_i = -Ax + (B - x) \left[\sum_k f(x_k) + \sum_k net_k \right] \quad (3.14)$$

L'utilisation de la définition $g(w) = w^{-1}f(w)$ donne:

$$x\dot{X}_i = BxX_i \sum_k X_k [g(xX_i) - g(xX_k)] + B(1 - X_i)net_i - BxX_i \sum_{k \neq i} net_k \quad (3.15)$$

Commencerons par le cas simple où $f(w)$ est une fonction linéaire ; $f(w) = w$.

Puisque $g(w) = w^{-1}f(w) \Rightarrow g(w) = 1$, et le premier terme de l'équation (3.15) va s'annuler.

Les variables réfléchies se stabilisent à la valeur : $X_i = \frac{net_i}{\sum_k net_k}$. D'autre part on obtient :

$$x_i = x^{eq} \frac{net_i}{\sum_k net_k}$$

Maintenant observons ce qui arrive après l'élimination du patron d'entrée.

Si le patron d'entrée est supprimé, nous avons : $net_i = 0$ pour tout i , donc $\dot{X}_i = 0$, ce qui implique que les variables réfléchies deviennent constantes dans le temps. Les activations des unités dépendent maintenant seulement de x , puisque $x_i = xX_i$.

L'équation (3.14) se réduit à :

$$\dot{x} = (B - A - x)x$$

Si $B \leq A \Rightarrow \dot{x} = 0$ et $x \rightarrow B - A$, et les activations se stockent en permanence dans les unités.

Etudions à présent le cas d'une fonction quadratique, $f(w) = w^2$. Dans ce cas, $g(w) = w$. On note que le premier terme de droite de l'équation (3.15) contient le facteur $[g(xX_i) - g(xX_k)]$. Pour la fonction quadratique, ce facteur se réduit à : $x[X_i - X_k]$.

Si $X_i > X_k$ pour toutes les valeurs de $i \neq k$, le premier terme de l'équation (3.14) est un terme exciteur.

Si $X_i < X_k$ pour tout $i \neq k$ le premier terme de l'équation (3.14) est un terme inhibiteur [5].

Ainsi, ce réseau tend à augmenter l'activation de l'unité avec la plus grande valeur de X_i .

Après que le vecteur d'entrée est enlevé, X_i devient positive seulement pour l'unité de la plus grande valeur de X_i .

Considérons maintenant une fonction qu'on peut utiliser pour former un réseau compétitif dont le principe est : le neurone vainqueur pour tout "winner takes all", c'est à dire le neurone vainqueur a comme pondération 1 et tout les neurones restant ont comme pondération 0.

La fonction quadratique (pour la généralisation $f(w) = w^n$, $n > 1$) peut être utilisée dans ce qu'on appelle "off-surround" [5] pour stocker seulement l'entrée gagnante.

Supposons que la sortie d'unité doit être limitée dans le temps, pour éviter l'augmentation de l'allure du fonction de sortie. Cette hypothèse est vérifiée par une fonction d'activation sigmoïde. L'analyse mathématique des équations (3.14) et (3.15) avec une telle fonction d'activation est un peu plus compliquée par rapport aux deux premiers cas[5].

A partir des figures (3.4) et (3.5), on remarque qu'il n'y a pas d'interconnexions entre les unités de la couche cachée. Dans une simulation numérique nous n'avons pas besoin de ces interconnexions pour déterminer l'unité gagnante. Un programme peut jouer le rôle de juge externe. L'unité gagnante va prendre une sortie égale à 1, les autres unités ayant toutes des sorties nulles. Donc le problème de normalisation des entrées et la détermination de l'unité gagnante sont résolus par des programmes numériques (ces deux facteurs n'ont aucune influence sur l'architecture du réseau).

Outstar :

L'étape finale de notre description des éléments constitutifs du CPN est la structure de Grossberg de l'outstar. Comme la figure (3.10.a) le montre, un outstar est composé de toutes les unités de la couche de sortie et une unité de la couche cachée.

Considérons l'outstar de la figure (3.11). Dans un premier temps, le stimulant conditionné (*CS*) est incapable de produire une réponse pour les unités qui lui sont connectées [5]. Un stimulant non conditionné (*UCS*) peut poser une réponse non conditionnée (*UCR*) [5]. Si le (*CS*) est présent pendant que le (*UCS*) cause le (*UCR*), alors l'excitation de connexion de l'unité (*CS*) vers l'unité (*UCR*) va aussi augmenter. Plus tard, le (*CS*) sera capable de causer un (*CR*) (le même que pour le (*UCR*)), même si le (*UCS*) est absent. Pendant la période d'apprentissage, l'unité qui gagne la compétition dans la couche cachée va produire un seul (*CS*) pour les unités de sortie. Le (*UCS*) est assuré par la portion y de la couche d'entrée. Parce que nous voulons apprendre au réseau le vecteur y , le (*UCR*) doit le même que y .

Après que l'apprentissage est terminé, l'application du (*CS*) va produire la valeur (*CR*) (le vecteur y'), pour l'appliquer sur les unités de sortie. Les valeurs de (*UCS*) vont être toutes des zéros.

Dans le CPN, la couche cachée participe dans les deux architectures : outstar et instar du réseau. Le rôle des instars est de reconnaître un patron d'entrée à travers la compétition. La sortie de ce dernier neurone devient le (*CS*) d'un outstar. L'outstar identifie le patron d'entrée. L'instar et l'outstar se complètent de la façon suivante:

« L'instar reconnaît un patron d'entrée et le classe, l'outstar identifie ou nomme les classes sélectionnées ».

Cela a poussé un chercheur à dire que l'instar est muet (dumb), et l'outstar est aveugle (blind) [5].

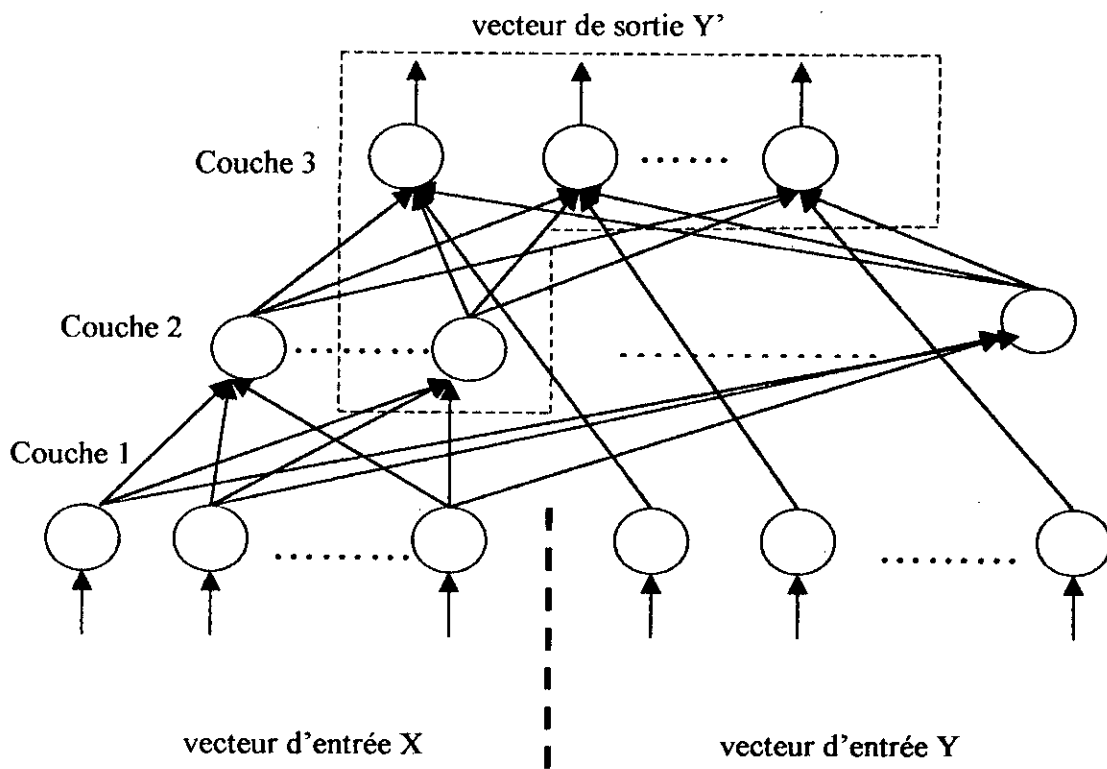
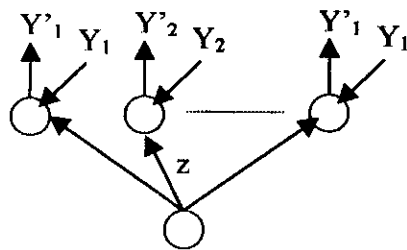


Figure (3.10. (a))



Figure(3.10.(b))

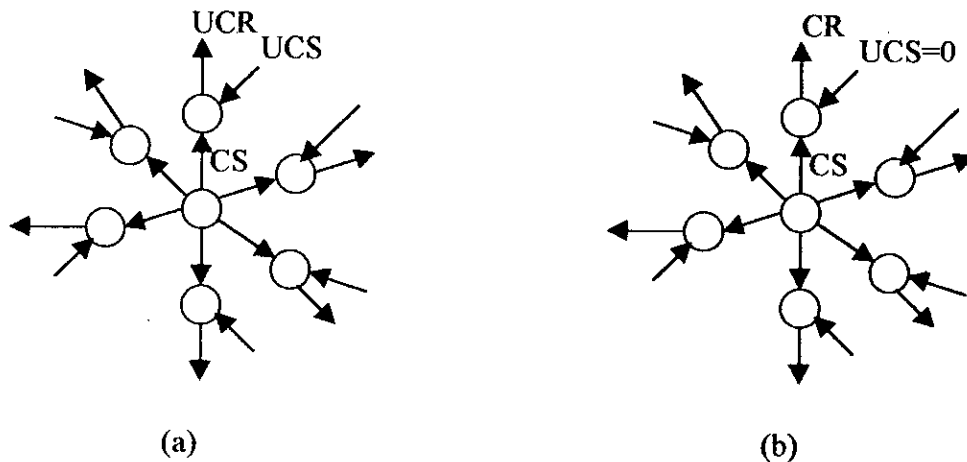


Figure (3.11)

Les équations qui dirigent les processus d'apprentissage de l'instar et de l'outstar sont similaires dans la forme. Pendant l'apprentissage, les valeurs de sorties de l'outstar sont exprimées par :

$$\dot{y}_i = -ay_i + by_i + c.net_i$$

similaire à l'équation (3.4) pour l'instar, sauf pour le terme ajouté dû à l'entrée d'apprentissage y_i .

Les paramètres a , b , c sont tous supposés positifs. La valeur du net_i est calculée à partir de la somme des produits entre les poids et les entrées des unités connectées à l'unité i . Pour l'outstar toutes les entrées sont nulles, sauf celle connectée à l'unité gagnante de la couche cachée. Donc, il y a un vecteur poids associé à chaque outstar.

L'activation *net-input* se réduit à un seul terme $w_{ij}z$, où z est la sortie de l'unité connectée.

Dans le cas du CPN $z = 1$.

On peut écrire :

$$\dot{y}_i = -ay_i + by_i + cw_{ij} \quad (3.16)$$

où j est le neurone de sortie connecté à celui qui a gagné la compétition de la couche cachée. En général, les paramètres a , b et c de l'équation (3.16) sont fonctions du temps. Dans notre cas, nous les considérons constants pour simplifier l'étude.

Après que l'apprentissage est terminée, c'est à dire l'équilibre des poids, la sortie de l'outstar s'exprime par l'équation différentielle suivante :

$$\dot{y}_i = -ay_i + cw_i^{eq} \quad (3.17)$$

où w_i^{eq} sont les valeurs d'équilibre des poids trouvés pendant l'apprentissage. Pour l'outstar la variation des poids est presque identique à celle de l'instar de l'équation (3.10).

L'équation qui décrit la variation des poids de l'outstar est :

$$\dot{w}_i = (-dw_i + ey_{i,z})U(z) \quad (3.18)$$

On note que le deuxième terme de l'équation (3.18) contient l'entrée d'apprentissage y_i , et non la sortie y'_i . La fonction $U(z)$ assure le non-apprentissage s'il n'y a pas d'entrée d'apprentissage ($z=0$).

Puisque z et $U(z)$ sont 1 pour l'unité gagnante de la couche cachée, l'équation (3.18) s'écrit :

$$\dot{w}_i = -dw_i + ey_i \quad (3.19)$$

l'équation (3.19) représente les connexions vers le neurone caché gagnant. Les connexions vers les autres unités de la couche cachée ne participent pas dans l'apprentissage des sorties. Si les vecteurs de sortie sont identiques pour tous les vecteurs, de la même classe, les valeurs d'équilibre des poids s'écrivent :

$$w_i^{eq} = \frac{e}{d} y_i$$

Si les vecteurs de sortie de la même classe sont éloignés, les valeurs d'équilibre des poids

$$\text{sont : } w_i^{eq} = \frac{e}{d} \langle y_i \rangle$$

La sortie d'équilibre devient :

$$y_i^{eq} = \frac{c}{a} \frac{e}{d} \langle y_i \rangle.$$

Parce qu'on cherche à obtenir des sorties identiques aux vecteurs de sortie, nous avons donc : $y_i^{eq} = \langle y_i \rangle$.

Posant $a=c$ et $d=e=1$ [5].

On obtient alors :

$$y_i^{eq} = \langle y_i \rangle = w_i^{eq} \quad (3.20)$$

Pour les simulations numériques, la forme souhaitable de l'équation (3.19) est :

$$w_i(t+1) = w_i(t) + \beta(y_i - w_i(t)) \quad (3.21)$$

III.5.2. Procédure des données du C.P.N :

Les différents éléments étant présentés, nous traitons à présent le CPN complet.

Plan d'action directe (forward mapping) :

Supposons que tout apprentissage a été fait et que nous sommes arrivés à l'étape de production. Nous avons un vecteur d'entrée, I , et nous voulons obtenir le vecteur de sortie, y , correspondant. Le processus est présenté dans la figure (3.12).

La procédure se traduit par les étapes suivantes :

- 1- Normaliser le vecteur d'entrée : $x_i = I_i / \sqrt{\left(\sum_n I_n^2\right)}$.
- 2- Appliquer le vecteur d'entrée normalisé à la portion X de la couche d'entrée. Appliquer un vecteur nul pour la portion Y de la couche d'entrée.
- 3- Puisque le vecteur d'entrée est déjà normalisé, la couche d'entrée est donc, responsable seulement de la distribution des données pour les unités des autres couches.
- 4- La deuxième couche est une couche compétitive. L'unité pour laquelle le vecteur poids est le plus proche du vecteur d'entrée va gagner la compétition et prend une sortie de I . Toutes les autres unités prennent des sorties nulles. La sortie de chaque unité peut être calculée par :

$$z_i = \begin{cases} 1 & \text{pour } \forall j \neq i \\ 0 & \text{ailleurs} \end{cases} \quad (3.22)$$

- 5- L'unique vainqueur de la couche compétitive excite un outstar.

Chaque unité dans l'outstar atteint une valeur d'équilibre rapidement, égale au poids de l'unité de sortie vers le neurone gagnant de la compétition (l'équation (3.20)). Si le i^{eme} neurone caché gagne la compétition, le vecteur de sortie devient : $y' = (w_{1i} w_{2i} \dots w_{mi})$, où m représente le nombre des neurones de sortie.

Nous avons :

$$y'_k = \sum_j w_{kj} z_j \tag{3.23}$$

Parce que $z_j = 0$ sauf pour $j=i$, donc $y'_k = w_{ki} z_i = w_{ki}$, ce qui est obtenu comme résultat dans la section précédente.

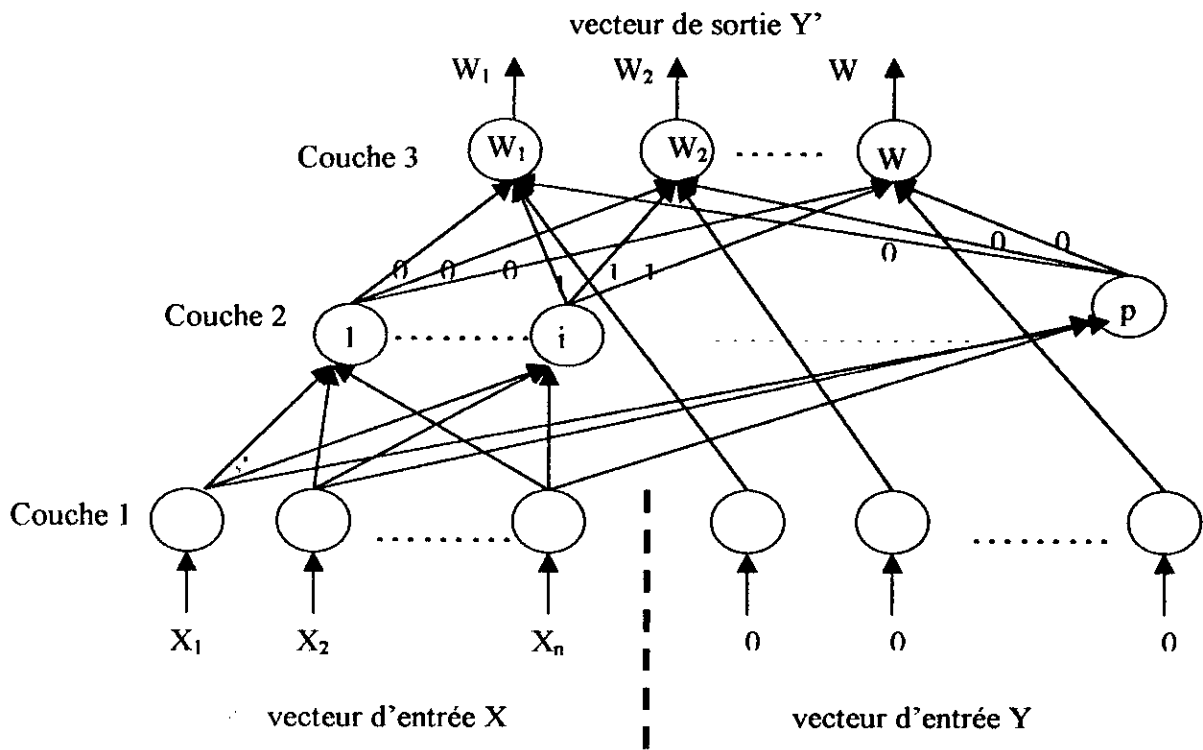
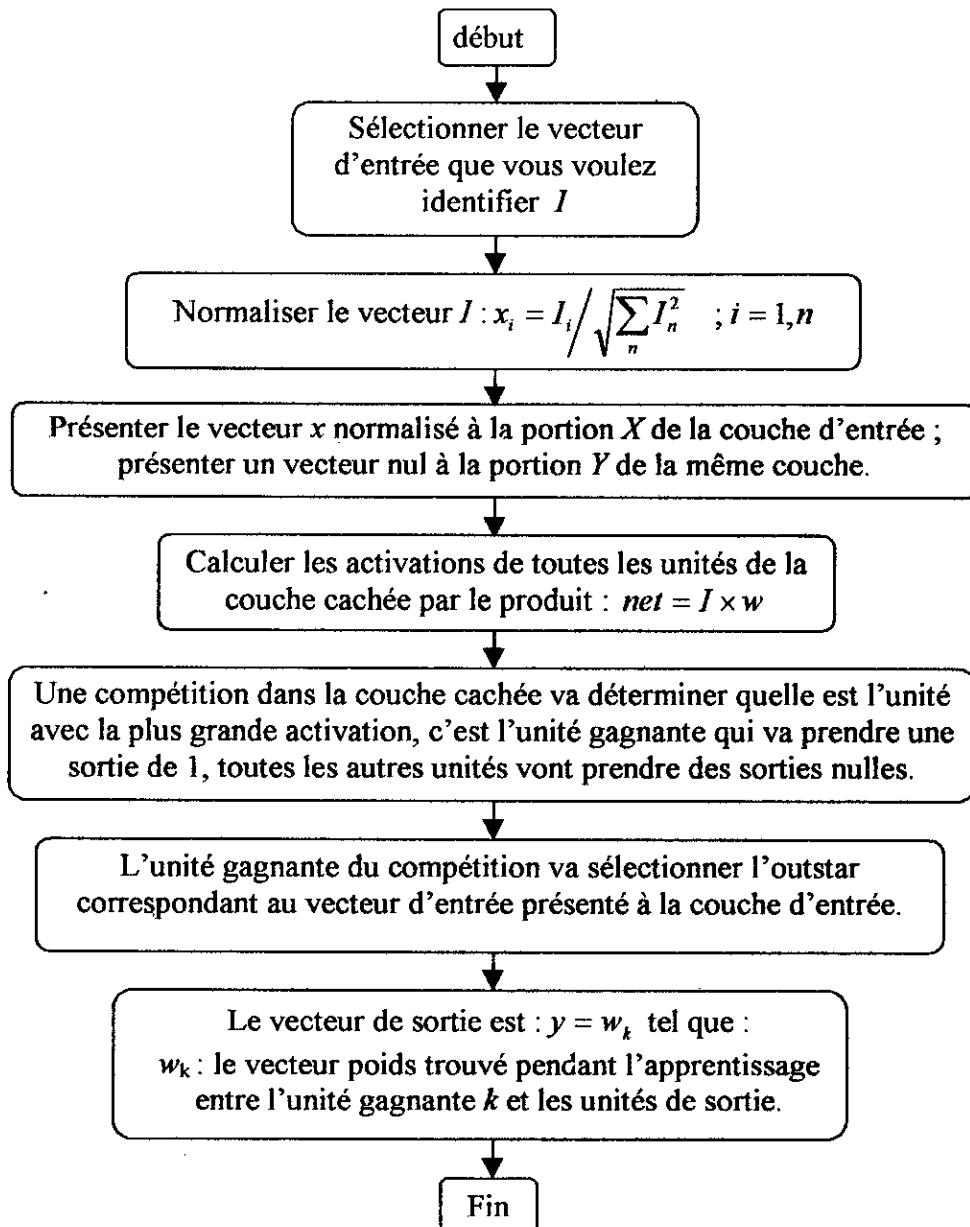


Figure (3.12)

Organigramme du plan d'action directe du CPN :

III.5.3. Apprentissage du CPN :

Ici encore, nous supposons qu'on fait une simulation numérique du CPN. On peut utiliser l'avantage des vecteurs d'entrée prénormalisés et un juge externe pour déterminer les gagnants de la couche cachée. On va aussi supposer qu'un ensemble de vecteurs d'apprentissage est suffisamment défini.

Parce qu'il y a deux différents algorithmes d'apprentissage utilisés pour le CPN, on va voir chacun des deux indépendamment de l'autre. On fait l'apprentissage complet de la couche compétitive avant de passer à l'apprentissage de la couche de sortie.

L'apprentissage des unités de la couche compétitive se fait suivant l'algorithme d'apprentissage de l'instar décrit dans la section (3.2). Parce qu'il y a pratiquement plusieurs instars dans la couche compétitive, la procédure d'apprentissage itérative de l'instar doit se modifier légèrement. On suppose que les vecteurs d'entrée se rassemblent dans différentes classes. Pour avoir une correcte classification pour chaque classe des vecteurs d'entrée, on procède comme suit :

Apprentissage de la couche compétitive :

- 1- Sélectionner un vecteur d'entrée parmi tous les vecteurs d'entrée pour l'utiliser dans l'apprentissage. La sélection doit être aléatoire.
- 2- Normaliser le vecteur d'entrée et l'appliquer à la couche compétitive du CPN.
- 3- Déterminer quelle unité gagne la compétition par le calcul d'activation pour chaque unité et sélectionner l'unité avec la plus grande activation (l'unité pour laquelle le vecteur poids est plus proche du vecteur d'entrée).
- 4- Calculer $\alpha(x - w)$ seulement pour l'unité gagnante, et mettre à jour le vecteur poids de l'unité gagnante suivant l'équation (3.12) :

$$w(t+1) = w(t) + \alpha(I - w(t))$$

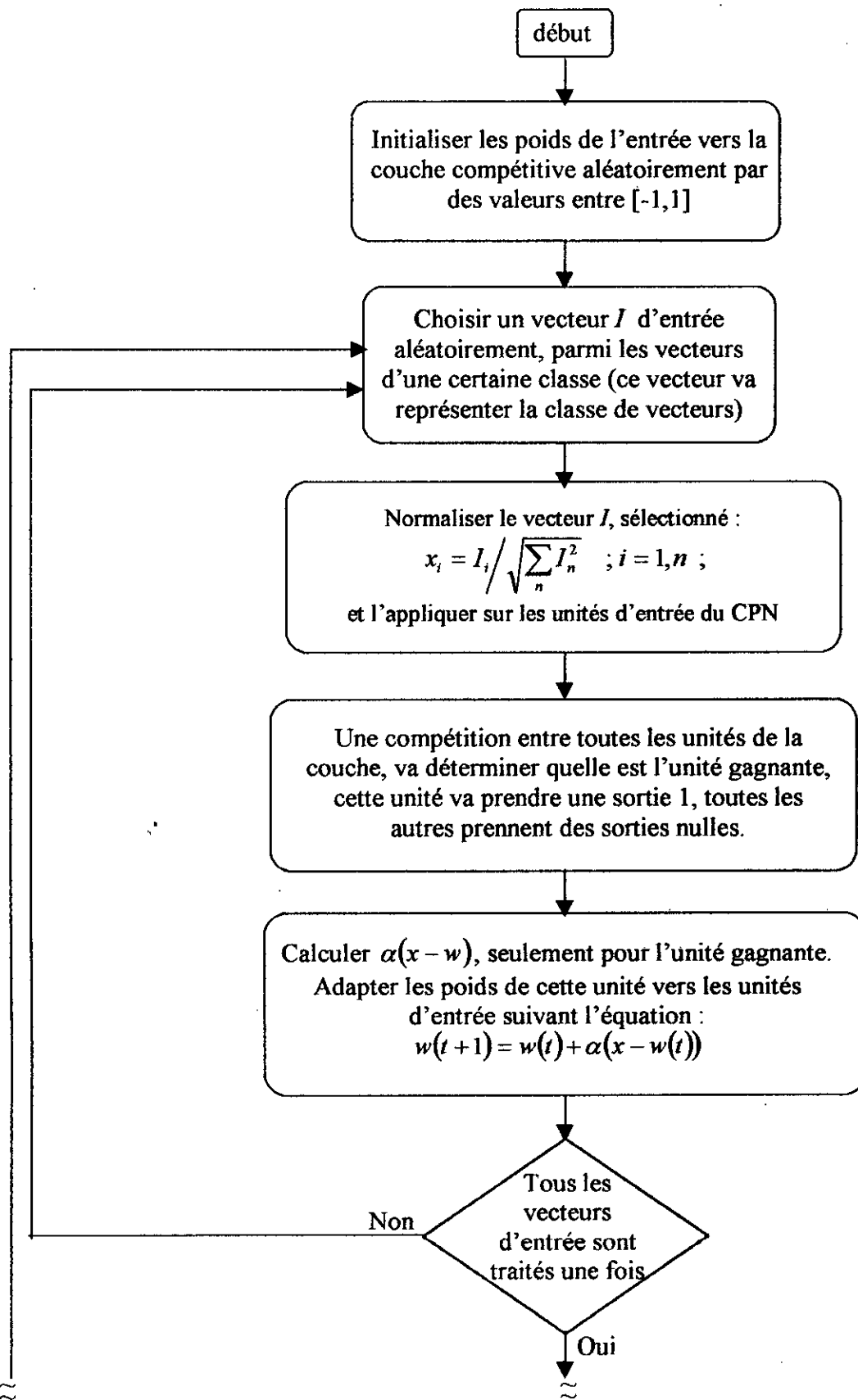
- 5- Répéter les étapes de 1 à 4 jusqu'à ce que tous les vecteurs d'entrée soient traités une fois.

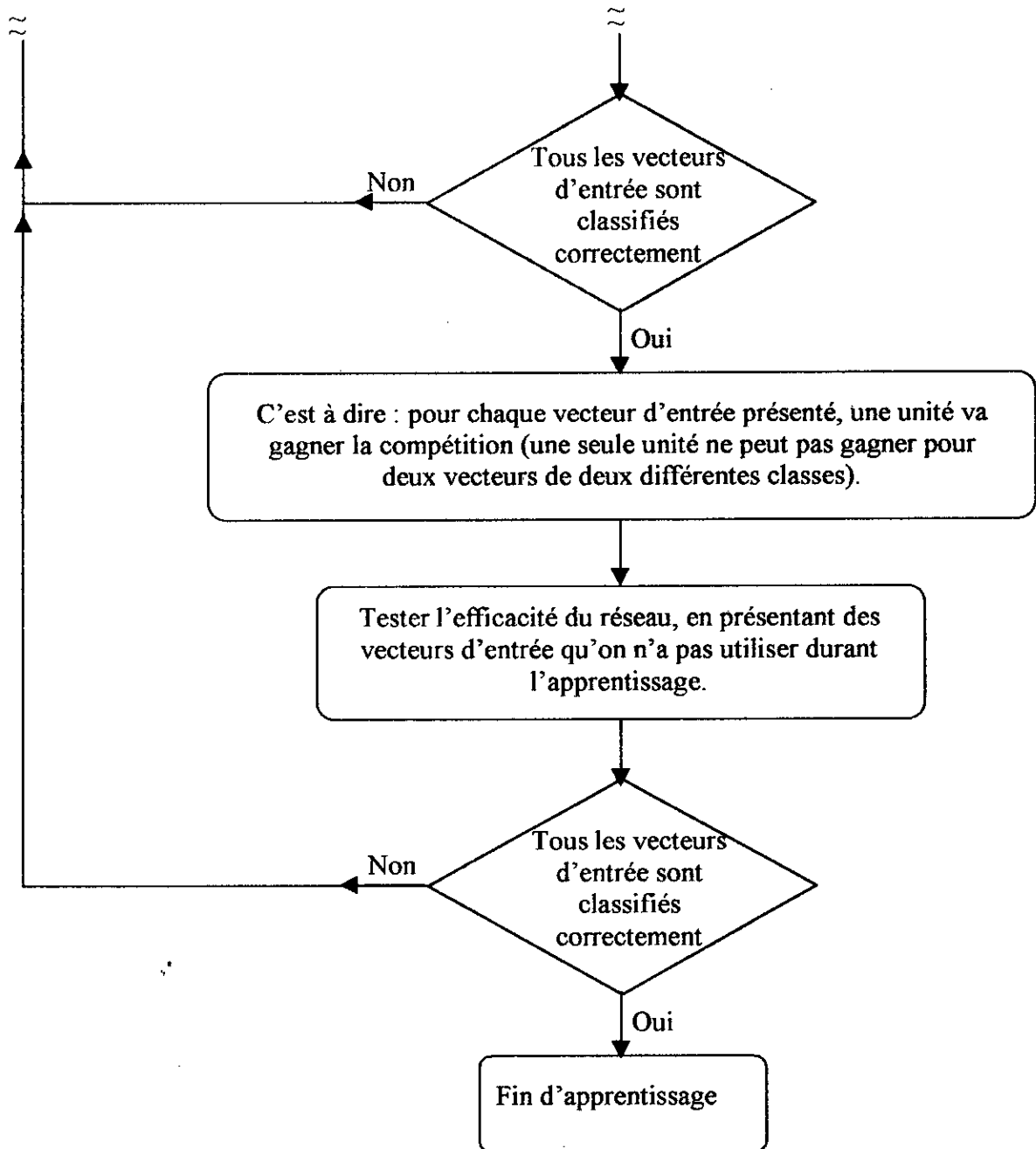
- 6- Répéter l'étape 5 jusqu'à ce que tous les vecteurs d'entrée soient classifiés correctement. Quand cette situation existe, une seule unité d'instar va gagner la compétition pour tous les vecteurs d'entrée de la classe.
- 7- Tester l'efficacité d'apprentissage en appliquant des vecteurs d'entrée des différentes classes qu'on n'a pas utilisé pendant la procédure d'apprentissage. Si quelques mauvaises classifications se présentent, un apprentissage supplémentaire passant par l'étape 6 devient nécessaire, jusqu'à ce que tous les vecteurs d'apprentissage se trouvent classés correctement.

L'un des points très important dans l'étude des réseaux de neurones est l'initialisation des poids. Pour tous les problèmes, sauf les simples, une initialisation aléatoire n'est pas suffisante. L'une des méthodes d'initialisation des poids utilisées est de mettre chaque vecteur poids égal à un représentant d'une classe d'entrée [1].

Après obtention de résultats satisfaisants dans l'apprentissage de la couche cachée, on passe à l'apprentissage de sortie.

Organigramme d'apprentissage de la couche compétitive :





Apprentissage de la couche de sortie :

Si nous avons plusieurs classes de vecteurs d'entrée, et tous les vecteurs de chaque classe produisent le même vecteur de sortie, il n'est pas nécessaire d'utiliser un algorithme d'apprentissage itératif. On doit seulement déterminer l'unité gagnante de la couche cachée pour une classe donnée, puis on met le vecteur poids entre l'unité cachée gagnante et toutes les unités de sortie égal au vecteur de sortie. Si la i^{eme} unité cachée gagne la compétition pour tous les vecteurs d'entrée de la classe où A est le vecteur de sortie, on met donc $w_{ki} = A_k$, où w_{ki} est le poids de la connexion entre la i^{eme} unité cachée et la k^{eme} unité de sortie. Si les vecteurs de sortie produits par les entrée de la même classe sont différents mais leur moyenne est connue, on met $w_{ki} = \langle A_k \rangle$, où $\langle A_k \rangle$ est la moyenne des vecteurs de sortie de la classe.

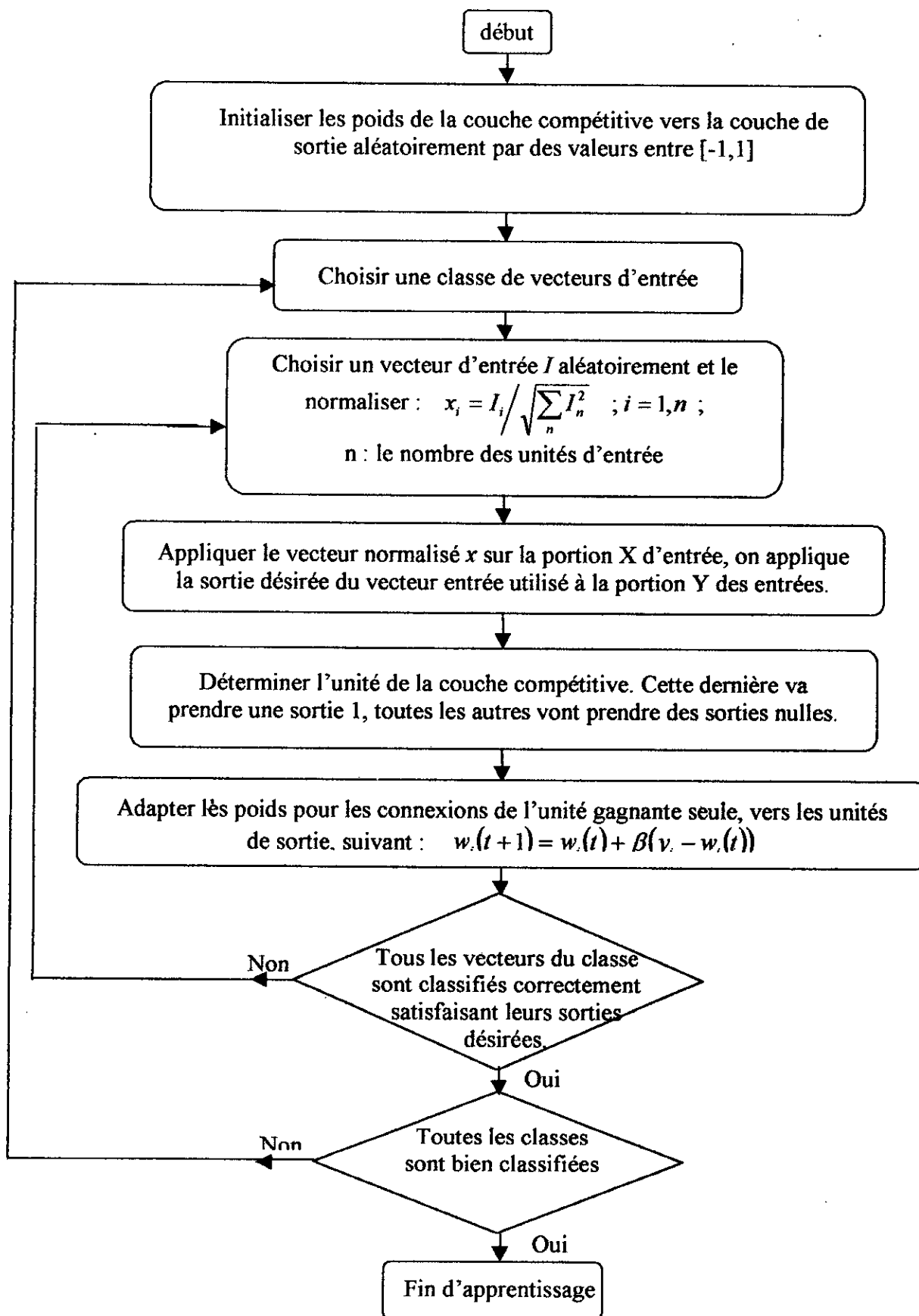
Si la moyenne des vecteurs de sortie des différentes classes est inconnue, l'apprentissage suit une procédure itérative basée sur l'équation (3.21). L'algorithme d'apprentissage de l'**outstar** est le suivant :

- 1- Appliquer un vecteur d'entrée normalisé, x_k , et son vecteur de sortie désiré correspondant, y_k , sur les portions, X et Y , de la couche d'entrée du CPN respectivement.
- 2- Déterminer l'unité gagnante de la couche compétitive.
- 3- Mettre à jour les poids des connexions de l'unité cachée gagnante vers toutes les unités de la couche de sortie suivant l'équation (3.21) :

$$w_i(t+1) = w_i(t) + \beta(y_i - w_i(t)).$$

- 4- Répéter les étapes de 1 à 3 jusqu'à ce que tous les vecteurs d'entrée de toutes les classes s'organisent satisfaisant les sorties.

Organigramme d'apprentissage des sorties :



III.5.4. CPN complet :

Nous avons traité précédemment le plan d'action directe. Mais le CPN complet présente également le plan d'action inverse comme l'illustre la figure (3.5).

Les deux portions de la couche d'entrée X et Y doivent être normalisés pour l'apprentissage du CPN complet. Comme pour le plan d'action directe du CPN (**forward-mapping CPN**), les deux vecteurs X et Y , sont appliqués aux unités de la couche d'entrée pendant le processus d'apprentissage. Après l'apprentissage, l'application d'un vecteur d'entrée (X, Φ) impose une sortie Y' , et l'application d'un vecteur d'entrée (Φ, Y) impose une sortie X' .

Parce que les deux portions de l'entrée X et Y sont connectées aux unités de la couche cachée, il y a deux vecteurs poids associés à chaque unité cachée. Un vecteur poids r correspond aux connexions vers l'entrée X , un autre vecteur du poids s correspond aux connexions vers l'entrée Y . Chaque unité dans la couche compétitive calcule son activation en utilisant :

$$net_i = r.x + s.y$$

Les sorties des unités de cette couche sont :

$$z_i = \begin{cases} 1 & net_i = \max \{net_k\} \\ 0 & \text{ailleurs} \end{cases}$$

pendant le processus d'apprentissage, les poids varient comme il suit :

$$\begin{cases} \dot{r}_i = \alpha_x (X - r_i) z_i \\ \dot{s}_i = \alpha_y (Y - s_i) z_i \end{cases}$$

Comme pour le plan d'action directe du CPN (forward-mapping CPN), seule l'unité gagnante qui va apprendre un vecteur d'entée donné.

Comme pour la couche d'entrée, la couche de sortie est partagée en deux différentes parties. Les connexions de la couche cachée vers les unités de la portion Y' de la couche

de sortie ont des poids w_i , et les connexions de la couche cachée vers la portion X' ont des poids V_i .

Les lois d'apprentissage de la couche de sortie sont :

$$\begin{cases} \dot{w}_{ij} = (-cw_{ij} + dy_i)z_j \\ \dot{v}_{ij} = (-cv_{ij} + fx_i)z_j \end{cases}$$

Comme pour le plan d'action directe du CPN, seuls les poids avec $z_j \neq 0$ vont apprendre.

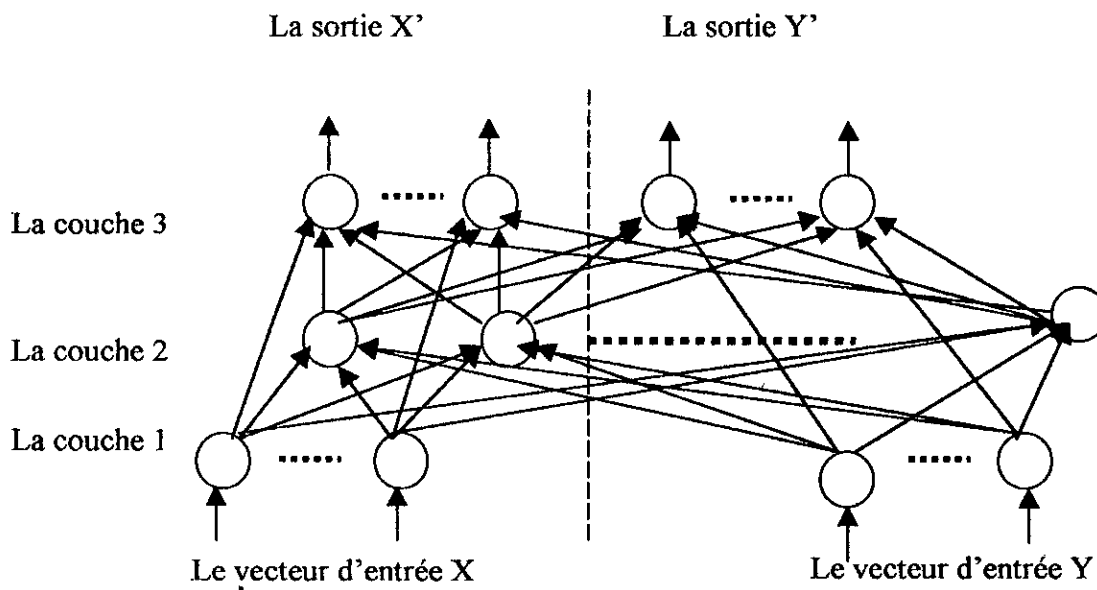


Figure (3.13)

III.6. Conclusion :

Dans ce chapitre nous avons présenté les définitions et les notions de base d'un réseau de neurones, telle que leur architecture, principe de fonctionnement et la notion d'apprentissage qui est un point très important dans l'étude des réseaux de neurones. Les caractéristiques intéressantes que ces réseaux présentent leurs ont permis de résoudre plusieurs problèmes, où des méthodes classiques ont montré leurs limites.

Dans un réseau de neurones, le traitement des données est parallèle et de manière distribuée, c'est à dire que toutes les unités du réseau travaillent simultanément, ce qui fait que le temps de calcul des opérations s'en trouve réduit.

Nous avons vu que le CPN est une combinaison de deux différents types des réseaux ; le modèle de KOHONEN utilisé pour sa couche cachée effectue une classification basée sur son fonctionnement compétitif ; le modèle de GROSSBERG est utilisé dans la couche de sortie. A partir du neurone gagnant de la couche de sortie, ce dernier impose la sortie du CPN.

Le CPN est un réseau de neurones adaptatif et peut être appliqué en temps réel. Les réseaux compétitifs, et en particulier le CPN, sont plus sélectifs en classification.

Chapitre 4 Identification des courts-circuits par le CPN

VI.1. Introduction :

L'étape finale de ce travail est de construire un réseau de neurone capable de résoudre notre problème. Les étapes précédentes nous ont permis de générer l'ensemble de données nécessaires pour la construction de notre réseau de neurones. Ce réseau aura pour but l'identification en temps réel de défauts qui peuvent apparaître dans un réseau électrique. Parmi les différents modèles existants, notre choix s'est porté sur le modèle du CPN. Ce modèle opère en mode non supervisé.

Dans ce qui suit, nous présenterons les différentes étapes de conception d'un tel réseau de neurones.

VI.2. Identification des défauts :

Pour identifier les différents types de courts-circuits on utilise le CPN. Les différents paramètres qui peuvent être utilisés pour décrire l'état d'un réseau électrique sont [4] :

- tensions aux jeux de barre
- courants de lignes
- déphasages entre les courants et les tensions
- paramètres de topologie du réseau

D'autres paramètres peuvent aussi être utilisés tels que les puissances active et réactive, l'admittance propre au nœud, etc.

Dans notre étude, et pour une raison de simplification, seules les tensions de jeux de barres sont utilisées.

Pour un réseau électrique triphasé quelconque, cinq types de défauts peuvent apparaître. Le réseau de neurones étudié ici peut identifier n'importe quel type parmi les cinq défauts possibles.

On peut résumer les différents types de courts-circuits dans le tableau (4.1).

Tableau 4.1 : classement de défauts.

Le code	Type du défaut	Symbole du défaut
0	Pas de défaut	NF
1	Court-circuit monophasé	SLGF
2	Court-circuit biphasé à la terre	DLGF
3	Court-circuit triphasé	3PHF
4	Court-circuit biphasé isolé	LLF

Les grandeurs utilisées comme entrées pour le réseau de neurones sont les trois tensions de phases d'un nœud quelconque (amplitudes) et les déphasages entre tensions de phases.

En combinant ces différentes entrées, deux architectures du CPN peuvent être utilisées :

Méthode 1 : Les amplitudes de trois tensions de phases seules ;

Méthode 2 : Les amplitudes de trois tensions de phases plus les déphasages entre ses tensions.

VI.2.1. Méthode 1 :

Conception du réseau de neurones :

Ayant fixé le choix du modèle, on optera pour le choix des valeurs efficaces des trois tensions de phases comme vecteurs d'entrée, représentant les différents états du réseau électrique lors des courts-circuits.

La sortie du réseau est formée de cinq nœuds correspondant aux différents types de défauts. Cette sortie peut se schématiser par un vecteur de cinq éléments, où chaque neurone

est lié aux neurones de la couche cachée par des connexions pondérées avec les vecteurs poids vers la couche cachée W .

La couche cachée de notre réseau est composée de 11 neurones, représentant tous les cas possibles de courts-circuits (un cas pour le court-circuit triphasé, trois pour le court-circuit monophasé, trois pour le court-circuit biphasé isolé et trois pour le court-circuit biphasé à la terre)

Donc, le CPN utilisé dans cette méthode est composé de trois couches dont l'architecture est montrée par la figure (4.1).

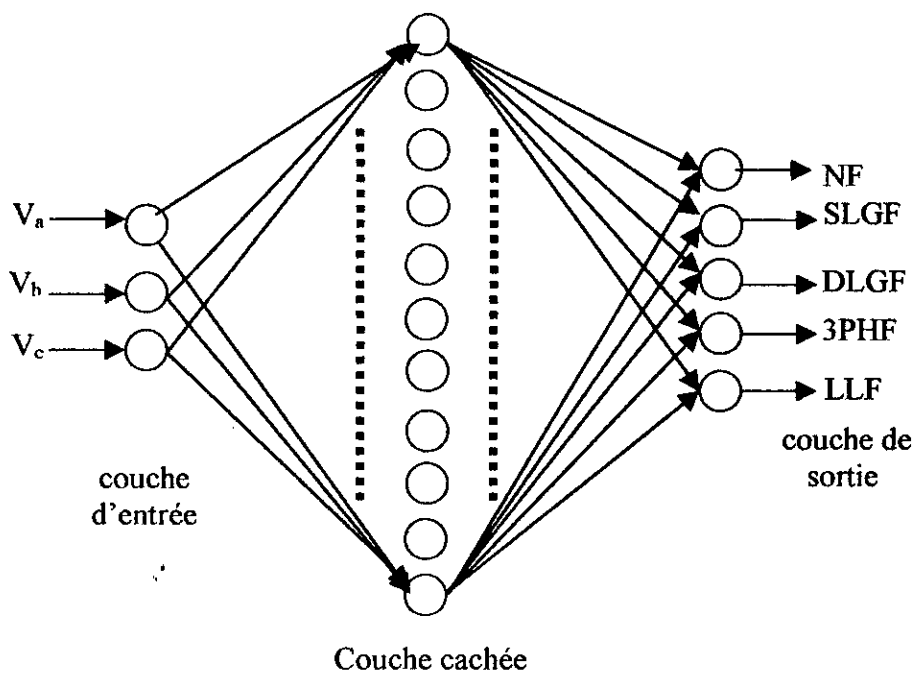


Figure (4.1)

- une couche d'entrée contient trois neurones représentant les trois tensions.
- une couche compétitive (cachée), contient tous les cas de défauts possibles, donc 11 neurones.
- une couche de sortie composée de cinq neurones représentant les cinq catégories de défaut.

Le principe du modèle utilisé est de minimiser l'écart entre les vecteurs poids, de la couche d'entrée vers la couche compétitive, et les vecteurs d'entrée dans un premier temps, puis il nomme la sortie correspondante à l'unité pour laquelle le vecteur poids est le plus proche au vecteur d'entrée appliqué. Cette minimisation est faite par l'adaptation des poids.

Adaptation des poids :

Notre réseau de neurones, comme indiqué au chapitre trois, est une combinaison de deux différents types de réseaux, d'où la présence de deux algorithmes d'adaptation de poids : un pour la couche compétitive et un autre pour la couche de sortie.

Suite à l'application de l'entrée, la couche de KOHONEN (la couche cachée) repère le neurone le plus excité (dont le vecteur poids est le plus proche de l'entrée). Le neurone obtenu adaptera ses poids vers tous les neurones de la couche d'entrée par la relation d'adaptation de poids :

$$w(t+1) = w(t) + \alpha(I - w(t))$$

de façon à se rapprocher graduellement de l'entrée. Cette opération sera effectuée à chaque itération jusqu'à l'obtention d'un réseau compétitif stable dont les poids resteront fixes.

L'adaptation des poids de la couche de KOHONEN étant terminée, le réseau compétitif est stable. On passe ensuite à la couche de sortie. Le neurone déclaré gagnant de la couche compétitive est le seul qui va émettre une sortie non nulle aux éléments de sortie. Si tous les vecteurs du même classe d'entrée arrivent à produire la même sortie, on n'aura pas besoin d'adapter les poids du neurone déclaré dans la couche compétitive vers les neurones de sortie de façon itérative, ce qui est le cas de notre étude : les vecteurs de sortie sont bien définis. Dans ce cas on pose les poids entre le neurone gagnant et les neurones de sortie les mêmes que les vecteurs de sortie correspondants. Comme si le rôle de la couche compétitive est de connaître le défaut, et le rôle de la couche de sortie est de nommer le défaut détecté.

Une fois l'adaptation de poids terminée pour la couche cachée ainsi que pour la couche de sortie, un neurone dans la couche cachée gagne la compétition pour tous les vecteurs d'une classe, et les neurones de sortie vont bien identifier toutes les classes possibles de l'entrée. La phase d'apprentissage est terminée. On doit valider ensuite les résultats obtenus.

Phase de validation :

Après la détermination des différentes classes composant le CPN, la phase de validation aura pour but de confirmer nos résultats.

Pour ce faire, on dispose du même réseau de neurones doté cette fois, d'une base de données différente. Les nouvelles données correspondent à des amplitudes de tension inutilisées pendant la phase précédente. Ce réseau subira les mêmes étapes que le réseau de neurones précédent.

Les nouvelles classes obtenues devront correspondre aux même groupe de neurones que dans le premier. On obtiendra, en définitif, les mêmes résultats.

Phase de test :

Cette phase aura pour objectif de tester les capacités de notre réseau stabilisé. Il lui sera présenté des entrées différentes, correspondantes à des entrées autres que celles utilisées dans les deux précédentes phases. Le réseau devra les classer correctement suivant les classes composant ses vecteurs d'entrée, ceci permettra par la même occasion d'évaluer sa capacité de généralisation.

Capacité de généralisation :

Après la construction, le réseau de neurones réalise une fonction continue des entrées vers les sortie. Dans une certaine mesure, il généralise, car il peut fournir une réponse à toute entrée et les réponses fournies sont arbitrairement proches si les exemples utilisés pour l'apprentissage le sont suffisants. Les réseaux de neurones sont de ce point de vue très intéressants, car certains de leurs comportements de généralisation sont intuitivement assez proches de la généralisation humaine [8].

D'un point de vue plus technique, la qualité de généralisation est liée à la qualité des frontières de classification établies par le réseau. Ces frontières ne doivent pas être trop proches des exemples appris ou très déséquilibrées par rapport aux différentes classes. Ainsi, un apprentissage trop poussé, c'est à dire en s'efforçant d'avoir une erreur très faible sur les exemples appris, nuit à la généralisation, car les frontières obtenues par le réseau suivent de très près les exemples appris.

La capacité de généralisation étant une qualité essentielle pour un système de classification, les réseaux de neurones, du point de vue qualitatif possédant de bonnes propriétés.

Exécution :

Présentation du réseau :

- Les entrées sont composées de trois éléments, les vecteurs d'entrées utilisés pendant l'apprentissage de notre réseau sont présentés au tableau 4.2.

Tableau 4.2 : les vecteurs d'apprentissage utilisés pour la première méthode

Code	Type du défaut	Entrées (pu)		
		V_a	V_b	V_c
1	NF	1	1	1
2	SLGF	0	1	1
3	SLGF	1	0	1
4	SLGF	1	1	0
5	DLGF	0	0	1
6	DLGF	0	1	0
7	DLGF	1	0	0
8	3PHF	0	0	0
9	LLF	0.5	1	0.5
10	LLF	1	0.5	0.5
11	LLF	0.5	0.5	1

- Après trois itérations de calcul, la couche compétitive de notre réseau a adapté ses poids. Donc, on a abouti à la stabilisation, c'est à dire : après l'application de tous les vecteurs d'entrée, tel neurone caché et seul va gagner la compétition pour le représentant de telle classe de vecteurs d'entrée. Les résultats trouvés pendant la phase d'apprentissage sont donnés au tableau 4.3.

Tableau 4.3 : les résultats d'apprentissage du réseau de neurones.

Le code	Type de défaut	Les entrées			Le neurone caché gagnant	Les sorties du réseau				
		V _a	V _b	V _c		Out1	Out2	Out3	Out4	Out5
1	NF	1	1	1	11	1	0	0	0	0
2	SLGF	0	1	1	10	0	1	0	0	0
3	SLGF	1	0	1	9	0	1	0	0	0
4	SLGF	1	1	0	8	0	1	0	0	0
5	DLGF	0	0	1	7	0	0	1	0	0
6	DLGF	0	1	0	6	0	0	1	0	0
7	DLGF	1	0	0	5	0	0	1	0	0
8	3PHF	0	0	0	4	0	0	0	1	0
9	LLF	0.5	1	0.5	3	0	0	1	0	0
10	LLF	1	0.5	0.5	2	0	0	1	0	0
11	LLF	0.5	0.5	1	1	0	0	1	0	0

Validation du réseau de neurones :

Pour valider les résultats trouvés pendant l'apprentissage, on a refait un autre apprentissage mais avec des entrées différentes de celles utilisées précédemment.

On a refait les mêmes étapes précédentes, et les résultats trouvés sont identiques à celles trouvés pendant l'apprentissage. Dans cette phase on a présenté au réseau de neurones 0.8 à la place de 1, 0.4 à la place de 0.5 et 0.2 à la place de 0 puis 0.9 à la place de 1, 0.45 à la place de 0.5 et 0.1 à la place de 0. Les résultats du test de validation pour la première base de données sont montrés au tableau 4.4. Pour la deuxième base de données, les résultats sont identiques aux tableau 4.4.

Tableau 4.4 : résultats du test de validation du réseau de neurones.

ode de défaut	Type de défaut	Entrées			Neurone caché gagnant	Sorties du réseau				
		V _a	V _b	V _c		Out1	Out2	Out3	Out4	Out5
1	NF	0.8	0.8	0.8	11	1	0	0	0	0
2	SLGF	0.2	0.8	0.8	10	0	1	0	0	0
3	SLGF	0.8	0.2	0.8	9	0	1	0	0	0
4	SLGF	0.8	0.8	0.2	8	0	1	0	0	0
5	DLGF	0.2	0.2	0.8	7	0	0	1	0	0
6	DLGF	0.2	0.8	0.2	6	0	0	1	0	0
7	DLGF	0.8	0.2	0.2	5	0	0	1	0	0
8	3PHF	0.2	0.2	0.2	4	0	0	0	1	0
9	LLF	0.4	0.8	0.4	3	0	0	1	0	0
10	LLF	0.8	0.4	0.4	2	0	0	1	0	0
11	LLF	0.4	0.4	0.8	1	0	0	1	0	0

Mise à l'épreuve du réseau de neurones :

Une fois le CPN validé, sa capacité de généralisation est testée en le soumettant à différentes entrées, les unes figurantes sur sa base de données, les autres correspondantes à de nouveaux défauts.

On a remarqué que le CPN a identifié correctement tous les entrées présentées.

Etude de cas :

Durant l'évolution complète de cette méthode avec tous les types de défauts possibles, et comme illustrée par la figure (4.2), qu'une certaine confusion apparaît entre :

- a) Le court-circuit biphasé isolé, LLF ;
- b) Le court-circuit biphasé mis à la terre, LLGF ;

Donc, le réseau de neurones étudié dans cette première méthode est incapable de distinguer entre le court-circuit biphasé à la terre et le court-circuit biphasé isolé. Pour tous les autres cas, le réseau indique correctement le type de défaut.

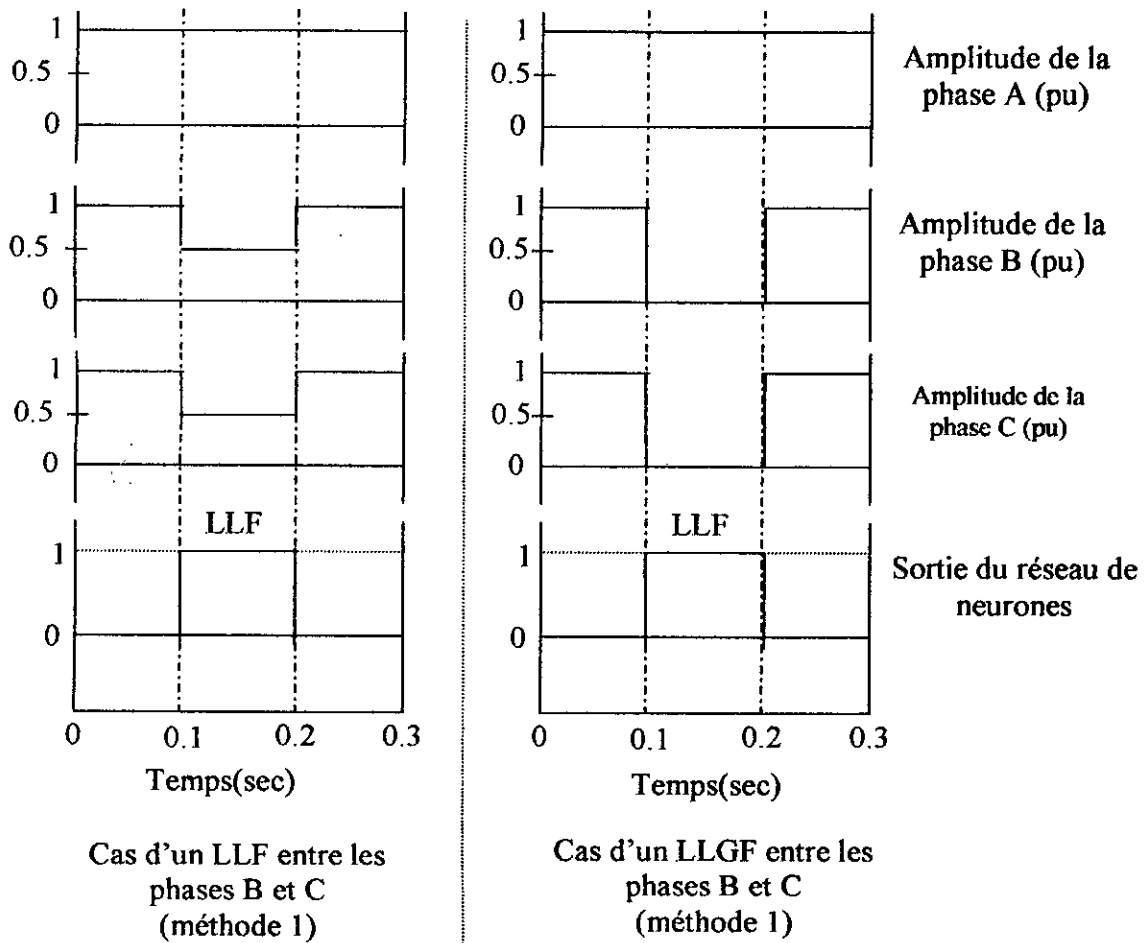


Figure (4.2)

VI.2.2. Méthode 2 :

Comme nous avons vu dans la première méthode, une confusion entre le LLF et le LLGF se présente. Pour distinguer entre ces deux types de courts-circuits, une autre information devient nécessaire. Les études de courts-circuits, ont montré que lors de l'apparition d'un court-circuit biphasé isolé de la terre (LLF), les amplitudes de tensions des deux phases court-circuitées se réduisent jusqu'à 0.5 pu, et le déphasage entre ces deux tensions devient nul (les tensions des deux phases court-circuitées étant en phase). La troisième phase qui n'est pas touchée par le court-circuit est déphasée de 180° par rapport aux deux autres phases. Le réseau de neurones de la première méthode indique un (LLF) si les amplitudes de tensions de deux phases sont égales, et différentes de l'amplitude de tension de la troisième phase. Cette condition est aussi accomplie dans le cas d'un (LLGF), où les tensions des deux phases court-circuitées s'annulent toutes les deux, donc égales, et la tension du troisième phase reste 1pu, d'où la confusion entre la détection des deux défauts.

Une information, arbitraire, supplémentaire devient nécessaire pour aider le réseau de neurones à distinguer entre le (LLF) et le (LLGF). Cette information est le déphasage entre les tensions de phases. C'est le principe de la deuxième méthode.

D'où la nécessité d'utiliser un réseau de neurones élémentaire par phase qui va détecter le déphasage entre les tensions. Les sorties des trois réseaux élémentaires sont liées aux entrées du réseau de neurone principal. Donc nous aurons un réseau de neurones principal et trois réseaux de neurones secondaires (un pour chaque phase). Le réseau de neurones utilisé pour cette méthode est montré à la figure (4.3). Le réseau de neurones propre à la phase A est montré par la figure (4.4), les deux autres réseaux (propres aux deux autres phases) sont identiques à celui de la phase A.

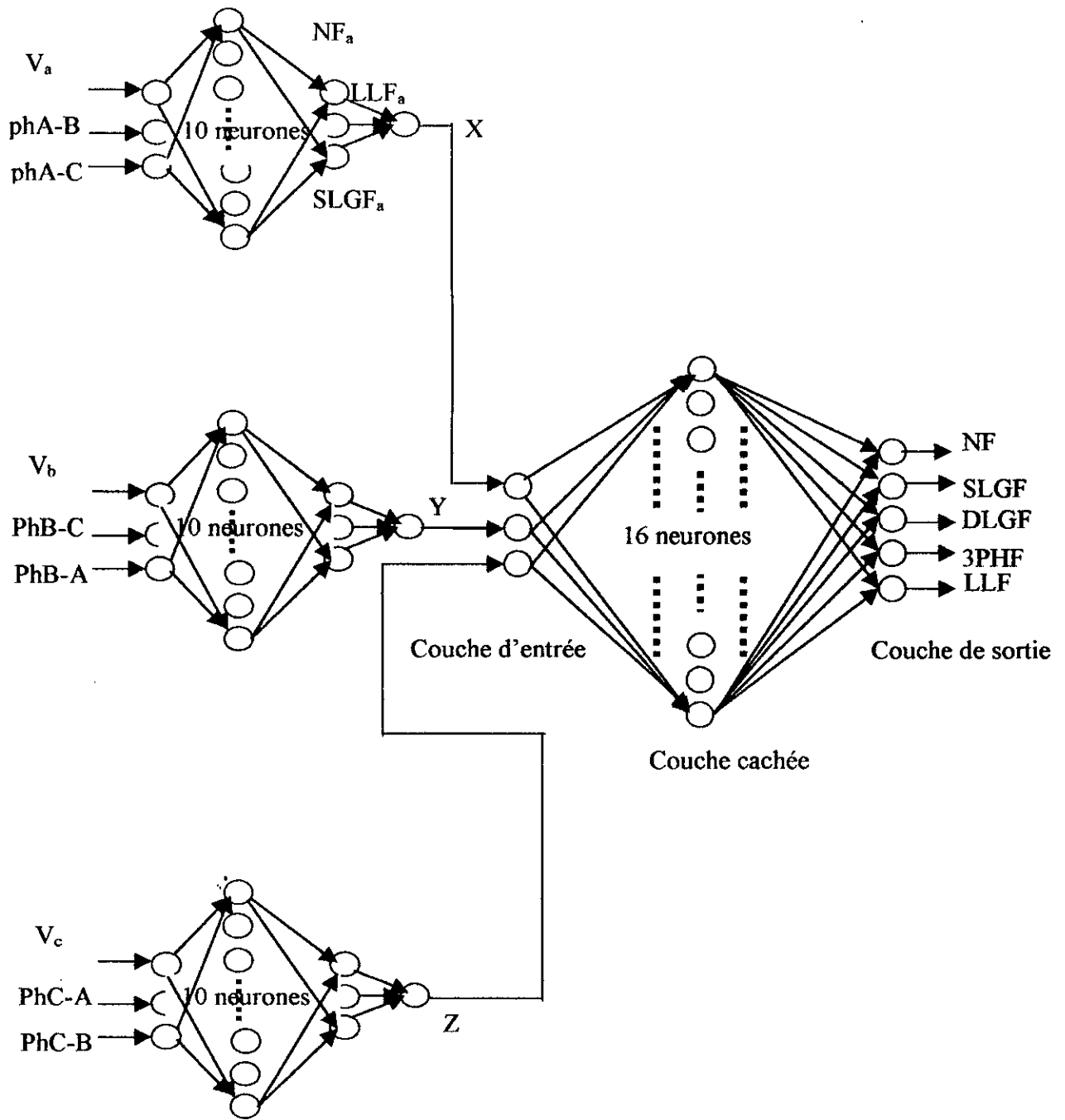


Figure (4.3)

a. Réseau de neurones principal :

Le réseau de neurones principal est presque similaire à celui utilisé dans la première méthode, la différence étant dans la couche compétitive, parce que nous avons 16 vecteurs d'apprentissage, correspondant aux 16 cas possibles en combinant les amplitudes des trois tensions de phases et les différents déphasages entre tensions. L'architecture du réseau de neurones principal est montrée à la figure (4.4).

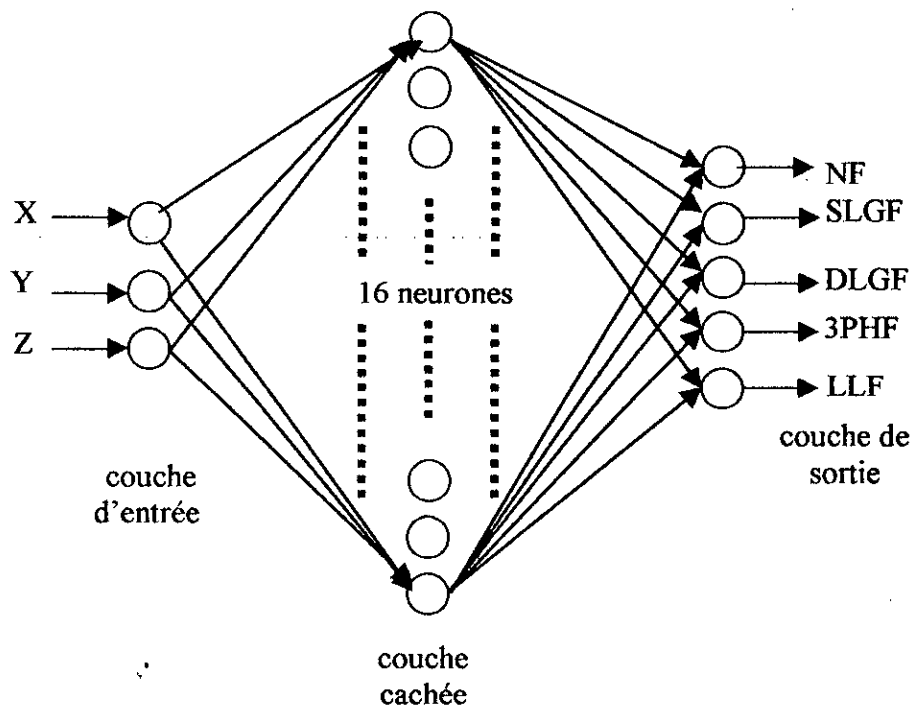


Figure (4.4)

Donc ce réseau est composé de trois couches :

- Une couche d'entrée avec trois neurones ;
- Une couche cachée avec 16 neurones correspondant aux 16 cas possibles ;
- Une couche de sortie avec cinq neurones correspondants aux cinq types de défauts possibles : NF, SLGF, LLGF, LLLF, LLF .

Les entrées de ce réseau principal sont fournies par les trois réseaux de neurones élémentaires.

b- Réseaux de neurones élémentaires par phase :

Pour pouvoir distinguer entre le (LLF) et le (LLGF), il nous faut une information sur l'état de déphasages entre les tensions de trois phases. Cette information peut être assurée par l'utilisation de trois réseaux de neurones propres aux trois phases. Le réseau de neurones élémentaire de la phase A est représenté par la figure (4.5), les deux autres réseaux propres aux deux autres phases étant similaires.

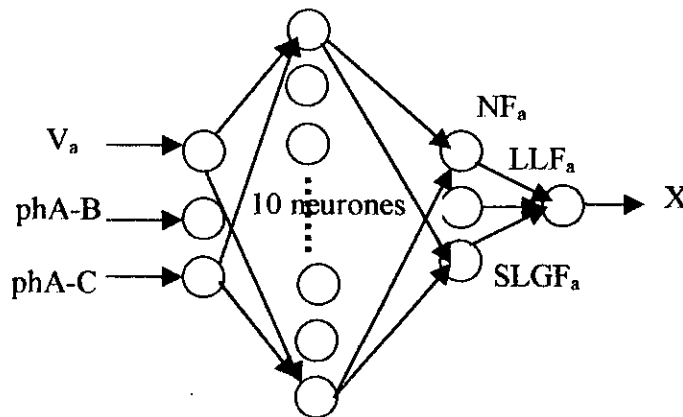


Figure (4.5)

Ce réseau est composé de trois couches :

- Une couche d'entrée avec trois neurones, représentant, respectivement : l'amplitude de tension de la phase *A*, le déphasage avec la phase *B*, (*phA-B*), et le déphasage avec la phase *C*, (*phA-C*).
- Une couche cachée avec 10 neurones, correspondants, aux dix cas possibles de vecteurs d'entrée.
- Une couche de sortie avec trois neurones.

Une sommation non pondérée des sorties de chaque réseau de neurones élémentaire est utilisée pour être injectée directement à l'entrée correspondante du réseau principal. La sortie du réseau élémentaire de la phase *A* correspond à l'entrée *X* du réseau principal, la sortie du réseau élémentaire de la phase *B* à l'entrée *Y* et la sortie du réseau élémentaire propre à la phase *C* à l'entrée *Z*.

Adaptation des poids :

Dans cette méthode, comme nous avons utilisé deux réseaux de neurones : un principal, et l'autre propre aux phases, l'adaptation des poids se fait pour chacun individuellement. Pour les réseaux élémentaires un seul apprentissage est suffisant, parce que les entrées sont les mêmes.

L'adaptation des poids se fait comme précédemment, en changeant les entrées d'apprentissage. La loi d'adaptation de poids est la même que celle utilisée dans la première méthode.

Il y aura aussi une phase de validation et une phase de test mais pour chaque réseau individuellement.

Exécution :***Présentation des réseaux de neurones :******1- Réseau de neurones propre aux phase :***

- Les entrées d'apprentissage sont des vecteurs de trois éléments, nous avons 10 vecteurs pour les faire apprendre par notre réseau, les 10 cas correspondent à tous les cas qui peuvent exister. Les vecteurs d'apprentissage de ce réseau sont données au tableau 4.5.

Tableau 4.5 : les vecteurs d'apprentissage utilisés pour le réseau de neurones propre à la phase A ;

Code	Type de défaut	Entrées		
		V _a	phA-B	phA-C
1	NFa	0.8	0.8	0.8
2	NFa	0.8	0.8	0.2
3	NFa	0.8	0.2	0.8
4	NFa	0.8	0.2	0.2
5	LLFa	0.5	0.8	0.2
6	LLFa	0.5	0.2	0.8
7	SLGFa	0.2	0.8	0.8
8	SLGFa	0.2	0.8	0.2
9	SLGFa	0.2	0.2	0.8
10	SLGFa	0.2	0.2	0.2

- Pour l'apprentissage du réseau de neurones propre à la phase A (le même pour les deux autres phases), une seule itération était suffisante pour adapter les poids. Toutes les classes d'entrée sont bien classifiées.

Les résultats d'apprentissage sont présentés au tableau 4.6.

Tableau 4.6 : les résultats d'apprentissage du réseau de neurones propre à la phase A ;

Code	Type de défaut	Entrées (pu)			Sorties			X
		V _a	phA-B	phA-C	Out1	Out2	Out3	
1	NFa	0.8	0.8	0.8	1	0	0	1
2	NFa	0.8	0.8	0.2	1	0	0	1
3	NFa	0.8	0.2	0.8	1	0	0	1
4	NFa	0.8	0.2	0.2	1	0	0	1
5	LLFa	0.5	0.8	0.2	0	-1	0	-1
6	LLFa	0.5	0.2	0.8	0	-1	1	-1
7	SLGFa	0.2	0.8	0.8	0	0	0	0
8	SLGFa	0.2	0.8	0.2	0	0	0	0
9	SLGFa	0.2	0.2	0.8	0	0	0	0
10	SLGFa	0.2	0.2	0.2	0	0	0	0

Validation des résultats :

Pour valider les résultats du test précédent, on a changé les vecteurs d'apprentissage d'entrée.

Tableau 4.7 : les vecteurs utilisés pour le réseau de neurones propre à la phase A, pendant la phase de validation et les résultats trouvés;

Code	Type de défaut	Entrées (pu)			Sorties			X
		V _a	phA-B	phA-C	Out1	Out2	Out3	
1	NFa	1	1	1	1	0	0	1
2	NFa	1	1	0	1	0	0	1
3	NFa	1	0	1	1	0	0	1
4	NFa	1	0	0	1	0	0	1
5	LLFa	0.5	1	0	0	-1	0	-1
6	LLFa	0.5	0	1	0	-1	1	-1
7	SLGFa	0	1	1	0	0	0	0
8	SLGFa	0	1	0	0	0	0	0
9	SLGFa	0	0	1	0	0	0	0
10	SLGFa	0	0	0	0	0	0	0

On a fait les mêmes étapes appliquées au test d'apprentissage, on a abouti à des résultats identiques à ceux trouvés dans le premier test, tel que le réseau de neurones atteint son équilibre et adapte ses poids dans la première itération de calcul. Après ce test, le réseau de neurones propres aux phases sera capable d'associer pour chaque entrée sa sortie correspondante.

2. Réseau de neurones principal :

Le réseau de neurone principal de cette méthode est presque identique à celui utilisé dans la première. Cependant, nous avons ici 16 neurones cachés, correspondant aux 16 vecteurs d'entrée utilisés pour l'apprentissage de ce réseau.

Les vecteurs d'apprentissage utilisés pour ce réseau sont montrés au tableau 4.8.

Tableau 4.8 : les entrées d'apprentissage du réseau de neurones principal ;

Code	Type de défaut	Entrées (pu)		
		X	Y	Z
1	NF	1	1	1
2	NF	-1	1	1
3	NF	-1	0	1
4	NF	1	-1	1
5	NF	0	-1	1
6	NF	1	1	-1
7	SLGF	0	1	1
8	SLGF	1	0	1
9	SLGF	1	1	0
10	LLGF	0	0	1
11	LLGF	0	1	0
12	LLGF	1	0	0
13	LLF	0	0	0
14	LLF	-1	1	-1
15	LLF	-1	-1	1
16	LLF	1	-1	-1

Après trois itérations, on a atteint l'équilibre et les poids se stabilisent, les 16 vecteurs se classifient correctement, et un seul neurone caché gagne la compétition pour chaque vecteur d'apprentissage.

Les résultats d'apprentissage sont donnés au tableau 4.9.

Tableau 4.9 : les résultats d'apprentissage du réseau de neurones principal ;

Code	Type de défaut	Entrées			Neurone gagnant	Sorties				
		X	Y	Z						
1	NF	0.8	0.8	0.8	4	1	0	0	0	0
2	NF	-0.8	0.8	0.8	2	1	0	0	0	0
3	NF	-0.8	0.2	0.8	1	1	0	0	0	0
4	NF	0.8	-0.8	0.8	2	1	0	0	0	0
5	NF	0.2	-0.8	0.8	13	1	0	0	0	0
6	NF	0.8	0.8	-0.8	11	1	0	0	0	0
7	SLGF	0.2	0.8	0.8	10	0	1	0	0	0
8	SLGF	0.8	0.2	0.8	9	0	1	0	0	0
9	SLGF	0.8	0.8	0.2	8	0	1	0	0	0
10	LLGF	0.2	0.2	0.8	7	0	0	1	0	0
11	LLGF	0.2	0.8	0.2	6	0	0	1	0	0
12	LLGF	0.8	0.2	0.2	5	0	0	1	0	0
13	LLLF	0.2	0.2	0.2	16	0	0	0	1	0
14	LLF	-0.8	0.8	-0.8	15	0	0	0	0	1
15	LLF	-0.8	-0.8	0.8	12	0	0	0	0	1
16	LLF	0.8	-0.8	-0.8	14	0	0	0	0	1

Validation des résultats :

Les vecteurs utilisés pour la phase de validation sont donnés au tableau 4.10. On a choisi des entrées non utilisées pendant la phase d'apprentissage. La validation est suffisante si on trouve des résultats identiques aux résultats trouvés durant la phase d'apprentissage, sinon notre apprentissage est devenu insuffisant, et on le refait pour une deuxième fois avec des entrées différentes de celle utilisées dans le premier.

Tableau 4.10 : les vecteurs du test de validation du réseau de neurones principal et les résultats trouvés;

Code	Type de défaut	Entrées			Neurone gagnant	Sorties				
		X	Y	Z						
1	NF	1	1	1	4	1	0	0	0	0
2	NF	-1	1	1	2	1	0	0	0	0
3	NF	-1	0	1	1	1	0	0	0	0
4	NF	1	-1	1	2	1	0	0	0	0
5	NF	0	-1	1	13	1	0	0	0	0
6	NF	1	1	-1	11	1	0	0	0	0
7	SLGF	0	1	1	10	0	1	0	0	0
8	SLGF	1	0	1	9	0	1	0	0	0
9	SLGF	1	1	0	8	0	1	0	0	0
10	LLGF	0	0	1	7	0	0	1	0	0
11	LLGF	0	1	0	6	0	0	1	0	0
12	LLGF	1	0	0	5	0	0	1	0	0
13	LLLF	0	0	0	16	0	0	0	1	0
14	LLF	-1	1	-1	15	0	0	0	0	1
15	LLF	-1	-1	1	12	0	0	0	0	1
16	LLF	1	-1	-1	14	0	0	0	0	1

Les résultats trouvés dans la phase de validation sont identiques à celles trouvés dans la phase d'apprentissage.

Une fois l'apprentissage terminé pour les deux réseaux de neurones : principal et propre aux phases, on peut passer à tester les deux réseaux ensemble : les entrées des trois réseaux de neurones propres aux phases sont les valeurs efficaces de tension de trois phases, plus les déphasages entre les tensions. Les entrées du réseau principal sont les sorties des trois réseaux propres aux phases.

Phase de test de deuxième méthodes :

Les vecteurs d'entrées et les résultats trouvés dans cette phase sont présentés au tableau 4.11.

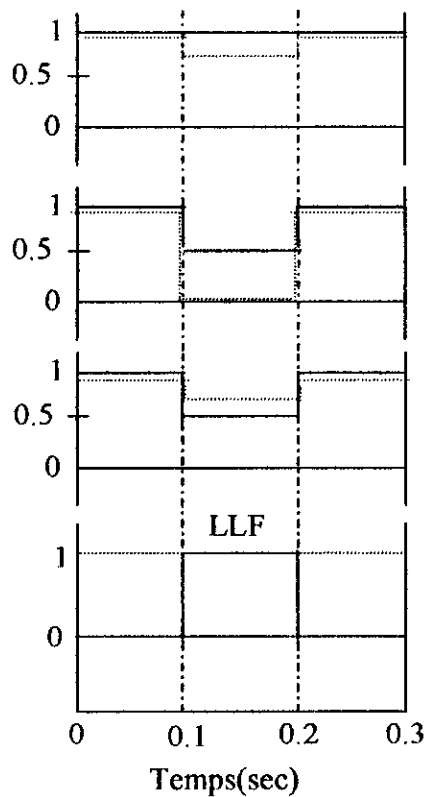
Tableau 4.11 : les résultats trouvés pendant la phase de test ;

Entrées						Sorties					Type de défaut
Va	Vb	Vc	Ph A-B	Ph A-C	Ph B-C	Out1	Out2	Out3	Out4	Out5	
1	1	1	1	1	1	1	0	0	0	0	NF
0	1	1	0/1	0/1	0/1	0	1	0	0	0	SLGFa
1	0	1	0/1	0/1	0/1	0	1	0	0	0	SLGFb
1	1	0	0/1	0/1	0/1	0	1	0	0	0	SLGFc
0	0	1	0	1	1	0	0	1	0	0	LLGFa-b
0	1	0	1	0	1	0	0	1	0	0	LLGFa-c
1	0	0	1	1	0	0	0	1	0	0	LLGFb-c
0	0	0	0	0	0	0	0	0	1	0	LLLF
0.5	0.5	1	0	1	1	0	0	0	0	1	LLFa-b
0.5	1	0.5	1	0	1	0	0	0	0	1	LLFa-c
1	0.5	0.5	1	1	0	0	0	0	0	1	LLFb-c

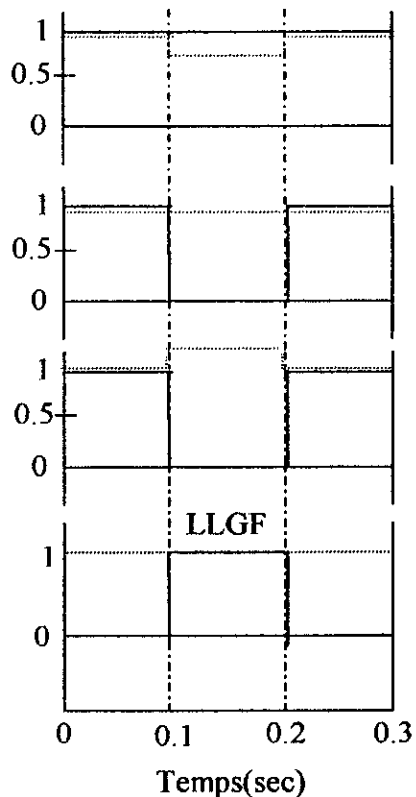
Après l'application des vecteurs d'entrée présentés au tableau 4.11, on a trouvé de bon résultats.

Etude de cas :

Les 5 types de défauts (montrés au tableau 4.1) sont classifiés correctement en utilisant le réseau de neurones du deuxième méthode. La distinction entre le court-circuit biphasé isolé et le court-circuit biphasé à la terre est faite dans cette méthode, ce qui n'est pas le cas avec la première méthode. Les résultats sont montré au tableau 4.11.



Cas d'un LLF entre les phases B et C avec la (méthode 2)



Cas d'un LLGF entre les phases B et C avec la (méthode 2)

- Amplitude de tension de la phase A (pu)
- Déphasage entre phases A et B (1pu=240°)
- Amplitude de tension de la phase B (pu)
- Déphasage entre phases B et C (1pu=240°)
- Amplitude de tension de la phase C (pu)
- Déphasage entre phases A et C (1pu=240°)
- Sortie du réseau de neurones

Figure (4.6)

VI.3. Conclusion :

Le CPN est un réseau de neurones capable de classer des vecteurs d'entrées regroupés dans ce qu'on appelle : classes d'entrée.

Du fait que le CPN est composé de deux différents réseaux de neurones (KOHONEN et GROSSBERG), son apprentissage est un peu compliqué en le comparant avec d'autres types de réseaux où on trouve une seule loi d'apprentissage.

Un point qui peut jouer un rôle important dans les études des réseaux de neurones est l'initialisation des poids. Une bonne initialisation peut conduire à l'équilibre dès la première itération. Une mauvaise initialisation peut amener le réseau de neurones à diverger.

Du point de vue rapidité, le CPN est très rapide parce qu'il atteint son équilibre avec un nombre d'itérations très petit.

L'identification des défauts de courts-circuits peut se faire correctement en utilisant le CPN.

Nous avons deux méthodes à appliquer. La première elle est simple, mais n'est pas suffisante pour la distinction entre le LLF et le LLGF. Pour la deuxième méthode, le nombre plus important d'éléments implique une complexité supplémentaire, et un temps de calcul plus important.

Conclusion générale

Les divers avantages des réseaux de neurones artificiels tels que : robustesse, rapidité et capacité de généralisation, ont permis à ces réseaux d'être largement utilisés dans plusieurs domaines d'application.

Dans cette étude, nous avons montré la possibilité d'utilisation des réseaux de neurones artificiels pour l'identification des défauts de courts-circuits pouvant apparaître dans un réseau électrique.

Les réseaux de neurones artificiels peuvent jouer un rôle très important pour identifier un défaut, du point de vue rapidité d'exécution ainsi que de la précision de réponse. Ceci est un atout intéressant pour l'opérateur dans un centre de répartition, car cela lui permet d'agir vite en fonction du problème posé.

Deux méthodes ont été présentées dépendement du choix des entrées utilisées (amplitudes de tensions des trois phases pour une première méthode, et amplitudes de tensions plus les déphasages entre les trois phases pour une deuxième). Un programme a été élaboré dans ce sens et des simulations ont montré des résultats très satisfaisants.

Si la première méthode n'a pas pu distinguer entre le court-circuit biphasé isolé de la terre et le court-circuit biphasé à la terre, l'utilisation de la deuxième a pu corriger ce handicap.

Donc, si on s'intéresse seulement à détecter le défaut sans être obligé de donner des informations complètes sur son type, la première méthode est préférable. Mais si on veut préciser le type de défaut (isolé ou à la terre), la deuxième méthode devient nécessaire. La deuxième méthode est en fait plus générale et plus complète que la première.

Une autre méthode d'identification ne faisant pas l'objet de ce travail pour des raisons de temps, peut être étudiée par ceux qui veulent aller plus loin dans ce projet. L'architecture du réseau de neurones utilisé dans cette autre méthode sera similaire à celle utilisée dans notre deuxième méthode mais avec un nombre de neurones cachés du réseau principal beaucoup plus grand. Les vecteurs d'entrée seront les valeurs instantanées de tension.

De plus, le rôle des réseaux de neurones ne se limite pas au problème de l'identification des défauts, mais peut être également appliqué pour leur localisation. Ainsi ce travail peut être également suivi d'une étude de localisation des défauts de courts-circuits.

Références bibliographiques

- [1] C. Corroyer, "Protection des réseaux électriques", *Techniques d'ingénieur D4800*.
- [2] J.F. Jodouin, "Réseaux de neurones, principes et définitions", *Hermès*, Paris 1994.
- [3] J.P. Barret, P. Bornard, B. Meyer, "Simulations des réseaux électriques", *Editions Eyrolles*, 1997.
- [4] G.W. Stagg and A.H. El-Abiad, "Computer Methods in Power System Analysis", *McGraw-Hill Book Company*, 1968.
- [5] J.A. Freeman., D.M. Skapura, "Neural Networks Algorithms Applications and Programming Techniques", *Addison-Wesley Publishing Company*, 1992.
- [6] F. Cahen, "Circuits et Réseaux en Régime permanent, tome I", *Gauthier-Villass Editeur*, 1970.
- [7] J. Lesenne, F. Notlet et G. Séguier, "Introduction à l'électrotechnique approfondie", *Technique et documentation*, 1981.
- [8] A. Bisma et A.N. Benachour, "Evaluation de la sécurité statique des réseaux électriques par les réseaux de neurones », *PFE USTHB*, 1997.
- [9] MM. Heniche, "Linéarisation de capteurs à l'aide des réseaux de neurones artificiels", *PFE ENP*, 1994.
- [10] P.M. Anderson and A.A. Fouad, "Power System Control and Stability", *IEEE PRESS, Editorial Board*, 1993.
- [11] D.P. Morgan and C.L. Scofield, "Neural Networks and Speech Processing", *Kluwer Academic Publishers*, 1994.